SACLA
2017
infinite possibilities

NORTH-WEST UNIVERSITY ®
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT

Proceedings of the 46th Annual Conference of the Southern African Computer Lecturers' Association (SACLA 2017)

July 3-5, 2017

Valley Lodge

Magaliesburg

South Africa

Editor: Janet Liebenberg

Copyright Notice

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT

SACLA ®
2017
infinite possibilities

The SACLA 2017 organising committee is not responsible for errors or omissions from individual papers contained in these proceedings. Technical electronic anomalies are possible and unavoidable during the compilation process. The publisher is not responsible for the accuracy and validity of information contained in these papers, nor is it responsible for the final use to which this book may be used.

The Chairman of the SACLA 2017 Programme Committee attests as follows:

> This is to confirm that all conference paper submissions that appear in these proceedings have been through a double---blind peer review process prior to acceptance into the final conference program.

Liebenberg

Editor: Janet Liebenberg
Published by the SACLA 2017 Organising Committee
Computer Science and Information Systems
Potchefstroom Campus
North-West University
Private Bag X6001
Potchefstroom
2520

ISBN: 978-1-86822-846-1

Message from the Conference Chair

Welcome to the 46th Annual Conference of the South African Computer Lecturers' Association (SACLA 2017). I trust that you will have an energizing and enlightening conference at Valley Lodge, Magaliesburg.

The goal of SACLA-conferences is to provide participants with an opportunity to share ideas, while maintaining a high level of academic input from all involved. You are encouraged to participate fully at the conference and to use the breaks and evenings to interact with colleagues from your own and other institutions.

The accepted papers reflect current trends in teaching and learning in Computer Science and Information Systems. This is in keeping with the theme of SACLA 2017, Keeping Education Relevant: Infinite possibilities.

We are fortunate to have an international keynote speaker from the USA, Grandon Gill. In addition, we have scheduled a workshop on 'The South African Computing Accreditation Board (SACAB) – Implementation and Documentation'.

On behalf of the SACLA community, I wish to express our deepest appreciation to our sponsors, AdaptIT, IITPSA, IBM, Oracle and SAP.

A successful conference requires the effort of many individuals. We would like to thank the members of the program committee for their hard work. We are grateful to the authors who submitted their papers to this conference. I also wish to extend my sincere thanks to all members of the organising committee and congratulations for a job well done.

We hope that everyone will have a good time at Valley Lodge.

Estelle Taylor
SACLA 2017 Conference Chair

Message from the Program Committee Chair

It is with great pleasure that I compile the papers after an intense period of reviewing. I extend my thanks and appreciation to the reviewers who provided extensive and insightful reviews.

I commend the authors for meeting the deadlines that are so important for the smooth running of the process of submissions and review. I thank the authors for using this forum to share and participate in the event.

This year we received 63 papers for review. The program committee consisted of both local and international experts in the fields of Computer Science and Information Systems Education, with the necessary expertise and interest in subjects relevant to the theme of the conference. The program committee had 53 members, of which 30 were international members. Each paper was reviewed by 3 reviewers in a rigorous double-blind peer review process. The programme chairs solicited additional expert reviews in cases where further clarity was warranted. We accepted 40 papers of high quality for presentation at SACLA 2017. Before publishing a paper, authors had to include the corrections as stated by the peer reviewers. Of the 40 presented papers, 22 papers will be published in the Springer publication: *Communications in Computer and Information Sciences* and therefore, only the titles and short abstracts of these papers are included in this publication.

Thank you again to all our reviewers and authors. I am looking forward to meeting all of you at the conference.

Janet Liebenberg
SACLA 2017 PC Chair

Organising Committee

Estelle Taylor	***Conference Chair***
Janet Liebenberg	***PC Chair***
Susan Campher	***Finance, General Management***
Henry Foulds	***Webmaster***
Trudie Benadé **Kobie Fourie** **Dimi Keykaan** **Anne Mans** **Jeanne-Marie Maree** **Dirk Snyman** **Kobus van Aswegen** **Henri van Rensburg**	***General Conference Management, Communications, Accommodation, Logistics, Technical Support***

SACLA 2017 Program Committee

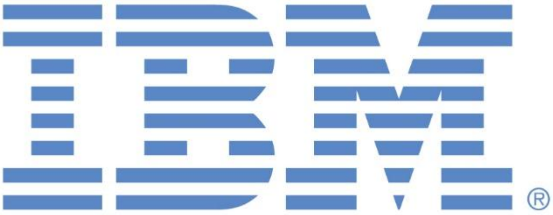
Name	University/Affiliation	Country
Liz Bacon	University of Greenwich	UK
Trudie Benade	North-West University	South Africa
Torsten Brinda	University of Duisburg-Essen	Germany
André Calitz	Nelson Mandela Metropolitan University	South Africa
Charmain Cilliers	Nelson Mandela Metropolitan University	South Africa
Peter Csaba-Ölveczky	University of Oslo	Norway
Jörg Desel	Fernuniversität in Hagen	Germany
Gunther Drevin	North-West University	South Africa
Lynette Drevin	North-West University	South Africa
Peter Forbrig	University of Rostock	Germany
Kurt Geihs	University of Kassel	Germany
Roelien Goede	North-West University	South Africa
Joanna Goode	University of Oregon	USA
Leila Goosen	University of South Africa	South Africa
Irene Govender	University of Kwazulu Natal	South Africa
Stefan Gruner	University of Pretoria	South Africa
Mariele Hagen	Dortmund University of Applied Sciences and Arts	Germany
Ruber Hernandez	University of Namibia	Namibia
Michaela Huhn	Ostfalia University of Applied Sciences	Germany
Magda Huisman	North-West University	South Africa
Eduan Kotze	University of the Free State	South Africa
Herbert Kuchen	University of Münster	Germany
Janet Liebenberg	North-West University	South Africa
Linda Marshall	University of Pretoria	South Africa
Audrey Masizana	University of Botswana	Botswana
Maggie McPherson	University of Leeds	UK
Dawn Medlin	Appalachian State University	USA
Katharina Mehner-Heindl	Offenburg University of Applied Sciences	Germany
Mohamed Mosbah	University of Bordeaux 1	France
Matthew Mullarkey	University of South Florida	USA
Jane Nash	Rhodes University	South Africa
Lakshmi Narasimhan	University of Botswana	Botswana
Gabriel Nhinda	University of Namibia	Namibia
James Njenga	University of the Western Cape	South Africa
Benny Nyambo	University of Zimbabwe	Zimbabwe
Kwete Nyandongo	University of Johannesburg	South Africa
Peter Ölveczky	University of Oslo	Norway
Julia Padberg	Hamburg University of Applied Sciences	Germany
Vreda Pieterse	University of Pretoria	South Africa
Karen Renaud	University of Glasgow	Scotland

Markus Roggenbach	Swansea University	UK
Michael Ruecker	Humboldt University of Berlin	Germany
Ian Sanders	University of South Africa	South Africa
Brenda Scholtz	Nelson Mandela Metropolitan University	South Africa
Lisa Seymour	University of Cape Town	South Africa
Upasana Singh	University of Kwa-Zulu Natal	South Africa
Estelle Taylor	North-West University	South Africa
Marko van Eekelen	Open University of the Netherlands	Netherlands
Corne van Staden	University of South Africa	South Africa
Janis Voigtländer	University of Nijmegen	Netherlands
Gottfried Vossen	University of Münster	Germany
Jim Woodcock	University of York	UK
Olaf Zukunft	Hamburg University of Applied Sciences	Germany

SACLA
2017
infinite possibilities

Sponsors

Annual Conference of the Southern African Computer Lecturer's Association



Program

Monday, July 3, 2017		
12:00 - 17:00	Arrival and Registration – Lounge Area (Reception)	
13:00 – 13:50	Lunch – Hotel restaurant	
13:50 – 14:00	Welcome / Opening Venue: Henry Hartley	
	Venue: Kingfisher HoD-meeting (meeting of the head(s) of departments / subject groups)	Venue: Henry Hartley **Session 1** **Issues relating to the relevance of teaching and learning of CS and IS** Session chair: Lynette Drevin
14:00 – 14:20	HoD-meeting (meeting of the head(s) of departments / subject groups)	Presentation1_1: Roelien Goede. Using Minesweeper to Teach Data Structures and Algorithms: a Problem-based Learning Perspective (#45).
14:25 – 14:45		Presentation1_2: Maureen van Den Bergh and Erica Pretorius. Cybercitizenship Awareness Module designed for first year university students (#4).
14:50 – 15:10		Presentation1_3: Vreda Pieterse, Hein S Venter and Stefan Gruner. Students' Over-Estimation of Their Own Academic Ability: A Case in Undergraduate Computer Science(#17).
15:15 – 15:35		Presentation1_4: Arthur James Swart. Undergraduate students who regularly complete online self-reflective assessments reap academic success (#42).
15:35 – 16:00	Break - Stoep	
16:00 – 16:20	HoD-meeting (meeting of the head(s) of departments / subject groups)	Presentation1_5: Vreda Pieterse and Katherine Malan. Authorized Cheat-sheets as an Educational Tool in Computer Science Examinations (#21).
16:25 – 16:45		Presentation1_6: Petri Jooste. A General Purpose Computer Lab with Linux and Virtual Machines (#49).
16:50 – 17:10	Meeting of the SACLA Executive Committee (André Calitz, Lisa Seymour, Estelle Taylor, Linda Marshall)	Presentation1_7: Johan Vorster and Leila Goosen. a Framework for University Partnerships Promoting Continued Support of e-Schools (#43).
19:00 – 21:00	Cocktail Dinner Venue: Maloney's Bar	

Tuesday, July 4, 2017		
07:00 – 08:30	Breakfast – Hotel Restaurant	
08:30 – 09:30	Venue: Henry Hartley Keynote address: Blending IT Research and Teaching to Create a More Locally Relevant Curriculum. Grandon Gill.	
 09:30 – 10:00 10:00 – 10:30	Venue: Henry Hartley Sponsor: IITPSA (Tony Parry and Adrian Schofield) Sponsor: Oracle (??)	
	Venue: Henry Hartley **Session 2** **Keeping education relevant to students of a new generation** Session chair: Lisa Seymour	Venue: Thomas Baines **Session 3** **Keeping education relevant by collaborating with industry** Session chair: Jan Kroeze
10:35 – 10:55	Presentation 2_1: Anwar Parker and Jean-Paul Van Belle. iGeneration as Students: Exploring the relative Access, Use of, and Perceptions of IT in Higher Education (#26).	Presentation 3_1: Eduan Kotzé. A Survey of Data Scientists in South Africa (#16).
11:00 – 11:20	Presentation 2_2: Daniel Le Roux and Douglas Parry. A New Generation of Students: Digital Media in Academic Contexts (#6).	Presentation 3_2: Roelien Goede. A critical systems perspective on project-based learning: Guidelines for using industry data for BI student projects (#44).
11:20 – 11:40	Break - Stoep	
11:40 – 12:00	Presentation 2_3: Estelle Taylor and Kobus van Aswegen. Students' approaches to learning: Is it changing? (#35).	Presentation 3_3: Carin Venter. Critical Reflection on a Data Warehouse/Business Intelligence Course: Does It Prepare Students to be Successful Practitioners? (#12).
12:05 – 12:25	Presentation 2_4: Nitesh Harry and Dr Sumarie Roodt. The effect of using YouTube in the Classroom for Student Engagement of the Net Generation on an Information Systems course (#41).	Presentation 3_4: Andre Calitz, Jean Greyling and Margaret Cullen. Industry vs Post-Graduate Studies: CS and IS Alumni Perceptions (#55).
12:30 – 12:50	Presentation 2_5: Alta Van der Merwe, Aurona Gerber and Hanlie Smuts. Mapping a Design Science Research Cycle to the Postgraduate Research Report (#31).	Presentation 3_5: Anine Kruger, Machdel Matthee and Marita Turpin. Information Systems as creative products: What are industry's expectations? (#34).
12:55 – 13:15	Presentation 2_6: Sibusisiwe Dube. The new generation of students' ICT needs and expectations: Case of a developing country (#39).	Presentation 3_6: Duane Boucher and Roxanne Piderit. Application of an Action Research Process: Reflections on an Undergraduate Information Systems Software Development Project (#57).
13:15 – 14:00	Lunch – Hotel Restaurant	
	Venue: Henry Hartley **Session 4** **Keeping education relevant by using technology and gaming in education**	Venue: Thomas Baines **Session 5** **Keeping CS/IS courses relevant** Session chair: Carina de Villiers

	Session chair: André Calitz	
14:00 – 14:20	Presentation 4_1: Eleftherios Nicolau and Lisa Seymour. Mobile and Game Usage, Gender and Attitude towards Computing Degrees (#8).	Presentation 5_1: H.W. Pretorius and M.J. Hattingh. Reflections of summer school learners on the factors influencing their poor performance in Systems Analysis and Design Course (#50).
14:25 – 14:45	Presentation 4_2: Ariel Rosenfeld, Abejide Ade-Ibijola and Sigrid Ewert. Regex Parser II: Teaching regular expression fundamentals via educational gaming (#5).	Presentation 5_2: Wai Sze Leung. Cheap Latex, High-End Thrills: A Fantasy Exercise in Search and Seizure (#58).
14:50 – 15:10	Presentation 4_3: Colin Pilkington. Questioning the value of vodcasts in a distance learning theoretical computer science course (#18).	Presentation 5_3: Romeo Botes and Imelda Smit. The infinity approach: a case study (#65).
15:15 – 15:35	Presentation 4_4: Leila Goosen and Ronell van der Merwe. Keeping ICTs in Education Community Engagement Relevant: Infinite possibilities? (#46).	Presentation 5_4: Linda Marshall. A comparison of the core aspects of the ACM/IEEE Computer Science Curriculum Volumes - How do the topics compare? (#23).
15:35 – 16:00	Break - Stoep	
16:00 – 16:20	Presentation 4_5: Eduan Kotzé. Augmenting a Data Warehousing Curriculum with Emerging Big Data Technologies (#3).	Presentation 5_5: Andre Calitz, Jean Greyling and Arthur Glaum. Evaluating Alumni Satisfaction of an CS/IS Department(#60).
16:25 – 16:45	Presentation 4_6: Lynette Drevin, Albie le Grange, Martin Park. The concept of mobile applications as educational tool to enhance information security awareness (#28).	Presentation 5_6: Hamzh Alaiat and Isabella Venter. Graduate attributes for Computer Science in Libya (#37).
16:45 – 17:30	Venue: Henry Hartley AGM (all delegates)	
19:00 - late	Venue: Cashan Gala Dinner	

Wednesday, July 5, 2017		
07:00 – 09:00	Breakfast – Hotel Restaurant	
09:30 – 10:00	Venue: Henry Hartley Sponsor: SAP (De Wet Naude) SAP Next-Gen - Connecting Next Generation Talents to the SAP Ecosystem	
10:00 – 10:30	Sponsor: IBM (Angeline Raseroka)	
10:30 – 11:00	Break - Stoep	
	Venue: Henry Hartley **Session 6 Keeping programming education relevant** Session chair: Janet Liebenberg	Venue: Thomas Baines Workshop: The South African Computing Accreditation Board (SACAB) – Implementation and Documentation. Prof André Calitz.
11:00 – 11:20	Presentation 6_1: Glenda Barlow-Jones and Duan van der Westhuizen. Problem solving as a predictor of programming performance (#14).	
11:25 – 11:45	Presentation 6_2 : Apostolos Paul Giannakopoulos. Programming: A wicked subject? (#27).	
11:50 – 12:10	Presentation 6_3: Hussein Suleman, Stephan Jamieson and Maria Keet. Testing Test-Driven Development (#47).	
12:15 – 12:35	Presentation 6_4: Glenda Barlow-Jones and Duan van der Westhuizen. Pre-entry attributes thought to influence the performance of students in computer programming (#19).	
12:40 – 13:00	Presentation 6_5: Abejide Ade-Ibijola. Automata-aided Estimation of Similarity in Novice Programs (#7).	
13:00 – 14:00	Lunch – Hotel Restaurant	

Table of Contents

Part 1: Short Abstracts

Eduan Kotzé. Augmenting a Data Warehouse Curriculum with Emerging Big Data Technologies

Abstract: In recent times the traditional database environment as we know it, has been subject to tremendous change. This change was predominately driven by the arrival of new types of datasets, which included social media, sensory logs and webpages. Organizations relying on relational database systems for their data warehouse architectures are finding innovative ways to integrate these datasets into their daily decision-making processes using big data technologies. Recent research, however, indicates that demand for graduates with big data and data warehousing skills far exceeds the supply of those graduating with the appropriate skills. Research has also indicated that the shortage of big data skills hamper organizations to diversify the platform types of their data warehouses. This need can only be addressed if students gain the necessary skills and knowledge in technologies and methodologies related to big data warehousing. The paper addresses the problem by reporting on a pilot study where a module on integrating big data into a data warehouse is used to educate postgraduate students. A key contribution of this paper is the description of innovative big data technologies related to data warehousing, module resources, virtual machines and sample projects. These could be helpful in supporting a hands-on learning experience with big data warehousing.

Ariel Rosenfeld, Abejide Ade-Ibijola and Sigrid Ewert. Regex Parser II: Teaching regular expression fundamentals via educational gaming

Abstract: We present an educational game that is expected to improve the teaching of and stimulate the interest in regular expressions. The game generates regular expressions using pseudo-random numbers and predefined templates and asks the player to provide strings that are in the language corresponding to the given regular expression while a timer counts down. Points in the game are awarded on how many strings the player was able to enter before time runs out.

Daniel Le Roux and Douglas Parry. A New Generation of Students: Digital Media in Academic Contexts

Abstract: The growing presence of digital media in the lives of university students signals a change in how use of such media in educational contexts should be viewed. Institutional focus on technologically mediated education and the promotion of blended learning initiatives further serve to encourage media use in academic settings. Scant attention has been afforded to the potential negative consequences arising from heightened media engagement. This is especially the case in areas of study where technological artifacts are often the medium and the subject of interest, for instance the computer and information sciences. In this study a survey methodology (n=1 678) is employed to investigate students' use of media, as well as the behavioural beliefs, norms and motivators surrounding such use. Findings suggest that demographic factors are irrelevant as predictors of media use — off-task media use during lectures is the norm for all students. Furthermore, no link has been found between institutional attempts to curb off-task media use and changes in students' behaviour. In terms of beliefs, our findings suggest that even students who believe use in academic contexts to be unacceptable, still frequently engage in media use in such contexts. While we acknowledge the value of,

and certainly encourage continued blended learning initiatives, our findings raise a red flag regarding their implementation.

Eleftherios Nicolau and Lisa Seymour. Mobile and Game Usage, Gender and Attitude towards Computing Degrees

Abstract: There is a global demand for graduates with computing skills as well as global shortages of computing professionals especially women. In Africa, these short-ages are more critical. To resolve this problem, studies in the past have argued for increased computing usage in schools that would lead to increased interest in computing degrees. Recently computing usage in the form of mobile devices and mobile gaming has increased substantially in schools yet in South Africa interest in studying computing at a tertiary level has not increased substantially. Hence, this study aims to determine whether mobile and gaming usage impacts school pupil's attitude towards computer related degrees. The survey of 292 South African secondary school pupils confirms low interest in computing degrees, especially by girls, as well as misperceptions of computing degrees. The study highlights the urgent need for computing career education. An IT Usage Attitude Model is proposed and validated in which mobile and gaming usage vary substantially based on gender but usage does not influence attitude towards computing degrees.

Glenda Barlow-Jones and Duan van der Westhuizen. Problem Solving as a Predictor of Programming Performance

Abstract: The purpose of this paper is to establish what correlational relationship exists between a student's problem solving ability and their academic performance in first-year level programming courses. With this aim in mind, the students' achievement in the programming courses is specified as the dependent variable and four programming aptitude tests for logical reasoning, non-verbal reasoning, numerical reasoning and verbal logic are specified as the independent variables. The study group consists of 379 students studying the National Diploma Business Information Technology at the University of Johannesburg, Johannesburg and the National Diploma Information Technology at the Tshwane University of Technology, Pretoria. The findings reveal there is a correlation between a student's logical reasoning, numerical reasoning and verbal logic and performance in computer programming modules. The correlation between students' non-verbal reasoning and performance in computer programming modules was, however, not significant.

Eduan Kotzé. A Survey of Data Scientists in South Africa

Abstract: In the past, humans generated data that was very structured and stored in relational database tables. However, with the introduction of mobile technologies and supercomputers, both humans and machines now generate enormous amounts of semi-structured and unstructured data as well. Academic programs at South African Higher Education Institutions have predominantly educated students in man-aging and storing data using relational database technology. However, this is no longer sufficient. South Africa, as a country, will need to educate more students to manage and process structured, semi-structured and unstructured data. The main purpose of this study was to examine the status of data scientists, a role typically associated with managing these new data sets, in South Africa. The study examined the skills, knowledge and qualifications these data scientists require to perform their daily tasks. The study found that to be a successful data scientist in South Africa, one

will need a wide range of skills and knowledge with respect to different aspects of data processing and data management. It was also evident that a data scientist should preferably not be an expert in a single discipline, but rather be multi-skilled with expertise in several disciplines. Finally, soft skills are also important and need to be included in the academic training of data scientists. The study combined the requirements that a data scientist should meet (qualification, skills and knowledge), and offered suggestions that could be considered when designing the curriculum for a data scientist academic program.

Colin Pilkington. Questioning the value of vodcasts in a distance learning theoretical computer science course

Abstract: Newer technologies have the potential to enhance how teaching and learning in higher education happen. Teaching and learning practice should take up the affordances of technology where they can be used as tools of instruction. The creation, use, and benefits of vodcasts in education provide a background to the study. The context of the provision of 12 supplementary vodcasts in a distance learning theoretical computer science course drove questions relating to whether these led to a statistically significant improvement in examination question, and final, marks. Vodcast access numbers and examination marks for the four semesters for which vodcasts were available, as well as the preceding four semesters, were analyzed. The results show that there was no statistically significant difference in marks that could convincingly be linked to the provision of the vodcasts. The results are discussed in relation to the concepts of communication, accessibility, and online use of learning materials.

Glenda Barlow-Jones and Duan van der Westhuizen. Pre-Entry Attributes Thought to Influence the Performance of Students in Computer Programming

Abstract: This study attempted to isolate seven pre-entry attributes that were thought to influence the performance of students in the module Development Software 1 (programming). The pre-entry attributes included students' problem solving ability, socio-economic status, educational background, performance in school mathematics, English language proficiency, digital literacy and previous programming experience. The following research question was posed: "To what extent do selected pre-entry attributes influence South African students' performance in computer programming modules?" The data analysed indicated that there is a correlation between the variables problem solving, digital literacy and previous programming experience and performance in programming modules. There was no correlation found between the variables socio-economic status, educational background, Grade 12 mathematics mark and English mark and performance in programming modules.

Linda Marshall. A comparison of the core aspects of the ACM/IEEE Computer Science Curriculum Volumes - How do the topics compare?

Abstract: The results of a comparison between the ACM/IEEE Computer Science curriculum volumes of 2001, 2008 and 2013 is presented. The 2013 curriculum volume does provide information regarding equivalences by referring to previous volumes that have been published. This information however is limited to the knowledge area changes that have taken place. These changes are at a very high level. A finer grained comparison is necessary. This paper considers the differences on the knowledge unit and topic levels of the curriculum specifications. To this end, a comparison between the curriculum

volumes is presented. A comparison framework that accepts curricula modelled as digraphs as input is applied. The results of the application of the framework are used to highlight the differences between the curriculum volumes.

Anwar Parker and Jean-Paul Van Belle. iGeneration as Students: Exploring the relative Access, Use of, and Perceptions of IT in Higher Education

Abstract: This paper explores iGeneration age students as they encountered e-learning in higher education. The paper examined the iGeneration as a distinct technologically enhanced generation and described them within the context of first-year students with specific techno-logical capabilities and needs. It was found that, although the literature strongly suggested that their technological preferences would outweigh those provided at tertiary education, they showed positive attitudes towards the use of technology at tertiary level. As such, the expectations associated with a new generation may not necessarily warrant a reform in order to suit the generation. However, the findings do suggest a high degree of consensus among the majority of responses indicating that, as per the literature, that they may be identified as a homogenous sample. Furthermore, it is believed that these students value the role that technology plays as a facilitator towards education, especially at university.

Apostolos Paul Giannakopoulos. Programming: A wicked subject?

Abstract: Studying programming in an Open Distance Learning setup could be more challenging than in a contact setup. It could be characterised as a wicked problem. Wicked problems are problems that are complex and current problem techniques fail to solve the problem. Typical wicked problems could be, poverty, unemployment, attrition rates in higher education, lack of sufficient housing and crime in the South African context. Wicked problems require a kind of unorthodox, innovative or creative way.
Programming as a rule can be considered a subject that presupposes the existence of a number of cognitive functions such as problem solving. Problem solving in general requires critical thinking and critical thinking is characterised by logic, decision making, paying attention to detail, the existence of all different types of knowledge (e.g. procedural, conceptual, strategic, schematic and declarative). All these are prerequisites in the learning of programming. This paper will show that treating programming as a wicked problem could shed some light as to why not many students can be successful in programming.

Alta Van der Merwe, Aurona Gerber and Hanlie Smuts. Mapping a Design Science Research Cycle to the Postgraduate Research Report

Abstract: Design Science Research (DSR) is used in different domains to construct artefacts. In the Information Systems (IS) domain, DSR is accepted as a research method for the development of IS artefacts since the publication of Hevner et al's paper on DSR in 2004 [1]. DSR has therefore also been adopted as a research strategy in postgraduate studies. One of the most challenging aspects of any postgraduate study is determining the structure of both the study and the research report, which should reflect all the components necessary to build a convincing argument that support the main research contribution. This challenge is aggravated when DSR, especially one of the proposed DSR methods, are adopted as research approach. Using an analysis of several postgraduate research reports that successfully adopted DSR, this paper developed a mapping between the proposed structure of a research report and the DSR process model of Vaishnavi and Kuechler's [2]. The mapping is presented in four scenarios and validated with examples. The mapping and scenarios were subsequently validated in a workshop with doctoral postgraduate students. The feedback results indicate that the mapping was useful for all the postgraduate DSR students that needed to structure their research reports.

Anine Kruger, Machdel Matthee and Marita Turpin. Information Systems as creative products: What are industry's expectations?

Abstract: This study presents the first step in exploring the match between IT managers' expectations of functional creativity within an information system, and the functional creativity of the information systems developed by final-year undergraduate IS students. The study commences by exploring the value of functional creativity in information systems. An appropriate means to assess functional creativity in information systems is sought. The Creative Product Assessment Model (CPAM) is accordingly motivated and presented. The CPAM is used as a means to elicit the expectations that IT managers in various IT industry sectors have of functional creativity within information systems. The CPAM also forms the roadmap for Phase 2 of the study where final-year IS student projects will be evaluated for functional creativity by the same IT managers. During Phase 1 of the study it is found that while IT managers value functional creativity in an information system, there are other creativity aspects that are considered to be more valuable. These aspects include the skills to design the creative information systems, the creative design process as well as the eventual end-user experience.

Estelle Taylor and Kobus van Aswegen. Students' approaches to learning: Is it changing?

Abstract: Some researchers are of the opinion that not much has changed in terms of how students prefer to study, whilst others feel that tertiary institutions have to adapt to a new generation of students. The aim of this paper is to compare the learning approaches of students in 2016, with that of students of 10 years ago (2006). This will be done to answer the research question: Are students' approaches to learning changing? Questionnaires were distributed electronically to students in 2006, and again in 2016. It appears that the learning approaches of these students have not changed much. Computer anxiety has increased, as well as the number of students unsure of their own chances of success. There is a decrease in students preferring to solve problems themselves rather than using available solutions and more students prefer to work under supervision. These results show that it is essential that teaching and learning is designed according to the needs of the specific group of students it is meant for, not a mythical "new generation" student.

Nitesh Harry and Sumarie Roodt. The effect of using YouTube in the Classroom for Student Engagement of the Net Generation on an Information Systems course

Abstract: This research paper aims to determine the effect on the engagement of Net Generation learners from using YouTube in the classroom. The education and engagement of the Net Generation learners are a growing challenge among institutions of higher learning. Net Generation arrivals overlaps with the advent of digital technology. Thus, this explains why the learners have dissimilar styles of learning due to their comfort with and use of digital technology. Literature on educating and engaging the Net Generation asserts incorporating the Web 2.0 elements; YouTube inside and/or outside classroom. The target sample includes Net Generation learners in their 2nd/3rd year at the University of Cape Town in the Commerce faculty enrolled in an Information Systems course. The research instrument used was a questionnaire. Two samples were included, the first consisted of learners currently enrolled in the course and the other, learners previously enrolled in the course. The result shows that the use of YouTube had a positive effect on the engagement of Net Generation learners.

Roelien Goede. A critical systems perspective on project-based learning: Guidelines for using industry data for BI student projects

Abstract: Higher education institutions strive to provide the information technology industry with professionals who are good problem solvers and self-directed life-long learners with a firm theoretical knowledge base. To achieve this goal, project-based learning is used as teaching approach. Data and requirements are supplied by an industry partner to fourth year information technology students enrolled for a module in business intelligence. The aim of the collaboration is to develop the students to the best of their ability, while generating benefits to the faculty members and the industry partner.

The success of the module in terms of deep learning of module outcomes, relies on the nature of the industry data. There are different perspectives on the nature and the relationship between industry and universities as well as on the nature of project-based learning. In order to understand and identify these different perspectives, critical systems thinking and especially critical systems heuristics were used.

The goal of the paper is to develop guidelines for using industry data for business intelligence students. The guidelines are initially compiled from literature on business intelligence and project-based learning. They are then refined from experiences of using a specific data set from a project-based learning perspective. An interpretive empirical investigation aimed at understanding different perspectives was conducted in accordance to the characteristics of critical systems heuristics.

Final guidelines are created by enhancing the initial guidelines with the results of the empirical investigation. The resulting guidelines support the interests of the students in terms of real world experience; faculty members in terms of support of academic outcomes; and the industry partner in terms of privacy of sensitive information.

Leila Goosen and Ronell van der Merwe. Keeping ICTs in Education Community Engagement Relevant: Infinite possibilities?

Abstract: The motivation for this investigation relates to objectives regarding the extent to which sample schools are characterized as institutions that exhibit student characteristics regarding utilizing Information and Communication Technologies (ICTs) to enhance learning; qualified and competent leaders using ICTs for the planning, management and administration of their educational environments; the adoption of ICTs by qualified and competent educators, who use these to enhance teaching and learning; access to ICT resources supporting curriculum delivery; connections to ICT infrastructure and their communities. The study locates ICTs, e- schools and what it means to be ICT capable in a theoretical conceptual framework. The study draws on the latest, most relevant research, in a literature review investigating e-schools regarding implementation progress made towards achieving the policy goal of the White Paper on e-education. Strategic objectives to structure implementation of this policy goal are also presented. A section follows explaining how the empirical research was undertaken, illustrating the research methods used in the mainly quantitative research design. A discussion of the findings provides details regarding the sample of 43 respondents from South African schools. The value of this study's findings is illustrated regarding filling gaps identified in literature, a summary of which concludes the paper.

Hussein Suleman, Stephan Jamieson and Maria Keet. Testing Test-Driven Development

Abstract: Test-driven development is often taught as a software engineering technique in an advanced course rather than a core programming technique taught in an introductory course. As a result, student programmers resist changing their habits and seldom switch over to designing of tests before code. This paper reports on the early stages of an experimental intervention to teach test-driven development in an introductory programming course, with the expectation that earlier incorporation of this concept will improve acceptance. Incorporation into an introductory course, with large numbers of students, means that mechanisms needed to be put into place to enable automation, essentially to test the test-driven development. Initial results from a pilot study have surfaced numerous lessons and challenges, especially related to mixed reactions from students and the limitations of existing automation approaches.

H.W. Pretorius and M.J. Hattingh. Reflections of summer school learners on the factors in-fluencing their poor performance in Systems Analysis and Design Course

Abstract: Educators in Higher Educational Institutions (HEI) are under constant pressure to improve their educational practices such as teaching. This study investigates possible causes for first year System Analysis and Design (SAD) course failures of Information System (IS) learners. First year failure is a significant contributor to non-progression statistics, which is a major concern for universities and educational practices which includes academic and non-academic factors.

To conduct the research, an interpretive research paradigm approach was followed, using qualitative methods. Data was collected from 29 Information System learners (male and female) who failed the first year SAD course at a prominent university in the Gauteng Province of South Africa. The constant comparative method was used to code data (responses articulated into questionnaires) into themes as the emerged. The data indicated that the respondents identified factors related to the educators, their chosen pedagogy, their confusion regarding the course content and lack of self-responsibility on their part as contributors to poor performance. The experiences and insights of respondents were used to make recommendations to educators to improve their educational practices, especially first year educators.

Andre Calitz, Jean Greyling and Margaret Cullen. Industry vs Post-Graduate Studies: CS and IS Alumni Perceptions

Abstract: Students graduating from Computer Science (CS) and Information Systems (IS) departments are faced with the question: Should I go and work in industry or should I complete post-graduate studies? The IT industry is currently experiencing a severe skill shortage and recruitment agencies and websites are advertising posts for more than 10000 IT job vacancies in South Africa. Recruitment agencies and industry representatives encourage IT graduates to accept positions in industry in order to gain "on-the-job" work experience. Academics are encouraging graduates to continue their studies and complete post-graduate qualifications to make them more marketable, attain a more diverse IT skill set and earn higher salaries in the future. An increased amount of post-graduate students is further also important for research in the CS/IS discipline.

This exploratory study focuses on CS and IS graduates and post-graduates' (Alumni) perceptions and their opinion regarding the decision to work in industry or complete post-graduate studies. The advantages of deciding to go and work in industry were compared to completing post-graduate studies. Studying a Master's degree in CS/IS versus studying an MBA were also investigated. This qualitative

research study amongst CS/IS Alumni (n=111) provided insight into the reasons students chose to go and work in industry or complete their post-graduate studies. Based on the results, 3rd year students are generally advised to complete at least a 4-year degree or Honours degree. The decision to continue with a CS/IS Masters or an MBA, followed by a PhD, depends on the individual's career plan to either stay in a technical environment, academia or to move into managerial positions. This research could assist academic IT departments in providing relevant advice to graduates completing their 3rd year students from a perspective of a graduate working in industry.

Wai Sze Leung. Cheap Latex, High-End Thrills: A Fantasy Exercise in Search and Seizure

Abstract: Quite often computer forensics students are afforded little or no opportunity to gain experience when it comes to retrieving digital evidence from a crime scene. For our offering, the exclusion was attributed to a lack of resources in terms of time, physical space, and equipment. To circumvent the aforementioned shortcomings, we tapped into the imaginations of our students to introduce a search and seizure exercise by means of roleplay simulation, which is seen to be a form of active learning. This paper describes how we adapted the problem-based learning activities approach employed by the Universities of Sunderland and Glamorgan, to offer our students a similar student-centered learning opportunity, supplementing the lack of a professional crime scene environment with a storyteller to fill in the details of what the students experience during their role-play simulation. We explain how we set up this storyteller-managed approach in implementing the exercise and how the requirements are conveyed to the students. We re-view our version of the search and seizure role-play simulation against the les-sons learnt with the Sunderland and Glamorgan ones, examining how future implementations can be revised to reflect best practices in offering our students improved role-playing simulation experiences.

Romeo Botes and Imelda Smit. The infinity approach: a case study

Abstract: As fourth year Information Technology students make the transition from under-graduate studies towards a post-graduate degree, an honors research project may be a daunting task. At the Z Campus of the XYZ University a series of lectures are offered in an attempt to guide students to make this transition. Students are firstly introduced to research paradigms and then each of the four prevalent in-formation systems paradigms is focused on to explain the methodology and methods applicable, and how it may be implemented. Each lecture in the lecture series is accompanied by an assignment. Although many obstacles present themselves in the completion of the honors projects, a prominent one was identified by the researchers. It concerns the interpretivist paradigm and specifically the collection of qualitative data with its accompanying data analysis. To assist students the analogy of the infinity symbol with an example were used to explain the concept. Important focuses of the explanation include the crossroad faced by researchers when they reach a point of saturation in a study – to gain insight into a phenomenon, and the implementation of the methodology in collaboration with the selected method to ensure the validity of the research. The article addressing these issues named the suggested process the infinity approach. This paper builds on this earlier research on the infinity approach by means of a small case study forming a pilot evaluation.

Part 2: Full-length papers

Cybercitizenship Awareness Module Designed for First Year University Students

Maureen van den Bergh[1] and Erica Pretorius[1]

[1] University of Johannesburg, Johannesburg, South Africa
maureenvdb@uj.ac.za
ericap@uj.ac.za

Abstract. These days, ordinary people have become cybercitizens, or inhabitants of cyberspace, and they are more connected than ever. Unfortunately with this connectedness, comes serious cybersecurity risks. It is therefore important that as cybercitizens, people follow proper Cybercitizenship when online, and develop good cybersecurity habits. Especially digital natives such as university students, who grew up with technology, and are comfortable using it in most aspects of their lives. They are deemed heavy users, but seem to ignore basic information security, which result in 40% of them experiencing cybercrime [1]. For this reason, the Center for Academic Technologies at the University of Johannesburg designed an online Cybercitizenship module (UJCYBER), aimed at first year students. It is an awareness program with the objective to support students in their journey of understanding, appreciating and reacting to the dangers that loom in cyberspace, in order to proactively create a positive cyber footprint. This paper presents the UJCYBER awareness module.

Keywords: Cybercitizenship; Netiquette; Cybersecurity; Cybercrime; Cyberattacks; Cyber Activities; Awareness training; UJCYBER.

1 Introduction

As the Internet became part of the everyday life of the average person, it is used for general browsing, communication, online banking, entertainment, travel bookings, and to find information about goods and services [2, 3]. As of 31 March 2017, there are an estimated 3,731,973,423 Internet users in the world, which equates to a 49.6% Internet penetration into the population [4].

Nowadays, social media are also integrated into society and in everyday things [5, 6]. As of January 2017, social media penetration globally is at 37%, which is a total of 2.789 billion social media users. For the time period, January 2016 to January 2017, South Africa showed a 15% increase in the number of persons using social media [7, 8].

We are globally more connected than ever [6, 9], with the Internet taking residence in both our private and professional lives [10]. Unfortunately this connectedness brings with it serious cyber security risks, because of increased opportunities to exploit vulnerabilities [10-12]. It is therefore important that individuals are educated on how to be responsible cyber citizens or Netizens. As echoed in a statement by Michael Kaiser, the executive director of the National Cyber Security Alliance (NCSA): "As every one of us, our families and our communities become increasingly connected, it becomes even more critical to practice good cybersecurity habits [13]. Also, properly educating users to take the responsibility to safeguard themselves in cyberspace, is essential. As reiterated by Robert Dix, the vice president of Juniper Networks responsible for global government affairs and public policy:

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

“Currently, there is a gap, primarily because users simply don't understand what the cyber threats are, how their information can be compromised or what to do. We can and must change that” [14].

Netizens or Net Citizens, are Internet users that “exist as a citizen of the world.” Netizens are referred to as residents of the Internet, participants of the cyber community, or occupants of cyberspace [15, 16]. Digital natives (also referred to as the ‘Net Generation’, the ‘Y Generation’ and ‘Millennials’), are young people (generation Y, born in 1977-1994; and Generation Z, born in 1995-2012) who grew up with technology. Digital natives are considered comfortable using the Internet in all aspects of their lives. Entrenched in the digital age, young people use computers, cell phones, listen to music online, play video games, and perform numerous other activities [11, 17, 18]. University students are generally deemed heavy users, especially of social media [5]. The United Nations’ International Telecommunication Union [19] report, indicate that while new Netizens .join the Internet on a daily basis, they do so with inadequate knowledge of security threats and the consequences of their online activities.

While the Internet and social media have many positive features to offer, the negative aspect with regards to cyber security risks is unfortunately also present [20]. 35% of younger users seem to ignore security aspect of cyberspace, which result in 40% of them experiencing cybercrime [1]. It is these “native speakers of the digital language” that are entering universities [11, 17, 18]. Consequently, it is important that universities properly educate students on how to be responsible cyber citizens, as echoed by Simon Kouttis of Stott and May: “The authorities and schools must do their part to foster large-scale awareness about cyber-security. And they should integrate it into the core curriculum as soon as students start using computing devices, rather than leaving it to degree level courses.”

This paper introduces the University of Johannesburg’s (UJ’s) online Cybercitizenship module (UJCYBER), as an approach to create awareness and educate students about proper Cybercitizenship and cyber security. The objective is to support students in their journey of understanding, appreciating and reacting to the dangers that loom in cyberspace, so that they may proactively create a positive cyber footprint.

But first, we present background information in Section 2, about what it means to be a responsible cyber citizen, by following proper Cybercitizenship and netiquette when online. We also discuss the necessity to develop good cyber security habits. This is followed by introducing the concept of cybercrime. A discussion of different types of cyberattacks, specifically aimed at the individual, and some cyber activities that exploit the user, and some cyber activities that the user themselves may indulge in. Section 2, ends with a discussion of awareness training, with an emphasis on providing awareness training that require students to critically think about appropriate topics covered, and apply security concepts. The methodology followed for this paper, is presented in Section 3, with Section 4 presenting the results, which manifests as UJ’s online Cybercitizenship module or UJCYBER. Concluding the paper with the discussion and conclusion, which is Section 5 and Section 6 respectively.

2 Background

To be a responsible cyber citizen, the individual should follow proper Cybercitizenship and netiquette when online, and develop good cyber security habits.

2.1 Cybercitizenship and Netiquette

A Netizen or Net Citizen, is an Internet user that “exist as a citizen of the world.” This user is referred to as a resident of the Internet, a participant of the cyber community, or an occupant of cyberspace [15, 16].

Cybercitizenship is when certain rules and conventions are followed by netizens, and computers are used in an ethical way. In being a good cybercitizen, the individual shows good Cybercitizenship by following some of the rules of the Internet [21].

Based on the Internet Advisory Board’s memo titled “Ethics and the Internet” [22], the Computer Ethics Institute created the “Ten commandments of computer ethics” [23], as a guideline to educate people on how to ethically use computers.

The Ten Commandments are:

1. Thou shalt not use a computer to harm other people.
2. Thou shalt not interfere with other people's computer work.
3. Thou shalt not snoop around in other people's computer files.
4. Thou shalt not use a computer to steal.
5. Thou shalt not use a computer to bear false witness.
6. Thou shalt not copy or use proprietary software for which you have not paid.

7. Thou shalt not use other people's computer resources without authorization or proper compensation.
8. Thou shalt not appropriate other people's intellectual output.
9. Thou shalt think about the social consequences of the program you are writing or the system you are designing.
10. Thou shalt always use a computer in ways that ensure consideration and respect for your fellow humans.

In relation to Cybercitizenship, netiquette is defined as being courteous and respectful when interacting online with other netizens. Netiquette is used to describe communication that takes place over the Internet, and is a combination of the words "network" and "etiquette". For example, when sending any comments or messages, be sure to reread the communication to ensure it may not be interpreted incorrectly, because messages (text, e-mail or instant) do not have the luxury of non-verbal communication to help with interpreting the intent of the message [21, 24, 25].

Virginia Shea's e-book titled "Netiquette" [26], has become "the" source for online manners [24]. In her book she describes ten core rules of netiquette. These rules serves as general guidelines of how to behave properly online, and serves both new and advanced users. Shea [26], states that the premise of the rules are to help with basic principles and help solve netiquette predicaments.

The ten core rules are:

1. Remember to be human.
2. Adhere to the same standards of behavior online, that you follow in real life.
3. Know where you are in cyberspace.
4. Respect other people's time and bandwidth.
5. Make yourself look good online.
6. Share expert knowledge.
7. Help keep flame wars under control.
8. Respect other people's privacy.
9. Don't abuse your power.
10. Be forgiving of other people's mistakes.

Good Cybercitizenship and netiquette helps the individual to possibly avoid irreversible mistakes and potentially not commit perceived acts of disrespect. Mistakes sent via email, posts on social media, or the Web, for which there may not be a way of removing the communication. Perceived acts of disrespect by others using the Internet or social media, because the individual is not following the unwritten rulesets [24, 25].

2.2 Cybersecurity

Information security includes protecting and maintaining the confidentiality, integrity and availability (CIA Model) of information and information infrastructures [9]. Cyber security, which falls within information security, includes protecting specifically the information infrastructures from criminal activities or cybercrimes. Information infrastructures include the Internet, telecommunication systems, computer systems, and industrial control systems [9]. Information security and cyber security both also include the human element. Information security is concerned with the role the human element plays in the security process. Cyber security is concerned with the human element as a potential target of cybercrimes, or even their potential involvement, albeit unknowingly, in cyberattacks [27].

As potential targets of cybercrime, it is necessary that cybercitizens practice good cybersecurity to protect themselves. Gelman and McCandlish [28], present a guide for protecting oneself when online. The guide suggests sidestepping spam or junk mail, recognizing online scams, identifying hoaxes, protecting yourself from identity theft, guarding against email privacy fraud, protecting against viruses, protecting intellectual property, and keeping your children safe. Andert and Burleson's [29] safety handbook, suggest protecting oneself by protecting your privacy when online, knowing the type of personalities, weirdos, and stalkers that are online, protecting against cybercrime and cyber theft (plagiarism), identifying scams (phishing scams, mouse trapping, hoaxes), recognizing and preventing identity theft, online security measures that can be taken, Internet spies (spyware and malware), child protection, virus attacks, and spam.

In an effort to protect themselves against cybersecurity activities, cybercitizens should understand the environment of cybercrime, and the types of attacks used by cybercriminals.

2.3 Cybercrime

A universally agreed definition for the term Cybercrime does not exist, therefore there are many varied definitions as interpreted by individuals and groups [30]. In South Africa the National Cybersecurity Policy Framework defines cybercrime as "…illegal acts, the commission of which involves the use of information and communication technologies" [31]. Cybercrime as defined by the South African Cyber Threat Barometer [9], is a crime that happens in cyberspace. The terminology refers to the use of computer technology to commit a crime or computer technology that was in some manner involved when a crime took place. Another definition, as proposed by the Electronic Communications and Transactions Amendment Bill of South Africa, and adopted by this paper, defines cybercrime as "any criminal or other offence that is facilitated by or involves the use of electronic communications or information systems, including any device or the Internet or any one or more of them" [32].

Even though there are many ways in which the individual could protect themselves when online, individuals could still fall victim to clever scams, shrewd social engineering efforts, cyberstalking, cyberbullying, undetected hacking efforts, and an array of other cybercrime activities [33].

Cyber criminals use the Internet to steal, manipulate, disrupt, destroy, and deny access to information. Then also to change the context in which the information is viewed in, in order to change the perceptions of users towards the information [9]. Cyber criminals do this via a variety of, but not limited to, cyber activities such as, identity theft, distributed denial of services (DDoS) attacks, botnets, malware especially phishing attacks, spam, ransomware, and social engineering [34-37].

2.4 Cyberattacks

Identity Theft. Prior to technological advancement, identity theft involved "donning a wig and masquerading as someone else" [38]. Nowadays identity theft is still defined as someone pretending to be someone else, it is just that now it is done online via the Internet. This type of online theft is referred to as Online Identity Theft. The aim of the pretender is to gain financially, by obtaining and using credit card information, identification numbers, addresses, etc. of the victim [37].

Malware: Spam, Phishing and Ransomware. Malicious code or rogue program is created by a threat agent, with the intent to cause disruption by stealing, terminating, and denying access to information systems and assets. Malicious code cause programs or parts of programs to perform in an unexpected way, and compromise a program's confidentiality, integrity and availability. Before execution it must be triggered by some form of action before infection takes place. It is usually disseminated via social media, e-mails, websites, USB flash drives etc. [34, 39].

*Spam***.** A very common type of cybercrime, and unfortunately not much can be done to keep them at bay. Spam is when the user is sent unsolicited commercial emails and messages. Much the same as traditional junk mail, just now it is in an electronic format. These messages are designed to lure users to click on the links within the emails and reveal some important information about themselves [37, 39].

Phishing. Another common type of cybercrime that uses email or imitation websites to send out misleading messages. Messages that initially look like they come from a legitimate source in order to trick the end user into providing information such as account numbers, pin codes, identity numbers, authentication credential etc. [34, 39].

Ransomware. The minute ransomware is activated, it encrypts the user's files, rendering their computer useless. The user's computer is held captive, and the user is informed via a warning message displayed on their screen. The message usually demands a payment in order for the files and thus the computer to be released [37].

Malware: Other Types. Below is a description of a few of the various types of malware:

Virus. A malicious program that changes other non-malicious programs by transferring malicious code to the other program. The virus attaches itself to the non-malicious program, with the aim of either terminating the original program, or to "piggy-back" on the original program [40].

Trojan horse. This type of malware is sent to a user, and arrives as a misleading program which appears as benign. It disguises itself as being legitimate, but contains malicious codes that may capture the user credentials of users [40].

Worms. This malicious program has the ability to replicate itself to other computers connected to a network. It differs from a virus via its medium of transfer. A worm transfers through networks, while a virus is able to transfer via any medium [40].

Spyware. This type of malware, tracks a user's digital activities. Tracking generally happens without the user being aware of this or having given permission. Regularly spyware is bundled with legitimate programs or camouflaged as legitimate software. When spyware goes undetected, it may limit bandwidth, reduce CPU power, obtain personal information, and generally frustrate users with multiple pop-up messages [41, 42].

Social Engineering. Social engineering is a broad term that defines how cyber criminals use different social skills to obtain information from a person via psychological manipulation. Cyber attackers uses social engineering to deceive people [36, 39], by gaining the person's confidence, in order to obtain the information they need. Information such as personal information, access credentials, company information or anything that may be of interest to the cyber criminals [34, 37].

Some types of social engineering attacks include:

Supposed Email from a Legitimate Friend. If a cybercriminal gained access to a friend's email account, the cybercriminal has access to all the contacts of this person. The criminal then may use the legitimate account to send out fake messages, urgently asking for help, or asking for a donation to a charity. The message may also provide a URL to click on, about something you "just have to see". Other times the message contains photos, music or videos, which contains malicious code. The victim unsuspectedly clicks the links, send a donation or download the attachments, because the email is from a friend [43].

Phishing Attempts. The victim is contacted by, what seems like, a legitimate organisation that is well known. The message format could be an email, instant message or text message that presents some kind of scenario. The scenario could be something like the victim has won some prize and they just need to supply banking credentials, their address, telephone number, and sometimes verify their person via their social security / identification number. Another scenario could be that a new form needs to be completed to verify the person and their details again to the organisation, because of some change that took place, and failure to do so will be detrimental to the person. A third scenario prey on the kindness of the victim, and asks for financial support for some worthy campaign [43].

Baiting Scenarios. They entice the individual towards something they want, such as downloading new music or movies, but the malicious websites only infect the victim's computer [43].

Quid Pro Quo Initiative. Quid pro quo is defined as "something for something." It involves the attacker phoning individuals in an organisation, pretending to be from technical support and offering help. Eventually the attacker will find an individual with a problem and "help" the victim by requesting commands to be entered and information given to help "solve" their problem, but actually the victim is giving the attacker access to launch malware.

Pretexting. An invented scenario or a pretext that is used to engross a victim with the story, in such a manner that intensifies the chance of the victim revealing some personal information or perform some act that they would not do under ordinary circumstances.

2.5 Cyber Activities

Cyberbullying. As agreed by researchers, cyberbullying is a form of hostility, practiced online via the Internet [44]. Smith, Mahdavi [45] define cyberbullying as "an aggressive, intentional act carried out by a group or individual, using electronic forms of contact, repeatedly and over time against a victim who cannot easily defend him- or herself". In addition, [46] explain that for cyberbullying to be understood as "true" cyberbullying, the intent of the perpetrator should be to hurt the other person, the victim should perceive the bullying as hurtful, the undesirable actions should be repeated, and the relationship between the perpetrator and victim considered as unequal.

Cyberstalking. In general, cyberstalking or online-based intrusion, is defined in terms of descriptions directly related to ordinary stalking [12]. Most psychiatric studies agree that stalking is when there is repeated intrusion into the life of an individual, by an offender [47]. This intrusion becomes abnormal when there are no less than 10 instances or when intrusion takes place over a period of one month [48]. This repeated intrusion causes the receiver much distress and feelings of danger [49]. A study by al-Khateeb, Epiphaniou [12], determined that offenders mostly use communication channels such as e-mails, mobile messages, and mobile call to exercise their online-based intrusions.

Sexting. Amongst adolescents (12 to 17 years of age), sexting is a regular occurrence. Sexting takes place by exchanging sexual explicit content, such as photos and videos, via text messages, using mobile devices and over the Internet [50-52].

Digital Piracy. Piracy, as in software piracy, is "the unauthorized use or reproduction of another's work" [53], by illegally copying, disseminating, and using the software [54]. Digital piracy, also known as Internet-based piracy or online piracy [55], targets content such as movies, music, games, books and television programs [56]. Many people, including young people, do not perceive online piracy as illegal [56, 57].

2.6 Awareness Training

Critical to an organisation's overall information security plan, is also ensuring that employees' information security weaknesses are addressed via increased information security awareness [58]. There are a number of methods that could be used to create awareness about information security. Some methods include the circulation of information security related messages via visible brochures, posters, flyers, banners, and trinkets (mouse pads, pens, coffee cups, and t-shirts). Other messages are in electronic format and include messages conveyed via e-mails, corporate intranets, newsletters, bulletin boards, screen savers, and videos. Formal discussions, meetings, and training (in various formats), lectures, and conferences are also used to transfer information related to information security [58, 59]. Of the methods mentioned, this study focuses on training as the preferred method, in the format of an online module.

A study by Abawajy [58], determined that security awareness delivery that incorporates text-, game-, and video-based methods, is more successful than using only one type of delivery method. Shaw, Chen [60], confirm that, as much as content is important, critical to the success of any security awareness program, is the delivery methods used. Numerous training efforts are unsuccessful, because of repetition, and a lack of involvement from the participant to critically think and apply the concepts learned [61]. Kritzinger and von Solms's [62] E-Awareness Model (E-AM), proposes a model that not only contains a component to create awareness about the risks of cyberspace, but also a second component that ensures the user absorbs the content from the first component.

Cyberattacks on unsuspecting users are increasing in number and frequency [58], while users are unaware or just "don't understand what the cyber threats are" [14], their information is compromised.

3 Methodology

The literature search include multiple databases, such as Science Direct, Digital Library, EBSCO, Google Scholar, and UJoogle (UJ's library resource). To build the whole picture, several topics were researched, which included major keywords such as Cybercitizenship, netiquette, cybersecurity, cybercrime, cyberattacks, cyber activities, and awareness training. Only articles published in English were included.

Because the emphasis of the study is on supporting students in their journey of understanding, appreciating and reacting to the dangers that loom in cyberspace, it was determined from literature that starting the journey towards awareness, an interactive awareness module is most appropriate. An awareness module that students could follow at their own pace, share their understanding of concepts (via online journals, comic strips, and online exercises), have discussions with other students (via blogs, wikis, and discussion forums), and give feedback (via an online survey) about the module.

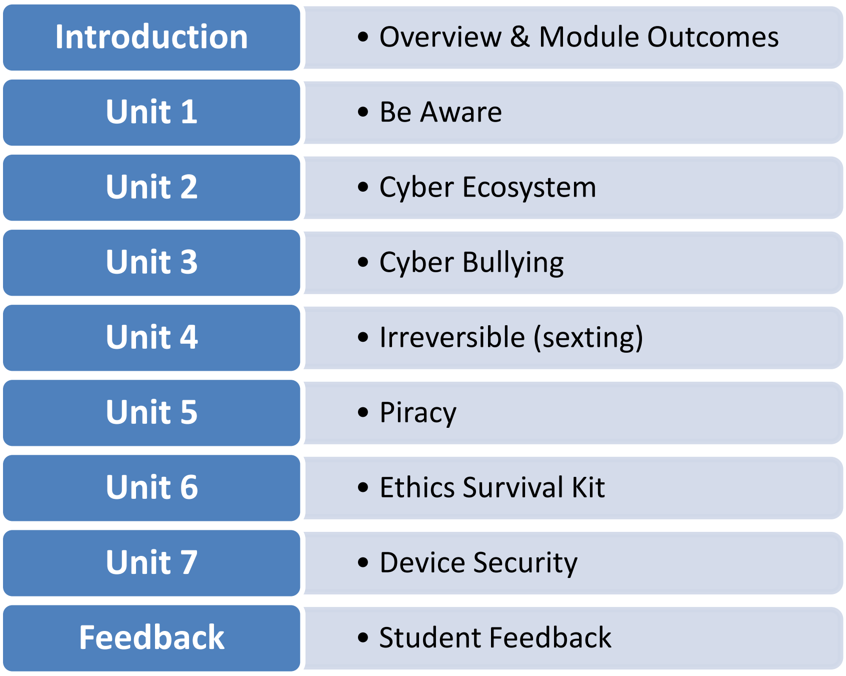
Next, in determining the content of the online module, the focus is on the individual, and the concepts that is important for a netizen to be aware of. Therefore, concepts such as Cybercitizenship, netiquette, and cyber security habits are discussed in the module, to create awareness of what it means to be a proper cybercitizen with good cyber security habits. Also, the concepts of cybercrime, cyberattacks, and cyber activities that specifically are aimed at the individual, are discussed. This aims to create awareness of the types of exploits used by perpetrators and illegal activities that the individual themselves may conduct (whether knowingly or not).

All of the above information was presented and discussed with experts at UJ's Academy of Computer Science and Software Engineering. This helped to identify the most common issues that students struggle with, in cyberspace. It was concluded that, for now, that the module will be directed at first year students, and that the module should also include "Know your rights" with references to South African law. The result is the UJCYBER online module.

4 Results: UJ's Cybercitizenship Online Module (UJCYBER)

The Centre for Academic Technologies at the University of Johannesburg created the Cybercitizenship online module, as an awareness program with the objective to support students in their journey of understanding, appreciating and reacting to the dangers that loom in cyberspace, in order to proactively create a positive cyber footprint.

The module starts with an overview and explain the module outcomes. Units 1 to 7 then discuss content such as awareness, the cyber ecosystem, cyber bullying, sexting, piracy, ethics, and device security (find most URLs to content in Appendix) The module concludes with a space for students to give feedback (See Fig. 1).



Introduction	• Overview & Module Outcomes
Unit 1	• Be Aware
Unit 2	• Cyber Ecosystem
Unit 3	• Cyber Bullying
Unit 4	• Irreversible (sexting)
Unit 5	• Piracy
Unit 6	• Ethics Survival Kit
Unit 7	• Device Security
Feedback	• Student Feedback

Fig. 1. UJCYBER Online Module Framework

4.1 UJCYBER Introduction

To begin with, the overview explains to the student that this module aims to support them in their journey of understanding, appreciating and reacting to the dangers that loom in cyberspace, so that they may proactively create a positive cyber footprint. The introduction also explains to the student what they should be able to do, after completing the module.

Which is:

- Describe the extent of possible threats looming in cyberspace,
- Explain the interaction between a human and technology in a virtual environment,
- Identify what constitutes inappropriate online behaviour,
- Establish how online conduct could lead to unforeseen consequences and realize what the possible drawbacks could be,
- Critique and debate the ethical dilemma with regard to justifying piracy (copyright infringement, illegal downloads),
- Summarize the basic cyber ethics they discovered during this unit,
- Devise a list of possible strategies and techniques to prevent falling victim to possible online attacks.

4.2 UJCYBER Unit 1: Be Aware

Unit 1 creates awareness about cyber security. A video about cyber security explains what could possibly happen when someone posts information such as a photo on their social media account, and how easy it is for cyber criminals to obtain their information to possibly use it in identity theft, how easily malware could be distributed to themselves and their friend list, and the interception of private messages. It includes a blog where students may post their own, or reply to others' comments about this video.

A web link to an online article discusses which new WhatsApp scams to watch out for in South Africa. It includes a discussion forum for students to post their views about WhatsApp scams.

An assignment type exercise tasks students to search the Web and find links about cyber security issues, adding the links and a short description to a Microsoft Word document, and submitting it as an assignment.

A private space, in the format of an online journal, is available to students to describe their actions when using the Internet. Student are encouraged to reflect on their online behavior and write in their journal. They can describe their actions/ interactivity on the Internet that could make them vulnerable to attacks or threats. Students are also encouraged to suggest any aspect they could improve when using the Internet.

A section on connectivity in cyberspace allows students to explain in their own words their understanding of connectivity threats in cyberspace, after a live chat with a friend. The student also investigates what hackers do and why or how the Central Intelligence Agency (CIA) intercept it, by creating a comic strip, storyboard or mind map to demonstrate possible threats related to communicating online.

Unit 1 concludes with links to South Africa's Electronic communications and Transactions Act of 2002, with specific reference to "Scope of protection of personal information" and "Principles for electronically collecting personal information." As well as an article about cybercrime in South Africa.

After unit 1 created an awareness of cyber security, unit 2 explains the cyber ecosystem.

4.3 UJCYBER Unit 2: Cyber Ecosystem

In explaining the cyber ecosystem, unit 2 introduces the student to cyber buzz words via an online video. After watching the video the student creates a blog entry with their own definition of what they think the concept of cyberspace is. Following a formal definition, the student updates a Wiki content page by completing a sentence to create their own formal definition of a cyber ecosystem. Further to this, an online test is available to assess the student's knowledge about cyber devices.

Web links to articles, teaches the student about differentiating between a web browser and search engine, as well as reviews about the most popular web browsers and search engines. Another online test, question students on their new found knowledge about web browsers and search engines. This then concludes unit 2.

Now the student has a better understanding about the cyber ecosystem, the concept of cyber bullying is introduced to the student in unit 3.

4.4 UJCYBER Unit 3: Cyber Bullying

Cyber bullying is introduced to the student via a video of other students reading aloud mean tweets that they received. This is followed by a formal definition of cyber bullying.

Continuing from the formal definition, students are presented with different scenarios depicting someone that is busy with cyber bullying. The students are asked to discuss the scenarios with a group of friends, to obtain different perspectives. A personal and private journal also allows the student to comment on the scenarios, and to voice their own cyber bullying experiences (if any).

A forum is also readily available for the student to give reasons why someone would participate in cyber bullying, and to suggest solutions about cyber bullying.

A link to South Africa's Protection from Harassment Act familiarizes the student with this act, and extracts explains to the student that defamatory remarks in chat rooms, on social networking sites, e-mails, text messages or instant messages to third parties are some of the methods of committing cyber bullying that fall within the ambit of this criminal offence.

Multiple reads about real life cyber bullying stories, show the student the serious consequences of cyber bullying.

Now the student is presented to the concept of netiquette, the rules of netiquette, and how to be safe in cyber space.

Unit 3 wraps up the concept of cyber bullying with reference to South Africa's Film and Publications Act 65 of 1996, stipulating that the distribution of hate speech is a criminal offence in that "Any person who knowingly distributes a publication which, judged within the context, advocates hatred that is based on race, ethnicity, gender or religion, and which constitutes incitement to cause harm, shall be guilty of an offence".

Unit 3 provides a comprehensive explanation of cyber bullying and the grave consequences of such acts, which leads to another serious concept, that of sexting. Unit 4 elaborates on sexting.

4.5 UJCYBER Unit 4: Irreversible (Sexting)

Sexting and the dangers thereof are discussed in unit 4. An introduction video shows the student just how easy it is to be sexting, without actively thinking about it. A second video explains how an innocent photo could be easily manipulated and quickly distributed on the Internet to total strangers, especially when you haven't set your security settings correctly. How this manipulated photo persistently going to re-appear on the Internet, possibly to be seen by a potential employer, or cybercriminal using it for identity theft.

The rest of unit 4 is dedicated to explaining Section 27 of South Africa's Films and Publications Act that specifies "any person who knowingly creates, produces, imports or possesses material (including publications and films) that contains a visual presentation of child pornograph**y** (child being defined as anyone below the age of 18) shall be guilty of an offence." This leads to an explanation of the ramifications of sexting and the national register for sex offenders.

Together with cyber bullying, sexting is one of many issues on the Internet with serious consequences. Another is piracy, and discussed in unit 5.

4.6 UJCYBER Unit 5: Piracy

Unit 5 creates mindfulness about piracy. A video is used to demonstrate how downloading illegal videos, music, software, PC games and e-books is a crime. A blog is available to students to post their own opinions about piracy and the advantages and disadvantages thereof.

Two South African Acts are introduced, namely, the Copyright Act No. 98 OF 1978, and the Electronic Communications and Transactions Act 25 of 2002. Explicit reference is made to the consequences of being found guilty of an offence under the Copyright Act, and the Electronic Communications and Transactions Act, with reference to fraud and forgery.

In addition the student is informed about plagiarism. Specifically UJ's plagiarism policy, UJ's policy on intellectual property, UJ's student regulations, and UJ's regulations for student discipline.

Armed with an awareness of cyber security and knowledge about cyber bullying, sexting, and piracy, unit 6 provides the student with an ethics survival kit.

4.7 UJCYBER Unit 6: Ethics Survival Kit

The survival kit include sections on defining cyber ethics, a video presenting a moral dilemma, and a blog for the student to comment on the video, to give their opinion on the person's ethical (or not) behavior.

Again students are referred to UJ's student regulations and UJ's plagiarism policy, with provides to the student an ethical base and parameters to within as a UJ student.

As much as ethical behavior is important, it is necessary that the student understands the responsibility of protecting their devices, because of others' unethical behaviors.

4.8 UJCYBER Unit 7: Device Security

Unit 7 starts off with a video on the topic of phishing and articles about phishing scams. A video about phishing and spear phishing, explains how it works and how organisations could protect their employees from these attacks. Articles introduces different types of phishing scams and tips on how the student could protect themselves against these types of attacks.

Further, a webpage link explains how the student could protect their android device against attacks. The article provide 10 tips to tighten security on their device. Following the article, the student enters into a group discussion about the security measures they can set up on their device to avoid hacking and interception.

Unit 7 concludes with the South Africa's Electronic Communications and Transactions Act 25 of 2002, as a reminder that "a person who performs any of the acts described in section 86 for the purpose of obtaining any unlawful advantage by using or producing fake data with the intent that it be considered or acted upon as if it were authentic, is guilty of an offence.".

After completing all the units, the student is afforded an opportunity to give their feedback.

4.9 UJCYBER Feedback

Last but not least, the feedback section allows the student the opportunity to provide feedback about the module. The feedback takes the format of an online survey.

5 Discussion

As we are globally more connected than ever [6, 9], with the Internet embedded into everyday life and social media integrated into society, we have become netizens. As echoed by Hauben and Huaben [15], and MacKinnon [16], we are residents of the Internet, participating in the cyber community, occupying cyberspace. But, unfortunately we participate in this cyber community without adequate knowledge of security threats and the consequences of our online actions [19]. This creates serious cyber security risks, because of increased opportunities for cybercriminals to exploit vulnerabilities [10-12]. It is therefore important that individuals are educated on how to be responsible cyber citizens. More so, students who are deemed heavy users [5], who seem to ignore basic information security, which result in many of them experiencing cybercrime [1]. Consequently, it is important that universities properly educate students on how to be responsible cyber citizens. Simon Kouttis of Stott and May says that "…authorities and schools must do their part to foster large-scale awareness about cyber-security. And they should integrate it into the core curriculum as soon as students start using computing devices…"

As a result, UJ's online Cybercitizenship module (UJCYBER) was created to addresses the issue of uninformed students, occupying cyberspace and participating in the cyber community without proper awareness of cyber security. As an interactive module, UJCYBER, which is aimed at first year students, provide them the opportunity to follow an awareness program, share their understanding of concepts (via online journals, comic strips, and online exercises), and have discussions with other students (via blogs, wikis, and discussion forums), while giving feedback (via an online survey) about the module to the creators.

As an interactive module, UJCYBER deliver awareness training by incorporating web links to articles, videos, and students even create a comic strip to demonstrate possible threats related to communicating online. This is consistent with Abawajy's [58] study, which determined that incorporating text-, game-, and video-based methods into a security awareness program, are more successful than using only one type of delivery method.

Further, UJCYBER also engage students by providing blogs where students are able to post their own, or reply to others' comments about cyber security and cyberspace. Students are able to use discussion forums to post their views on topics such as WhatsApp scams and cyberbullying. Online journals, live chats with friends, updates to Wiki content, and group discussions, are all elements that add to the interactive element of the UJCYBER module. Cone et al.[61], stress the fact that numerous training efforts are unsuccessful, because of repetition, and a lack of involvement from the participant.

Online tests and an assignment are more elements included in the UJCYBER module, which according to Kritzinger and von Solms's [62] E-Awareness Model (E-AM), it is important to ensure that the user absorbs content.

Cybersecurity awareness training focusing on both content and engagement, is powerful in its ability to empower students to become aware and informed cybercitizens.

6 Conclusion

In this paper, we studied two main topics, namely cyber concepts that are important for a netizen to be aware of, and how to establish awareness of these concepts. We specifically looked as cyber concepts that have an impact on the individual. Concepts such as Cybercitizenship, netiquette, cyber security habits, cybercrime, cyberattacks, and cyber activities. Our investigation and collaboration with experts suggest, that as a start, these initial concepts are important to include in an awareness program, aimed at first year students. It was also determined that the best way to introduce these concepts and create awareness, was via an online cyber awareness module**.** An awareness module that students could follow at their own pace, share their understanding of concepts with peers, have discussions with other students, and give feedback about the module to the authors. The resultant UJCYBER module offers a foundation for cyber security awareness training, and future work that could expand upon this module, to possibly include other relevant concepts. Further studies could also test behavioral change, by using the module as a tool to educate and then test behavioral changes, after introduction to the module. The module could further be used as a tool to teach (via lecturer-based teaching), lessons cyber security.

References

1. Norton Cyber Security Insights Report. *Norton Cyber Security Insights Report*. 2016 [cited 2017 31 Jan]; Available from: https://www.symantec.com/content/dam/symantec/docs/reports/2016-norton-cyber-security-insights-report.pdf.
2. Schafer, J.A., *Spinning the web of hate: Web-based hate propagation by extremist orgnizations.* Journal of Criminal Justice and Popular Culture, 2002. **9(2)**(2): p. 69-88.
3. Haythornthwaite, C. and B. Wellman, *The Internet in iveryday life: An introduction*. 2002: Blackwell Publishers, Fall 2000.
4. Internet World Stats. *Usage and Population Statistics*. 2017 [cited 2017 26 May]; Available from: http://www.internetworldstats.com/stats1.htm.
5. Lau, W.W.F., *Effects of social media usage and social media multitasking on the academic performance of university students.* Computers in Human Behavior, 2017. **68**: p. pp.286-291.
6. Bertot, J.C., P.T. Jaeger, and D. Hansen, *The impact of policies on government social media usage: Issues, challenges and recommendations.* Government Information Quarterly, 2012. **29**(1).
7. World Wide Worx and Fuseware, *SA Social Media Landscape 2017*. 2017: http://www.worldwideworx.com/wp-content/uploads/2016/09/Social-Media-2017-Executive-Summary.pdf.
8. Kemp, S., *Digital in 2017 Global Overview*. 2017, We Are Social & Hootsuite.
9. The South African Cyber Threat Barometer, *A strategic public-private partnership (PPP) initiative to combat cybercrime in SA*. 2012.
10. Kouttis, S., *Improving security knowledge, skills and safety.* Computer Fraud & Security, 2016. **2016(4)**: p. pp. 12-14.
11. Reid, R. and J. Van Niekerk, *Snakes and ladders for digital natives: information security education for the youth.* Information Management & Computer Security, 2014. **22(2)**: p. pp. 179-190.
12. al-Khateeb, H.M., et al., *Cyberstalking: Investigating formal intervention and the role of Corporate Social Responsibility.* Telematics and Informatics, 2017. **34**: p. pp. 339-349.
13. PR Newswire Association LLC, *National Cyber Security Awareness Month Launches with a Call to Action for All Digital Citizens to STOP. THINK. CONNECT.(TM)*. 2016: Retrieved from http://0-go.galegroup.com.ujlink.uj.ac.za/ps/i.do?p=AONE&sw=w&u=rau_itw&v=2.1&it=r&id=GALE%7CA464987512&sid=ebsco&asid=25e993893a31f2208d47b218f6636545.
14. PR Newswire Association LLC, *AFCEA International Cyber Committee Calls on Government to Teach Citizens How to Protect Cyber Ecosystem*. 2016: http://0-go.galegroup.com.ujlink.uj.ac.za/ps/i.do?p=AONE&u=rau_itw&id=GALE|A441514958&v=2.1&it=r&sid=ebsco&authCount=1#.
15. Hauben, M. and R. Huaben. *Netizens: On the History and Impact of Usenet and the Internet*. 1997 [cited 2017 14 Feb]; Available from: http://www.columbia.edu/~rh120/.
16. MacKinnon, R., *The Netizen.* Development, 2012. **55(2)**: p. pp. 201-204.
17. Prensky, M., *Digital Natives Digital Immigrants.* On the Horizon, 2001. **9(5)**.
18. Kennedy, G.E., et al., Australasian Journal of Educational Technology, 2008. **24(1)**: p. pp. 108-122.
19. International Telecommunication Union, *Measuring the Information Society*. 2013.
20. Popescul, D. and M. Georgescu, *SOCIAL NETWORKS SECURITY IN UNIVERSITIES: CHALLENGES AND SOLUTIONS* Annals of the Alexandru Ioan Cuza University - Economics, 2016. **62**(1): p. pp. 53-63.
21. Nottoli, S. *Cyber Citizenship, Netiquette, and Bullying*. 2013 [cited 2017 20 Feb]; Available from: http://schoolinternetsafety.pbworks.com/w/page/27593131/Cyber%20Citizenship,%20Netiquette,%20and%20Bullying.
22. Internet Activities Board. *Ethics and the Internet*. 1989 [cited 2017 20 Feb]; Available from: https://tools.ietf.org/html/rfc1087.
23. The Computer Ethics Institute. *The Ten Commandments of Computer Ethics*. 1992 [cited 2017 20 Feb]; Available from: http://computerethicsinstitute.org/images/TheTenCommandmentsOfComputerEthics.pdf.
24. Jacob, L.M., *Etiquette in technology*, in *Salem Press Encyclopedia Of Science*. 2016.
25. Kozík, T. and J. Slivová, International Journal of Engineering Pedagogy, 2014. **4(3)**: p. pp. 67-70.
26. Shea, V., *Netiquette*. 1994, Albion Books.
27. von Solms, R. and J. van Niekerk, *From information security to cyber security.* Computers & Security, 2013. **38**: p. 97-102.
28. Gelman, B. and S. McCandlish, *Protecting Yourself Online: The Definitive Resource on Safety, Freedom, and Privacy in Cyberspace*. 1998: Harper San Francisco.
29. Andert, S. and D.K. Burleson, *Web Stalkers: Protect Yourself from Internet Criminals & Psychopaths*. 2005: Rampant TechPress, North Caroline, USA.
30. Internet Safety Campaign Africa. *Safety & Security Guide*. 2016 [cited 2017 24 Jan]; Available from: https://cybercrime.org.za/.
31. State Security Agency, *The National Cybersecurity Policy Framework (NCPF)*. 2012, Government Gazette No. 39475. p. pp. 66-95.
32. Department of Communications, *ELECTRONIC COMMUNICATIONS AND TRANSACTIONS AMENDMENT BILL, 2012*. 2012, STAATSKOERANT, 26 OKTOBER 2012.
33. Cross, M., *Social Media Security: Leveraging Social Networking While Mitigating Risk*. 2014: Elsevier Inc.

34. Bendovschi, A., *Cyber-Attacks – trends, patterns and security countermeasures.* Procedia Economics and Finance, 2015. **28**: p. pp. 24-31.
35. Lilley, C., R. Ball, and H. Vernon, *Experiences of 11-16 year olds on social networking sites*. 2014: London: NSPCC.
36. Brown, C.S.D., *Investigating and Prosecuting Cyber Crime: Forensic Dependencies and Barriers to Justice.* International Journal of Cyber Criminology, 2015. **9**(1): p. pp. 55-119.
37. Khanse, A., *Types of Cybercrime Acts and Preventive Measures*. 2014, The Windows Club.
38. Demshock, H.M., *What Can Be Done to Combat Tax-Related Identity Theft?* Journal of Financial Service Professionals, 2016. **7**(3): p. pp. 16-19.
39. Whitman, M.E. and H.J. Mattord, *Principles of Information Security, International Edition*. Fourth ed. 2012: Course Technology CENGAGE Learning. 619.
40. Pfleeger, C.P., *Security in Computing*. 1997, Prentice-Hall International, Inc. 573.
41. Lee, Y. and K.A. Kozar, *Investigating Factors Affecting the Adoption of Anti-Spyware Systems.* Communications of the ACM, 2005. **48**(8): p. pp. 72-77.
42. Biscontini, T., *Spyware*. 2016, Salem Press Encyclopedia Of Science, Research Starters, EBSCOhost, viewed 2 Feb 2017.
43. Webroot Inc. *What is Social Engineering?* 2017 [cited 2017 2 Feb]; Available from: https://www.webroot.com/za/en/home/resources/tips/online-shopping-banking/secure-what-is-social-engineering.
44. Sun, S., X. Fan, and J. Du, *Cyberbullying Perpetration: A Meta-Analysis of Gender Differences.* International Journal of Internet Science, 2016. **11(1)**: p. pp. 61-81.
45. Smith, P.K., et al., *Cyberbullying: Its nature and impact in secondary school pupils.* Journal of Child Psychology & Psychiatry, 2008. **49**: p. pp.376-385.
46. Vandebosch, H. and K. Van Cleemput, *Defining Cyberbullying: A Qualitative Research into the Perceptions of Youngsters.* CyberPsychology & Behavior, 2008. **11(4)**: p. pp. 499-503.
47. Strawhun, J., N. Adams, and M.T. Huss, *The assessment of cyberstalking: an expanded examination including social networking, attachment, jealousy, and anger in relation to violence and abuse.* Violence and Victims, 2013. **28(4)**: p. pp. 715-730.
48. Pathe, M.T., P.E. Mullen, and R. Purcell, *Same-gender stalking. J. Am. Acad. Psychiatry Law 28 (2), 191–197.* Journal of the American Academy of Psychiatry and the Law, 2000. **28(2)**: p. pp. 191-197.
49. Mullen, P.E., M. Pathé, and R. Purcell, *Stalking: defining and prosecuting a new category of offending.* International Journal of Law and Psychiatry, 2004. **27**: p. pp. 157-169.
50. Kimberly, C., et al., *Young Adult Women, Sexting, and Risky Sexual Behaviors.* Journal of Health Disparities Research and Practice, 2017. **10(1)**: p. pp. 1-8.
51. Mitchell, K.J., et al., *Prevalence and Characteristics of Youth Sexting: A National Study.* Pediatrics, 2011.
52. Strassberg, D.S., et al., *Sexting by high school students: an exploratory and descriptive study.* Arch Sex Behav., 2013. **42(1)**: p. pp. 15-21.
53. Oxford Living Dictionaries. 2017, Oxford University Press.
54. Brunton-Smith, I. and D.J. McCarthy, *Explaining Young People's Involvement in Online Piracy: An Empirical Assessment Using the Offending Crime and Justice Survey in England and Wales.* Victims and Offenders, 2016. **11(4)**: p. pp. 209-533.
55. Danaher, B., M.D. Smith, and R. Telang, *Copyright Enforcement in the Digital Age: Empirical Evidence and Policy Implications.* Communications of the ACM, 2017. **60(2)**: p. pp. 68-75.
56. Warnes, K., *Online piracy*. 2017, Salem Press.
57. Ingram, J. and S. Hinduja, *Neutralizing music piracy: An empirical examination.* Deviant Behavior, 2008. **29(4)**: p. pp. 334-366.
58. Abawajy, J., *User preference of cyber security awareness delivery methods.* Behaviour & Information Technology, 2014. **33(3)**: p. pp. 236-247.
59. Whitman, M.E. and H.J. Mattord, *Management of Information Security. Fourth Edition.* 2013: Course Technology CENGAGE Learning.
60. Shaw, R.S., et al., *The impact of information richness on information security awareness training effectiveness.* Computers & Education, 2009. **52(1)**: p. pp. 92-100.
61. Cone, B.D., et al., *A video game for cyber security training and awareness.* Computers & Security, 2007. **26(1)**: p. pp. 63-72.
62. Kritzinger, E. and S.H. von Solms, *Cyber security for home users: A new way of protection through awareness enforcement.* Computers & Security, 2010. **29(8)**: p. pp. 840-847.

Appendix

UJCYBER UNIT 1

Watch Out for These New WhatsApp Scams in South Africa:
https://mybroadband.co.za/news/security/162206-watch-out-for-these-new-whatsapp-scams-in-south-africa.html

Electronic Communications and Transactions Act, 2002 No. 25 of 2002:
http://internet.org.za/ect_act.html#Scope_of_critical_database_protection and
http://internet.org.za/ect_act.html#Principles_for_electronically_collecting_personal_information

SA Ranks World's Third Highest Cybercrime Victims:
http://businessmediamags.co.za/sa-ranks-worlds-third-highest-cybercrime-victims-2/

UJCYBER UNIT 2

Web Browser vs Web Engine:
http://www.differencebtw.com/difference-between-search-engine-and-browser/

Top 15 Most Popular Web Engines:
http://www.ebizmba.com/articles/search-engines

The Best Internet Browser Software of 2017:
http://www.toptenreviews.com/software/internet/best-internet-browser-software/

UJCYBER UNIT 3

Cyberbullying Scenarios:
http://greenfieldtalkscyberbullying.blogspot.co.za/

South Africa's Protection from Harassment Act:
http://www.justice.gov.za/forms/form_pha.html

Real Life Stories:
http://www.puresight.com/Real-Life-Stories/real-life-stories.html

One in Five SA Teens Cyber Bullied:
http://www.itweb.co.za/index.php?option=com_content&view=article&id=146442

The Girl Who Got Even – A True Cyberbullying Story:
http://choices.scholastic.com/story/girl-who-got-even-true-cyberbullying-story

South Africa: OMG! "OMUHLE" Gela on Cyberbullying:
http://www.africanhitz.com/south-africa-omg-omuhle-gela-on-cyber-bullying/

Celebs Throw Cyber Bullies a Curveball:
http://www.timeslive.co.za/thetimes/2015/10/12/Celebs-throw-cyber-bullies-a-curveball1

Amanda Todd, Cyberbullying, and suicide:
http://cyberbullying.org/amanda-todd-cyberbullying-and-suicide

South Africa's Films and Publications Act:
http://www.gov.za/sites/www.gov.za/files/a65-96.pdf

Be Aware, Be Cybersafe:
http://eagle.unisa.ac.za/elmarie/

UJCYBER UNIT 4

South Africa's Films and Publications Act:
http://www.gov.za/sites/www.gov.za/files/a65-96.pdf

South Africa’s National Register for Sex Offenders (NRSO):
http://www.gov.za/services/national-register-sex-offenders-nrso

UJCYBER UNIT 5

South Africa’s Copyright Act No. 98 OF 1978:
http://www.wipo.int/wipolex/en/text.jsp?file_id=130429

South Africa’s Electronic Communications and Transactions Act 25 of 2002: http://www.gov.za/documents/electronic-communications-and-transactions-act

UJCYBER UNIT 6

Cyber ethics:
http://etec.ctlt.ubc.ca/510wiki/Cyberethics

UJCYBER UNIT 7

Cyber Crime is a growing reality cautions financial coach:
http://www.702.co.za/articles/13945/cyber-crime-is-a-growing-reality-cautions-financial-coach
The Banking Association of South Africa – Phishing Scams:
http://www.banking.org.za/consumer-information/bank-crime/phishing-scams

10 Tips To Tighten Security On Your Android Device:
http://www.hongkiat.com/blog/protect-your-android-device/

South Africa’s Electronic Communications and Transactions Act 25 of 2002: http://www.gov.za/documents/electronic-communications-and-transactions-act

Automata-aided Estimation of Similarity in Novice Programs

Abejide Ade-Ibijola

Formal Structures, Algorithms, and Industrial Applications Research Cluster
School of Consumer Intelligence and Information Systems
University of Johannesburg, Auckland Park Bunting Road Campus
Johannesburg, South Africa
researcher@abejide.com, abejideai@uj.ac.za

Abstract. Source code plagiarism is very common among undergraduate com-puter science students and a lot of research has been carried out on how it can be detected, penalised, controlled or even stopped. In this paper, we propose a new approach to the detection of possibly plagiarised programs written in C++ using deterministic finite automaton (DFA) abstractions. The two programs to be checked for similarity are first normalised, granulated and abstracted to a DFA structure referred to as Single Program Deterministic Finite Automata or SPDFA. Then a newly proposed algorithm is used to map the alphabets of the two SPDFAs. If there is a one-to-one mapping of the symbols in both alpha-bets, we conclude that the programs are totally similar. We have also presented a prototype software application called the Exact Code Matcher that implements this technique as a proof of concept. This detection technique is a new appli-cation of finite automata theories, it is efficient for cloned or lexically altered programs, precise with no false positives and portable across different platforms.

Keywords: Program similarity; Source code plagiarism; Deterministic finite automaton; Regular languages; Automata applications.

1 Introduction

"*Let me have your program, please, I will change all the variable names to something else … the Lecturer wouldn't know…*" This is often the plead of a *parasitic* student to its host when asking to copy their programming assignments, often in introductory programming courses. On one hand, we have Lecturers that are capable of picking-up plagiarised programs if they are to compare two programs, especially when one of the programs is a direct replica of the other. On the other hand, we know that Lecturers do not have the time to compare one-to-n-1 programs (comparing a student's program with that of other students in the class, n being a large class size) or even worse, n-to-n 1 programs (manually comparing every student's programs to every other students' program). This problem has been addressed by many authors using a variety of techniques, especially for detection [9, 10, 16]. From literature, we know that:

1. source code plagiarism is a problem in higher institutions of learning, especially among computer science students [8],
2. the "how" and "why" students plagiarise are known [11, 26, 29],
3. a number of tools exist to detect plagiarised programs [6, 27], among which we can find open source one [4], and
4. more research is needed to find new techniques of detecting plagiarised programs in-order to improve the performance of existing ones as documented by Roy and Cordy [28].

In this paper, we look at the field of Formal Language Theory for a solution to this problem. We have abstracted both the benchmark program (original program) and the test program (student's program in question) to deterministic finite automata (DFAs) each representing a single program. This type of DFA is referred to as a Single Program Deterministic Finite Automata (SPDFA). An SPDFA is constructed by granulating the lines of a program

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

and creating transitions between the lines of the program. The last line of the program leads to a final or an accepting state. Given two SPDFAs, we have presented new algorithms that check if they accept the same string, regardless of the apparent difference in their alphabets. Hence, two programs are inferred to be totally similar if their SPDFAs are the same. Here are the contributions of this paper; we have: proposed a new approach to detecting plagiarised student programs using DFAs, and developed a tool that implements the approach and tested this tool with identical student programs.

This paper is organised as follows. Section 2 discusses the problem statement and ex-plains why this work is worthwhile. Section 3 presents a background to this work with definition of used terms and related work. Section 4 describes the process of trans-forming student programs to strings. Section 5 presents new algorithms for matching program strings. Section 6 introduces a the idea of totally similar programs using the SPDFA abstraction with some test cases and discusses the implied properties of SPDFAs. Section 7 presents more test cases and results, a new tool that detects similar programs using the new technique, and how the new technique performs with respect to existing ones. Section 8 presents conclusions and future work.

2 Problem Statement

Undergraduate students, when given programming assignments, are often expected to write programs that perform the same task. This is known as functionally similar pro-grams [15, 17, 23]. However, most often than not, students involve in copying programs from their peers [8] and even other sources such as: paying someone to do the job [11], or getting online communities to copy from [26, 29]. Hence, maintaining academic integrity and standard by detecting plagiarised programs using software becomes a desire. Many of such tools exist already, but none with a technique based on the theory of finite automata. In this paper, we present a new technique for detecting similar stu-dent programs with main interest in lexically altered programs. The main questions answered in this paper are, how do we:

1. abstract student programs to SPDFAs, a type of DFA?
2. determine if two SPDFAs (representing two different student programs) accept the same program string? — if they do, then they are regarded as totally similar.

We have addressed these questions by designing new algorithms for constructing SPDFAs from student programs, and mapping the alphabets of two SPDFAs to determine sim-ilarity. This type of approach (i.e. using DFA formalism) to manipulating and making inference on programs have been previously studied [1, 3]. In this paper, the idea is extended to similarity estimation.

2.1 Motivation

Despite that many tools have been proposed for detecting source code similarity, the following are the reasons why we have researched into finding a new method. Firstly, *Performance*: to improve — if possible — on the performance of the existing techniques, and secondly, *Automata Application*: to attempt and apply automata theory in detecting program similarity.

3 Background and Related Work

3.1 Overview of Source Code Plagiarism

Source code plagiarism is the textual alteration of computer programs originally owned by another person [9]. This is a problem that is on the rise in undergraduate computer science courses that involves submitting programming assignments [22]. There are two broad categories of source code plagiarism: Lexical and Structural alterations [19]. Lexical alterations involve changing the program text slightly and usually does not require much knowledge of the semantics, while structural alterations involve reordering the code fragments or even using a different control structure. Lexical alterations are also known Type I and II of program cloning [28]. The approach proposed in this paper focuses on detecting lexical alterations. This includes detecting programs that were altered by:

1. reformatting — to change the outlook,
2. editing its comments,
3. adjusting its output (by adding or removing prompts), and

4. adjusting program's granularity — breaking down lines into simple operations or merging them up to condensed operations.

3.2 Terms

The following are the terms used in this paper.

Definition 1 (Programming Plan [12]). *A programming plan or plan is a description of how an operation can be performed. A plan may refer to an entire algorithm, a program segment or even a line of code.*

Definition 2 (Symbol, Alphabet, String [7, 21]). *A symbol is an item. An alphabet is a finite set of symbols. A string the concatenation of zero or more symbols.*

Definition 3 (Regular Languages [21]). *In formal language theory, a regular language $\mathcal{L}$ over a given alphabet $\sum$ is defined as: the empty symbol ϕ, the set of empty string $\{\lambda\}$, the set $\{a\}$ for $a \in \sum$, if $\mathcal{L}_1$ and $\mathcal{L}_2$ are regular languages, then $\mathcal{L}_1\mathcal{L}_2$ — the concatenation and $\mathcal{L}_1|\mathcal{L}_2$ — the alternation, and $\mathcal{L}^{*}{}_{1}$, this is known as the Kleene star. Only the above and any other languages formulated from them are regular languages.*

Definition 4 (DFA [7]). *A deterministic finite automaton or DFA is a five-tuple $M = (q_\circ, Q, \sum, \delta, F)$, where: $q_\circ$ is the start state, Q are the nonempty set of states, with $q_\circ \in Q$, $\sum$ is an alphabet of input symbols, $\delta : Q \times \sum$ is the transition function, and $F \subset Q$ is a set of final states. F, Q, and $\sum$ are all finite sets.*

Definition 5 (Semantic Token and Program String [1]). *A semantic token or symbol, denoted by a_1 or b_1, is a well granulated program line. Examples in C++ are: **float average**;, **int main()**, and **sum = sum + i**;. The program string of a program, often denoted with w, is the sequence of its semantic tokens.*

Definition 6 (SPDFA [1]). A Single Program Deterministic Finite Automaton or SPDFA is a DFA that accepts only one program string.

3.3 Related work

In this section we review related work under three categories: the pre-detection process, techniques used for similarity estimation, and existing tools for similarity detection.

Pre-detection This involves preprocessing activities such as removing white spaces and comments, recognizing tokens, and slicing the programs to the desired granularity level [28, 17, 2].

Techniques Parse trees or annotated abstract syntax trees (AST) based techniques have been used to detect similar code fragments. Both programs are abstracted to AST and subtrees of the AST is compared to determine similarity [5, 30]. Approaches based on Program Dependency Graphs or PDGs are presented in [14, 17]. PDGs are directed-graph structures used to represent data and control flows in programs. Here subgraphs are compared to detect similarity. More on PDGs can be found in [18]. Additionally, other methods include normalizing identifiers by substituting them with defined tokens, metrics based approaches using similarity measurement by parts, i.e. counting the occurrence of syntactic structures such as loops, and so on [28].

Tools Perhaps the most popular tool available for program similarity detection is the Measure of Software Similarity (MOSS) [6]. Others include JPlag [27], Plaggie [4], and so on.

The pre-detection stage is often present in every technique and used to prepare the program for algorithmic manipulation, and similarly, the new technique presented in this paper also does preprocessing using regular expressions. However, the existing techniques described here based on Parse trees or ASTs, PDGs, metrics, etc and are very different from that we offer — DFAs. In terms of tools, we only present a pro-totype called Exact Code Matcher that demonstrates how our new technique can be implemented.

4 Programs to Strings

In this section, we describe the first stage of our new approach — converting programs to strings.

4.1 Preprocessing and granulation

Firstly, it is important to clean-up the student programs to be compared. For this task, we have adapted the preprocessing and granulation modules of the NOPRON [2] tool. These modules work as follows:

Preprocessing The preprocessing module takes a student's program and removes white spaces, comments, and prompts. This is aided by a repository of regular expressions that recognizes and distinguishes between the necessary program text to be kept for comparison and the unnecessary ones to be discarded. The pre-processor is implemented with the Microsoft's .Net regular expression library, i.e. system.text.regularexpressions namespace [25].

Granulation The granulation module takes the cleaned program from the preprocessing module and uses line delimiters in C++ (such as the end of line character or ;, begin and end braces "{" and "}", and the new line delimiters) to separate the lines. The lines are also trimmed to remove padded white spaces. The well-granulated lines of code are then stored in a list structure as an attribute of a program class.

We then proceed to abstract the student programs to regular languages.

4.2 Program String as a Regular Language

The technique presented in this paper is based on the following assumptions:

Student programs have finite steps Any given student program is of finite length, i.e. finite number of lines — *this needs no proof*. Since student programs are based on specific algorithms, and a known property of algorithms is finiteness[1] [13], then we conclude that student program have finite number of lines.

A student program is *regular* By taking each semantic tokens of the program as a symbol of an alphabet and writing a regular expression to accept a concatenation of all symbols, we can present a student program as a single string (program string). For example, let a_0a_1 … a_n be the concatenation of the semantic tokens in a student's program over the alphabet $\sum= \{a_1|1 \leq i \leq n\}$ where n is the number of lines in student's program, then the program string $w = a_0a_1 \ldots a_n$. This is a regular language which can be written as $\mathcal{L} = \{w\}$ containing one string w, with the regular expression $r = (w)$. A DFA can be constructed to represent this language as shown in Figure 1. A formal proof of regularity and finiteness of student programs is given in [1].

Having established that student program can be abstracted to any formal representation of a regular language (such as DFAs or regular expressions), we proceed to present new algorithms for SPDFA construction from student programs.

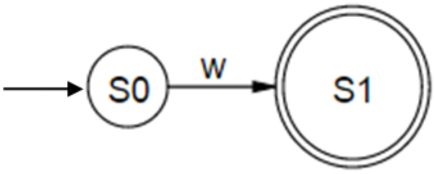


Fig. 1. A DFA that accepts a student's program string w

4.3 SPDFA Construction

Here we present an algorithm for constructing an SPDFA from a given student program. Algorithm 4.1 shows the construction process. We start by creating a start state, and then new states afterwards and add each semantic token in the student's program as transitions between the states. If a newly created state is done on the last token of the student's program, this state is set as the accepting state of the SPDFA. We have implemented Algorithm 4.1 using the DFA state object for the SPDFA states, DFA object for the SPDFA, generic of Transition Function objects for the SPDFA's transitions, and Semantic Token object to implement the symbol denoting a transition

[1] every algorithm must terminate after a number of steps

between SPDFA states. Class diagrams showing the attributes and methods of these objects are shown in Figure 2.

```
Data: A student's program $p_s$
Result: An SPDFA $M$ that accepts the student's program string
create new DFA object and initialize it as an SPDFA $M$ to a ;
create new DFA state object and set SPDFA $M$'s start state attribute to the new state;
add new state to SPDFA's list of states;
create another new DFA state object which represents the next state from the start state;
create a new Transition Function object (i.e. $\delta$) as the first transition from start state to
next state on the first semantic token in $p_s$;
add transition to $M$'s list of transitions;
foreach remaining semantic token $\in$ program $p_s$ do
    create and add a new DFA state object to $M$'s list of states;
    set this state to final or accepting if it is the last semantic token of $P_s$;
    create and add a new Transition Function object with new state on this semantic
    token;
end
return SPDFA $M$
```

Algorithm 4.1: Constructing SPDFA from a student program

Each SPDFA is generated from a unique student program, over a distinct set of semantic tokens in that student's program. Hence, the alphabet of every SPDFA is unique. However, to be able to compare two SPDFAs, we will need to map their alphabets. This is discussed in the next section of this paper.

5 Matching Program Strings

As implied by the earlier stated algorithm for constructing SPDFAs (Algorithm 4.1), each SPDFA accept a single program string, which are not defined over the same alpha-bet as the other SPDFAs. This is because students often write their programs using different identifiers and literals. Therefore, each SPDFA's alphabet will be different.

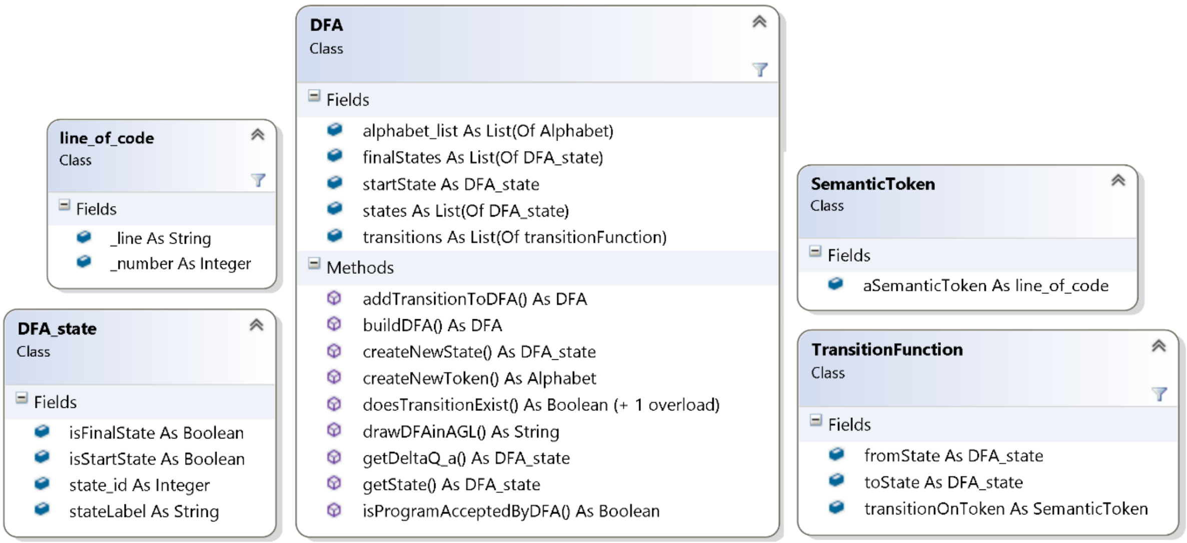
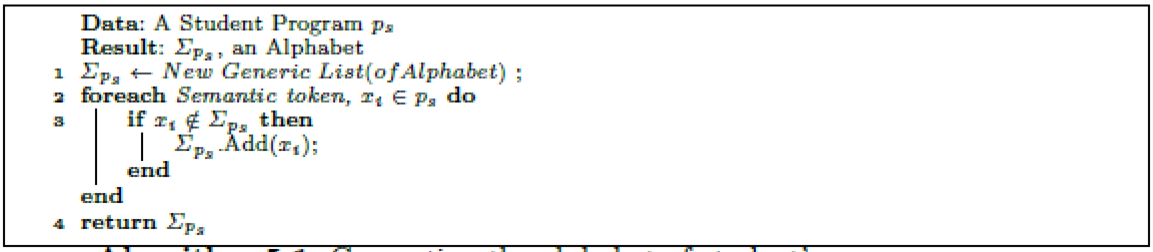

DFA
Class
Fields
alphabet_list As List(Of Alphabet)
finalStates As List(Of DFA_state)
startState As DFA_state
states As List(Of DFA_state)
transitions As List(Of transitionFunction)
Methods
addTransitionToDFA() As DFA
buildDFA() As DFA
createNewState() As DFA_state
createNewToken() As Alphabet
doesTransitionExist() As Boolean (+ 1 overload)
drawDFAinAGL() As String
getDeltaQ_a() As DFA_state
getState() As DFA_state
isProgramAcceptedByDFA() As Boolean
line_of_code
Class
Fields
_line As String
_number As Integer
DFA_state
Class
Fields
isFinalState As Boolean
isStartState As Boolean
state_id As Integer
stateLabel As String
SemanticToken
Class
Fields
aSemanticToken As line_of_code
TransitionFunction
Class
Fields
fromState As DFA_state
toState As DFA_state
transitionOnToken As SemanticToken

Fig. 2. Class diagrams for SPDFA objects

In this section we present an algorithm for generating a different alphabets for each SPDFA.

5.1 Alphabet of Student Programs

We now present an algorithm for generating the alphabet $\sum p_s$ the union of the semantic tokens in a program p_s in Algorithm 5.1. The algorithm iterates through a post-granulation stage program and adds each unique semantic token to the list of symbols of the program's alphabet.



```
Data: A Student Program $p_s$
Result: $\Sigma_{p_s}$, an Alphabet
$\Sigma_{p_s} \leftarrow$ New Generic List(of Alphabet) ;
foreach Semantic token, $x_i \in p_s$ do
    if $x_i \notin \Sigma_{p_s}$ then
        $\Sigma_{p_s}$.Add($x_i$);
    end
end
return $\Sigma_{p_s}$
```

Algorithm 5.1: Generating the alphabet of student's program

Remark 1 (Distinguishing Alphabets). To detect similarity, it is crucial to know what alphabet a program string is composed from. We therefore introduce different notations for the symbols of the two program strings to be compared. The symbols of the first alphabet are denoted with x_{i}, while that of the second is denoted with y_i.

5.2 Matching Semantic Tokens of Different Alphabets

Let $w_1 = x_0 x_1 x_2 \ldots x_n$ and $w_2 = y_0 y_1 y_2 \ldots y_n$ be two program strings, where x_i are symbols from the first student's alphabet and b_i are symbols from the second student's alphabet. The challenge is to determine: when is $x_i = y_i$? This can be determined by mapping the alphabets of the two strings w_1, w_2. This section presents an algorithm that does this. First, we determine if the programming plan on the corresponding lines in both programs are similar using the plan similarity definition presented in [1].

Definition 7 (Plan Similarity [1]). *Two distinct plans* x_i and y_i *are similar if:*

1. *both plans contain exact same characters, or*
2. *the following three conditions are true:*

(a) *the number of tokens in x_i is the same as in y_i, and*

(b) *every token in all position in each plan are of the same literal category, e.g. both are identifiers, same keywords, or numeric values.*

(c) *if a numeric literal appears in the same position in both plans, the floating value of both literals must be the same, i.e. 7 = 7.0, but 12.1 ≠ 12.0.*

Conditions 2a, 2b and 2c must be satisfied for plans to be similar, except when Condition 1 is true.

Example 1 (Similar plans). The following are examples of similar plans:

1. z = x + y; *is similar to* r = p + q; (satisfies rule 2a and 2b),
2. b = b + 3; *is similar to* f = f + 3.0; (satisfies 2a, 2b and 2c),
3. cout<<a; *is similar to* cout<<b; (satisfies 2a and 2b), and
4. nfactorial = 1; *is similar to* nfactorial = 1; (satisfies 1).

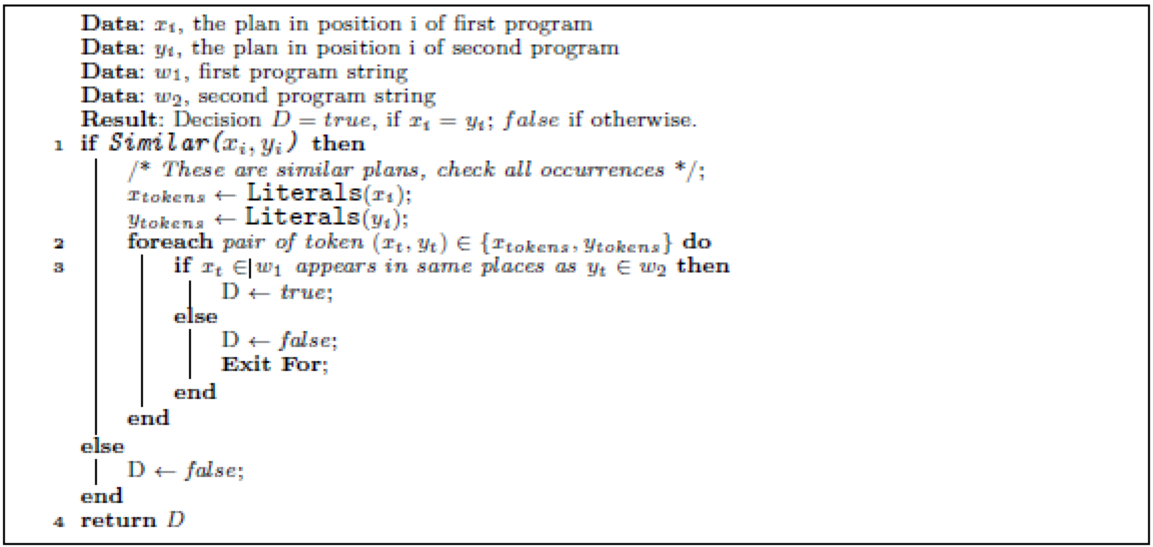
Example 2 (Dissimilar plans). Instances of dissimilar plans are:

1. cin>>a; *is not similar to* cout<<b; (fails rule 2b),
2. c = c + 1; *is not similar to* c += 1; (fails rule 2b. In this case, even though the programming language allows these strings to be valid synonyms, but since we are comparing strings, and our SPDFA will not have additional information to decide this equivalence, we conclude that these strings are different. This conclusion still works perfectly well for lexical alterations).
3. cout<<x+y; *is not similar to* cout<<e+1.0; (fails rule 2b) and
4. #include <iostream> *is not similar to* #include <string> (fails rule 2b).

Now we present an algorithm for matching two plans. Algorithm 5.2, takes two students' plans[2] and decides if they are the same. The algorithm checks if they are similar and if all the literals in both plans have the same occurrence in both students' programs. This algorithm enables us to compare the two program strings.

6 Picking Similar Programs

Given two SPDFAs representing two student programs, the task of checking if the programs are similar is now reduced to the task of matching their respective program strings.



```
Data: $x_i$, the plan in position i of first program
Data: $y_i$, the plan in position i of second program
Data: $w_1$, first program string
Data: $w_2$, second program string
Result: Decision $D = true$, if $x_i = y_i$; $false$ if otherwise.
if Similar($x_i, y_i$) then
    /* These are similar plans, check all occurrences */;
    $x_{tokens} \leftarrow$ Literals($x_i$);
    $y_{tokens} \leftarrow$ Literals($y_i$);
    foreach pair of token $(x_t, y_t) \in \{x_{tokens}, y_{tokens}\}$ do
        if $x_t \in| w_1$ appears in same places as $y_t \in w_2$ then
            D $\leftarrow$ true;
        else
            D $\leftarrow$ false;
            Exit For;
        end
    end
else
    D $\leftarrow$ false;
end
return D
```

Algorithm 5.2: Mapping plans from two alphabets

6.1 Totally Similar Programs

In this section we estimate program similarity in percentages. First, we establish what we mean by *totally similar programs*.

Definition 8 (Totally Similar Programs). *Let p_1 and p_2 be two student programs, we say that p1 is totally similar to p2, if for some alphabets $\sum_{p1}$ and $\sum_{p2}$, there exist two regular languages $\mathcal{L}p_1$ and $\mathcal{L}p_2$, each containing a single program string w_1 and w_2, such that:*

$$w_1 = w_2 \mid \sum p1 \rightarrow \sum p2, \text{ is an injective (one-to-one) mapping of the symbols in } \sum p1 \text{ to the symbols in } \sum p2$$

6.2 No Decisions

With Definition 8, we can decide if the program strings of two student programs are totally similar, i.e. their program strings will be 100% the same. The reverse of this scenario is if the program strings only contain matching substrings. The following are the ways we may classify this:

1. We can use a string similarity estimation algorithm (such as Levenshtein distance) to provide a percentage of similarity between the program strings, and hence the programs. The challenge with this approach is that during alphabet matching (see Algorithm 5.2), symbols in program A that are not found in program B are not matched.

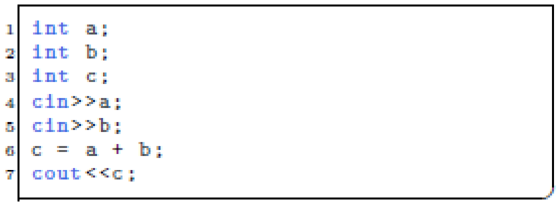
[2] same as the a semantic tokens

Example 3 (Unfair Percentage). Consider the two program fragments P_1 and P_2 in Listing 1.1 and Listing 1.2. Let both programs be over arbitrary alphabets with symbols a_i and b_i respectively. Then the program string of P_1 is: $a_0a_1a_2a_3a_4a_5a_6$, while that of P_2 is: $b_0b_1b_2b_3b_4b_5b_6$. Using the mapping function in Algorithm 5.2 to find $P2$ in the alphabet of P_1, we will still arrive at $b_0b_1b_2b_3b_4b_5b_6$. Which gives us a 0% match. This is because our algorithm will not see the identifiers a as x, b as y, or c as z because of Line 6 in both programs. The plans on this line are dissimilar, contains all three variables and hence, none of the symbols from both alphabets can be matched.

This does not produce a *fair* percentage. By inspection, we would have concluded that the programs contain 6 out of 7 matching lines and hence, should have about 86% percentage similarity. This, of course, reveals that our program string abstraction is very poor at detecting partial similarity, and we are left with the second classification option.

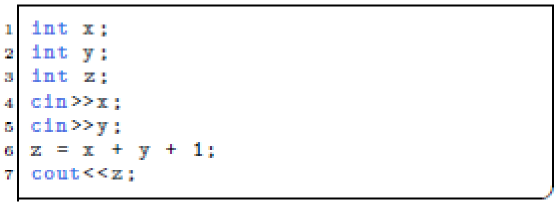
2. No decision. We can chose not to classify program strings that do not match completely. This is because the percentage generated does not always reflect the situation in the programs.

Listing 1.1. P_1



```
1 int a;
2 int b;
3 int c;
4 cin>>a;
5 cin>>b;
6 c = a + b;
7 cout<<c;
```

Listing 1.2. P_2



```
1 int x;
2 int y;
3 int z;
4 cin>>x;
5 cin>>y;
6 z = x + y + 1;
7 cout<<z;
```

6.3 Matching Exact Programs

To ascertain that two programs are *totally similar*, we first have to construct SPDFAs for each program and check with either of the following methods:

SPDFA equivalence. The two SPDFAs are equivalent if they accept the same language[3].
Mapping plans. Using Algorithm 5.2, we can check if the plans of both programs produce the same program string.

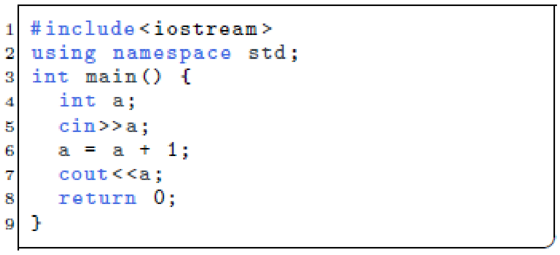
Here we present examples of the estimation process for totally similar programs using two programming problems: the next integer problem, and the sum of n numbers (with recursion) problem.

6.4 Next Integer Problem

We consider a scenario where a student programmer (Bob) had copied the code of another (Alice). In Listing 1.4, Bob tries to alter his program by changing variable names, and adding comment to the Alice's version in Listing 1.3. However, before SPDFA construction, the programs are first granulated, and stripped of comments, prompts and all unnecessary texts with the preprocessor presented in [2] for preprocessing and granulation.

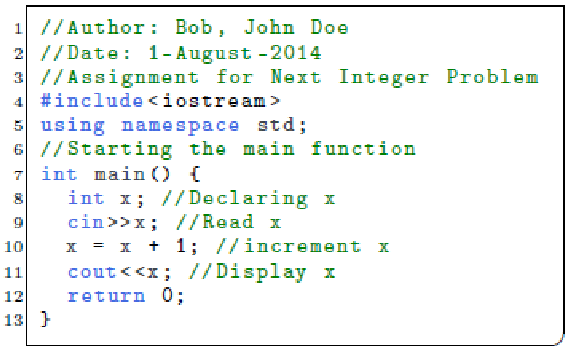
[3] Two DFAs are equivalent if they accept the same regular language [21]

Listing 1.3. Alice's original program for the next integer problem



```
#include<iostream>
using namespace std;
int main() {
  int a;
  cin>>a;
  a = a + 1;
  cout<<a;
  return 0;
}
```

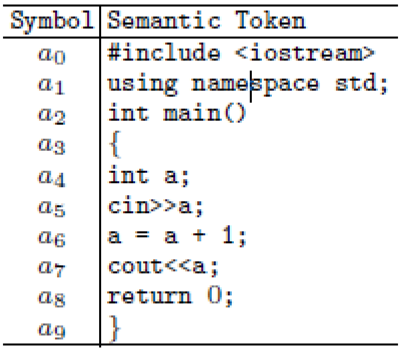
Listing 1.4. Bob's program copied from Alice for the next integer problem



```
//Author: Bob, John Doe
//Date: 1-August-2014
//Assignment for Next Integer Problem
#include<iostream>
using namespace std;
//Starting the main function
int main() {
  int x; //Declaring x
  cin>>x; //Read x
  x = x + 1; //increment x
  cout<<x; //Display x
  return 0;
}
```

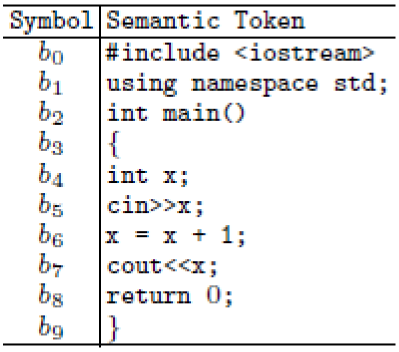
The SPDFAs produced from these programs are displayed in Figure 3 and Figure 4 respectively. The different alphabets of the SPDFAs' languages are also shown in Table 1 and Table 2.

Table 1. Alice's Alphabet



Symbol	Semantic Token
a_0	#include <iostream>
a_1	using namespace std;
a_2	int main()
a_3	{
a_4	int a;
a_5	cin>>a;
a_6	a = a + 1;
a_7	cout<<a;
a_8	return 0;
a_9	}

Table 2. Bob's Alphabet



Symbol	Semantic Token
b_0	#include <iostream>
b_1	using namespace std;
b_2	int main()
b_3	{
b_4	int x;
b_5	cin>>x;
b_6	x = x + 1;
b_7	cout<<x;
b_8	return 0;
b_9	}

Alice's SPDFA accepts the string $a_0a_1 \ldots a_9$ (see Table 1) while Bob's SPDFA accepts $b_0b_1 \ldots b_9$ (see Table 2). Using Algorithm 5.2, we noted that: $a_i = b_i$ 9, that is, a0a1 : : : a9 = b0b1 : : : b9. These SPDFAs accept the same language of a single program string, hence, we conclude that the programs are *totally similar*.

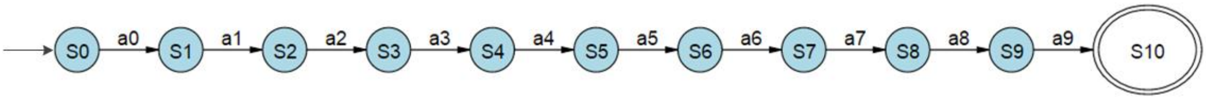


Fig. 3. SPDFA for Alice for the next integer problem

6.5 The Sum Problem with Recursion

In this section we present how the new technique works with a modular program — sum of n numbers that calls SumUp() recursively[4].

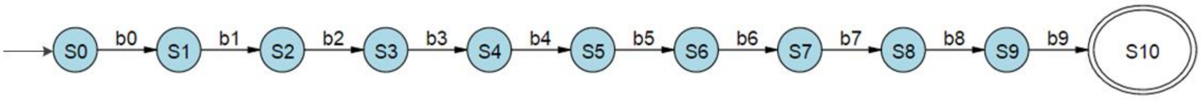
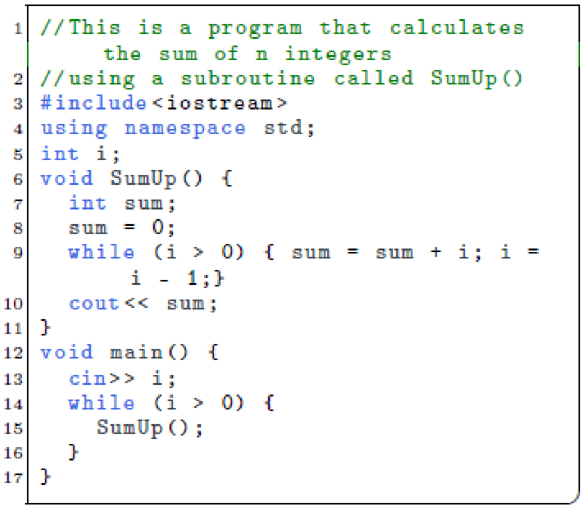


Fig. 4. SPDFA for Bob for the next integer problem

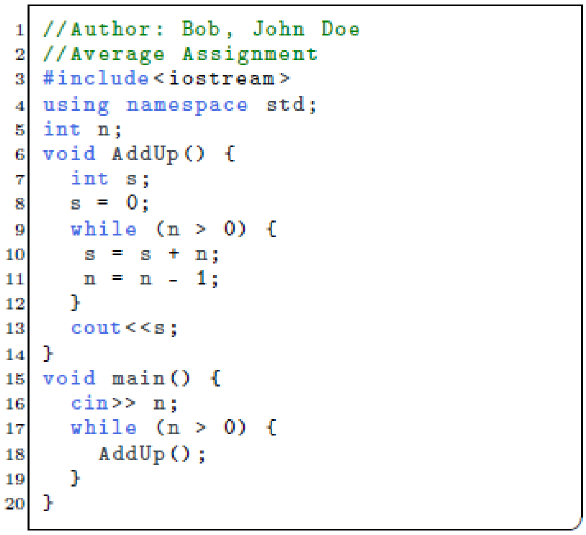
[4] This same program was presented as example/test case in [2] and [24]

Listing 1.5. Alice's original program for the recursive sum problem



```
//This is a program that calculates the sum of n integers
//using a subroutine called SumUp()
#include<iostream>
using namespace std;
int i;
void SumUp() {
  int sum;
  sum = 0;
  while (i > 0) { sum = sum + i; i = i - 1;}
  cout<< sum;
}
void main() {
  cin>> i;
  while (i > 0) {
    SumUp();
  }
}
```

Listing 1.6. Bob's program copied from Alice for the recursive sum problem



```
//Author: Bob, John Doe
//Average Assignment
#include<iostream>
using namespace std;
int n;
void AddUp() {
  int s;
  s = 0;
  while (n > 0) {
   s = s + n;
   n = n - 1;
  }
  cout<<s;
}
void main() {
  cin>> n;
  while (n > 0) {
    AddUp();
  }
}
```

The process is similar to the next integer problem's example. In this case, Alice's and Bob's programs are shown in Listing 1.5 and 1.6. As usual, Bob is the one copying Alice in this example but he tries to change not only the comment and variables of the programs but also the granularity and formatting.
Our technique generated the following two program strings from the programs' SPDFAs:

1. $a_0a_1a_2a_3a_4a_5a_6a_7a_4a_8a_9a_{10}a_{11}a_{10}a_{12}a_4a_{13}a_{14}a_4a_{15}a_{10}a_{10}$, and
2. $b_0b_1b_2b_3b_4b_5b_6b_7b_4b_8b_9b_{10}b_{11}b_{10}b_{12}b_4b_{13}b_{14}b_4b_{15}b_{10}b_{10}$

These strings are found to be the same using the alphabet mapping technique and hence, we conclude that the programs are *totally similar*.

6.6 Implied Properties of SPDFAs

So far we have described the abstraction of any given student program (procedural paradigms only) to a regular language and shown how strings from two of such languages can be compared to decide if they are equivalent or not. One advantage of this abstraction is the ability to extend the properties of regular languages as presented in Martin [21] to SPDFAs. Here we present these properties and what they mean for program similarity estimation, under three categories.

Equivalence Let $\mathcal{L}(M)$ be the language of any given SPDFA M, then:

1. $\mathcal{L}(M)$ is equivalent to the language of the regular expression that represent the set of its program string, i.e. $\mathcal{L}(M) \equiv \{w\}$, where w is the program string. This property enables us to represent student programs regular expressions, with a look-up table that points to the alphabet of the language as demonstrated earlier in this paper.
2. For every SPDFA, it must be possible to construct an SPNFA (Single Pro-gram *Non-deterministic* Finite Automaton), a non-deterministic version of the SPDFA with one or more lambda transitions.
3. It must be possible to: write a *regular* grammar, a *prefix* grammar, and construct an *alternating finite automaton* for the language of any given SPDFA.
4. It must be possible to design a read-only Turing machine that accepts the language of any given SPDFA, and also possible to define the language in Monadic Second-Order Logic.

Closure Let $\mathcal{L}(M)$ be the language of any given SPDFA, then $\mathcal{L}(M)$ is closed under the:

1. union, concatenation, Kleene star, and set complement algebraic operations.

2. homomorphism operations (string and inverse string homomorphisms).
3. reverse operation, $\mathcal{L}(M)^{R}$. This implies that it is possible to have a reverse arrangement of the tokens of a program string, as an operation — even though in our domain of application, this will not be executable on any compiler or useful for any inference.

Decidability Given the languages of any two different SPDFAs, $\mathcal{L}(M_1)$ and $\mathcal{L}(M_2)$, we can decide if:

1. $\mathcal{L}(M_1)$ is contained in $\mathcal{L}(M_2)$, i.e. $\mathcal{L}(M_1) \subseteq \mathcal{L}(M_2)$ or vice-versa.
2. both languages are *disjoint*, meaning they have no semantic token in common - this is particularly useful for checking two entirely distinct programs.
3. a program string w is a member of a language $\mathcal{L}(M_1)$— this is known as the *membership problem*.
4. a language is empty, or universal. SPDFAs after constructions will always have a program string, however, the emptiness or universality questions can be asked if operations have been performed on the language of an SPDFA.

Additional properties of SPDFAs are: *minimality* (every SPDFA has the least number of states and cannot be minimised further), and *singularity* (an SPDFA accepts only one program string). With these properties, one can ask further interesting questions about SPDFA and how they may evolve in program similarity estimation.

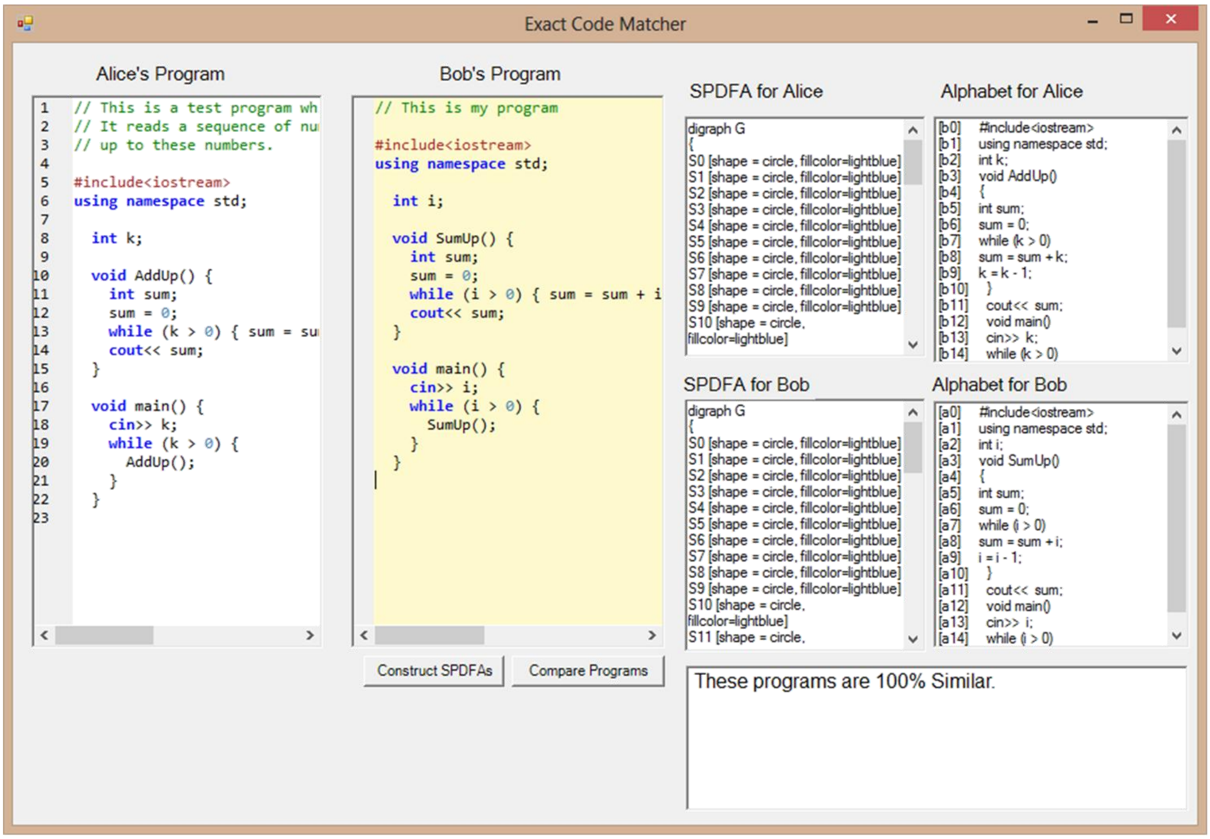
7 Experiment and Performance Analysis

In this section we present more test cases and an analysis of how the SPDFA approach did against other well used techniques for detecting similarity in programs.

7.1 The Exact Code Matcher

First we describes a simple tool, called the Exact Code Matcher that matches two seemingly different programs using the SPDFA matching technique discussed earlier in this paper. In Figure 5, Exact Code Matcher takes two programs looking different, granulates them and tells if they are 100% similar or not. As part of the report generated, the Exact Code Matcher displays the SPDFAs of the two programs (as AGL scripts[5]) and their corresponding alphabets. It also shows a feedback of its assessment which is either 100% or 0% (representing no decision).

[5] AGL is an online tool for drawing directed graphs. http://rise4fun.com/Agl



Alice's Program

```
// This is a test program wh
// It reads a sequence of nu
// up to these numbers.

#include<iostream>
using namespace std;

  int k;

  void AddUp() {
    int sum;
    sum = 0;
    while (k > 0) { sum = su
    cout<< sum;
  }

  void main() {
    cin>> k;
    while (k > 0) {
      AddUp();
    }
  }
```

Bob's Program

```
// This is my program

#include<iostream>
using namespace std;

  int i;

  void SumUp() {
    int sum;
    sum = 0;
    while (i > 0) { sum = sum + i
    cout<< sum;
  }

  void main() {
    cin>> i;
    while (i > 0) {
      SumUp();
    }
  }
```

SPDFA for Alice

```
digraph G
{
S0 [shape = circle, fillcolor=lightblue]
S1 [shape = circle, fillcolor=lightblue]
S2 [shape = circle, fillcolor=lightblue]
S3 [shape = circle, fillcolor=lightblue]
S4 [shape = circle, fillcolor=lightblue]
S5 [shape = circle, fillcolor=lightblue]
S6 [shape = circle, fillcolor=lightblue]
S7 [shape = circle, fillcolor=lightblue]
S8 [shape = circle, fillcolor=lightblue]
S9 [shape = circle, fillcolor=lightblue]
S10 [shape = circle,
fillcolor=lightblue]
```

Alphabet for Alice

```
[b0]    #include<iostream>
[b1]    using namespace std;
[b2]    int k;
[b3]    void AddUp()
[b4]    {
[b5]    int sum;
[b6]    sum = 0;
[b7]    while (k > 0)
[b8]    sum = sum + k;
[b9]    k = k - 1;
[b10]     }
[b11]    cout<<  sum;
[b12]    void main()
[b13]    cin>>  k;
[b14]    while (k > 0)
```

SPDFA for Bob

```
digraph G
{
S0 [shape = circle, fillcolor=lightblue]
S1 [shape = circle, fillcolor=lightblue]
S2 [shape = circle, fillcolor=lightblue]
S3 [shape = circle, fillcolor=lightblue]
S4 [shape = circle, fillcolor=lightblue]
S5 [shape = circle, fillcolor=lightblue]
S6 [shape = circle, fillcolor=lightblue]
S7 [shape = circle, fillcolor=lightblue]
S8 [shape = circle, fillcolor=lightblue]
S9 [shape = circle, fillcolor=lightblue]
S10 [shape = circle,
fillcolor=lightblue]
S11 [shape = circle,
```

Alphabet for Bob

```
[a0]    #include<iostream>
[a1]    using namespace std;
[a2]    int i;
[a3]    void SumUp()
[a4]    {
[a5]    int sum;
[a6]    sum = 0;
[a7]    while (i > 0)
[a8]    sum = sum + i;
[a9]    i = i - 1;
[a10]     }
[a11]    cout<<  sum;
[a12]    void main()
[a13]    cin>>  i;
[a14]    while (i > 0)
```

```
These programs are 100% Similar.
```

Fig. 5. The Exact Code Matcher

7.2 More Test Cases

Here we apply the new approach in computing the similarity of hypothetical[6] student programs for a class of 10 students. These programs are shown in Listing 1.7 through Listing 1.16. The programming problem addressed by these programs is finding the sum of the first n-integers, given n. After running a similarity measurement experiment using SPDFAs implemented with the .Net framework using Visual Studio IDE, we have written the output of the experiment to the *Immediate Window*[7] as it is shown in Figure 6. The experiment takes a class of 10 students' programs, attempts to construct SPDFAs for the 10 programs in turn and determine the similarity between every program in the set and other programs of the set. The output of this similarity experiment is further presented in a cross-table (or pivot table). Since we know that a program will be 100% similar to itself, then there is no reason to compare a student's program with itself, i.e. no need to do a $n \times n$ comparison. Similarly, our comparison only holds for the upper part of the matrix (*see* Table 3) because it is redundant to compute the similarity of two pair of programs twice. Similarity measure is a commutative operation, i.e. *similarity* $(a,\ b) \equiv$ similarity$(b,\ a)$. In Table 3, the lower matrix is marked "x" to indicate that the cell is a redundant computation. Hence, given any program pair $(p_i\,, p_j)$, $i,\ j \in [1; n]$, we only compute similarity for the set $(p_i\,, p_j)$ if $i,\neq j$ and $\forall\ i,\ j\ \exists\ (p_i\,, p_j)$.

[6] These are not real life student programs, they are hypothetical data designed to test the presented technique. The names in the programs were generated with a pseudo-random names generator.

[7] An output screen in Microsoft's Visual Studio, similar to system.out.

Listing 1.7. *Student*$_1$

```
 1 #include <iostream>
 2 using namespace std;
 3 int main() {
 4   int sum, n, i;
 5   sum=0;
 6   cin>>n;
 7   for(i=1; i<=n; i++){
 8     sum+=i;
 9   }
10   cout<<sum;
11   return 0;
12 }
13 // Name: Silva Araujo
14 // Student number:
       98829346
```

Listing 1.8. *Student*$_2$

```
 1 // Name: Jaana
       Wiljakainen
 2 // Number: 76283222
 3 #include <iostream>
 4 using namespace std;
 5 int main() {
 6   int thesum;
 7   int number;
 8   int counter;
 9   thesum=0;
10   cin>>number;
11   for(counter=1; counter
         <=number; counter
         ++){
12     thesum+=counter;
13   }
14   cout<<thesum;
15   return 0;
16 }
```

Listing 1.9. *Student*$_3$

```
 1 //Student submission for
       Assignment 1
 2 //Name: Linda Hendrickson
 3 #include <iostream>
 4 using namespace std;
 5 int main() {
 6   int totalsum, num, i;
 7   //now i will initialize
         the total sum to
        zero
 8   totalsum=0;
 9   //read in a value
10   cin>>num;
11   for(i=1; i<=num; i++){
12     totalsum = totalsum +
           i;
13   }
14   cout<< totalsum;
15   return 0;
16 }
```

Listing 1.10. *Student*$_4$

```
 1 // Mike Shuster
 2 // Student number:
       67382980
 3 // Name: Leia Berggren
 4 #include <iostream>
 5 using namespace std;
 6 int main() {
 7   int p, a, j;
 8   p=0;
 9   cin>>a;
10   for(j=1; j<=a; j++){
11     p+=j;
12   }
13   cout<<p;
14   return 0;
15 }
```

Listing 1.11. *Student*$_5$

```
 1 #include <iostream>
 2 using namespace std;
 3 int main() {
 4 // Student's name:
         Blanche Couch
 5 // Assignment submission
 6   int sum;
 7   int n;
 8   int i;
 9   sum=0;
10   cin>>n;
11   for(i=1; i<=n; i++){
12     sum+=i;
13   }
14   cout<<sum;
15   return 0;
16 }
```

Listing 1.12. *Student*$_6$

```
 1 #include <iostream>
 2 using namespace std;
 3 int main() {
 4 // Nina Mishina's program
 5 // Student number:
       90040267
 6   int s; int n; int i;
 7   s=0;
 8   cin>>n; i=1;
 9   while (i<=n){
10     s+=i; i= i+ 1;
11   }
12   cout<<s;
13   return 0;
14 }
```

Listing 1.13. *Student*$_7$

```
 1 #include <iostream>
 2 using namespace std;
 3 int main() {
 4   // Author's name: Alex
         Fuller
 5   int total;
 6   int n;
 7   int j;
 8   total=0;
 9   //read in a value
10   cin>>n;
11   for(j=1; j<=n; j++){
12     total = total + j;
13   }
14   cout<< total;
15   return 0;
16 }
```

Listing 1.14. *Student*$_8$

```
 1 #include <iostream>
 2 using namespace std;
 3 int main() {
 4   // Student Name: Rosa
         Musser
 5   // Matric Number:
         78927740
 6   int sum; int number;
         int counter;
 7   sum=0; cin>>number;
 8   for(counter=1; counter
         <=number; counter
         ++){
 9     sum+=counter;
10   }
11   cout<<sum;
12   return 0;
13 }
```

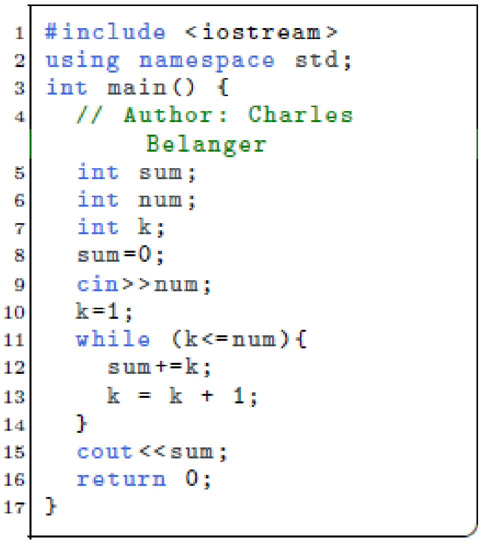
Listing 1.15. *Student*$_9$

```
 1 #include <iostream>
 2 using namespace std;
 3 int main() {
 4   // Leia Berggren,
         67362312
 5   int thesum, number,
         counter;
 6   thesum=0;
 7   cin>>number;
 8   cout << "You entered"
         << number;
 9   for(counter=1; counter
         <=number; counter
         ++){
10     thesum+=counter;
11   }
12   cout<<thesum;
13   return 0;
14 }
```

We proceed to make inferences from Table 3. If an entire row or column has zero values, this means that the program of the student on this row or column is unique — observe that this is true for column 9. We can understand why our algorithm classified this program as unique by inspecting student 9's program (see Listing 1.15). On Line

8, the student decided to output the same number entered by the user on the previous line. After preprocessing, this line remains as a semantic token in this student's program string. By inspection, this program is similar to Listing 1.14 but since SPDFA matches verbatim strings, it is classified as a *no decision*. Hence, Listing 1.15 is regarded as unique. The algorithm identified other similarities and clustered them in Table 3 as follows: Programs 1 ≡ 2 ≡ 4 ≡ 5 ≡ 8 (row 1), 3 ≡ 7 (row 3), and 6 ≡ 10 (row 6). These listings however have many lexical alterations such as comments added in different places and different variable identifiers used.

Listing 1.16. $Student_{10}$



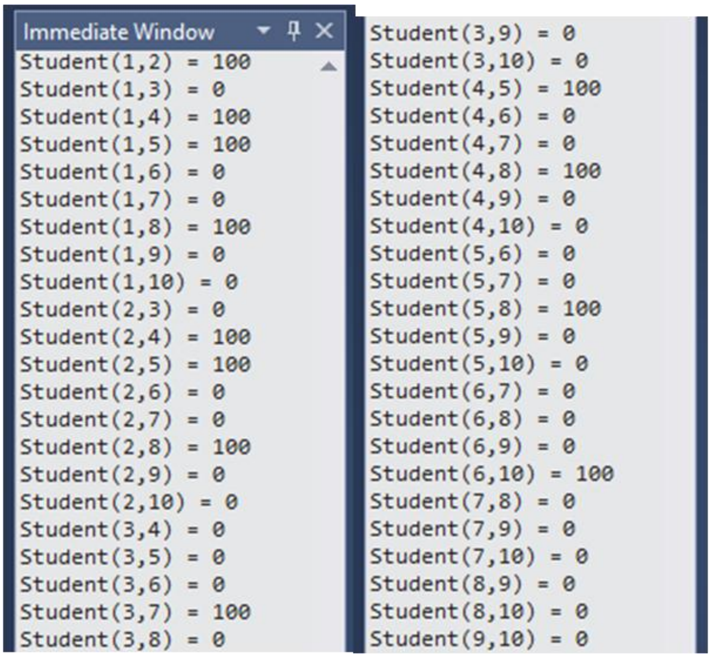
```
#include <iostream>
using namespace std;
int main() {
  // Author: Charles
     Belanger
  int sum;
  int num;
  int k;
  sum=0;
  cin>>num;
  k=1;
  while (k<=num){
    sum+=k;
    k = k + 1;
  }
  cout<<sum;
  return 0;
}
```

7.3 High-Level Evaluation

Here we compare our prototype's technique (the SPDFAs approach) with existing techniques in terms of portability[8], precision[9], and scalability[10]. Table 4 shows how the new technique stands with existing one. The performance of the existing techniques shown in this table are as published by Roy and Cordy [28].

Table 3. Class similarity comparison of n to n -1 students, where n = 10

	Student									
Student	1	2	3	4	5	6	7	8	9	10
1	x	100	0	100	100	0	0	100	0	0
2	x	x	0	100	100	0	0	100	0	0
3	x	x	x	0	0	0	100	0	0	0
4	x	x	x	x	100	0	0	100	0	0
5	x	x	x	x	x	0	0	100	0	0
6	x	x	x	x	x	x	0	0	0	100
7	x	x	x	x	x	x	x	0	0	0
8	x	x	x	x	x	x	x	x	0	0
9	x	x	x	x	x	x	x	x	x	0
10	x	x	x	x	x	x	x	x	x	x



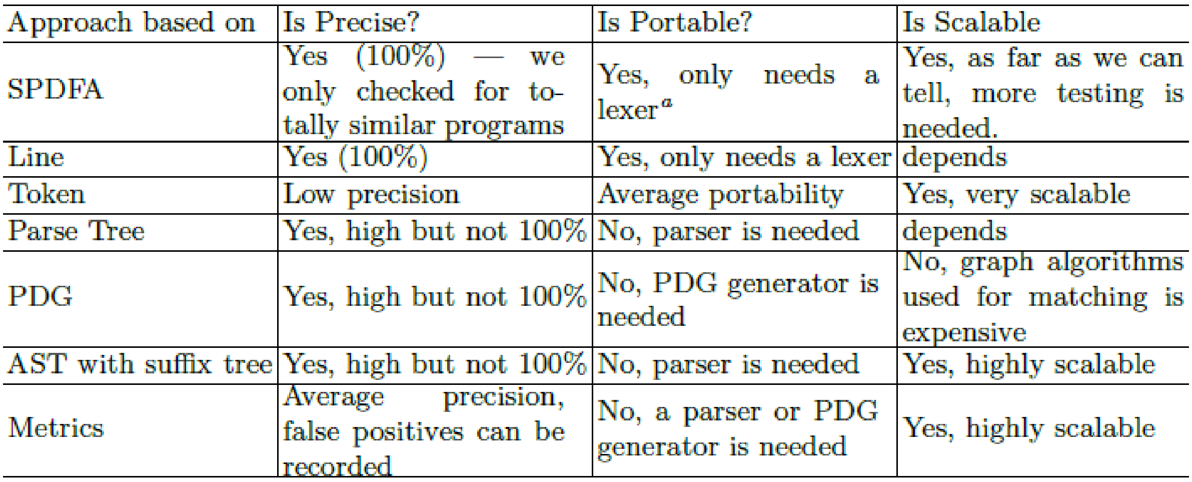
```
Student(1,2) = 100
Student(1,3) = 0
Student(1,4) = 100
Student(1,5) = 100
Student(1,6) = 0
Student(1,7) = 0
Student(1,8) = 100
Student(1,9) = 0
Student(1,10) = 0
Student(2,3) = 0
Student(2,4) = 100
Student(2,5) = 100
Student(2,6) = 0
Student(2,7) = 0
Student(2,8) = 100
Student(2,9) = 0
Student(2,10) = 0
Student(3,4) = 0
Student(3,5) = 0
Student(3,6) = 0
Student(3,7) = 100
Student(3,8) = 0
Student(3,9) = 0
Student(3,10) = 0
Student(4,5) = 100
Student(4,6) = 0
Student(4,7) = 0
Student(4,8) = 100
Student(4,9) = 0
Student(4,10) = 0
Student(5,6) = 0
Student(5,7) = 0
Student(5,8) = 100
Student(5,9) = 0
Student(5,10) = 0
Student(6,7) = 0
Student(6,8) = 0
Student(6,9) = 0
Student(6,10) = 100
Student(7,8) = 0
Student(7,9) = 0
Student(7,10) = 0
Student(8,9) = 0
Student(8,10) = 0
Student(9,10) = 0
```

Fig. 6. Output on Immediate Window (or console) showing iterations of experiment

From Table 4, we see that the new SPDFA technique does well in terms of precision like most other techniques with high precision, however, SPDFAs offer 100% precision — a complete program string matching, and this is better than all other techniques except for the line technique. However, SPDFAs are more *intelligent* than line techniques because the consider the whole program as a single string. In terms of portability, SPDFAs does better than five different approaches, namely the Token, Parse-tree, PDG, Metrics, and AST with Suffix techniques — *this is good*, and means one can con-sider adopting our technique when thinking of a highly distributable student similarity detector. Finally, in terms of scalability, the documented assessments of other systems are quite vague, and similarly, for SPDFAs, we cannot comment on that in this paper until we have tried it on many programs — *a large sample size*.

[8] ability to distribute it across platforms
[9] avoiding false positives
[10] can it be implemented on a large scale?

Table 4. Benchmarking SPDFAs with existing techniques



Approach based on	Is Precise?	Is Portable?	Is Scalable
SPDFA	Yes (100%) — we only checked for totally similar programs	Yes, only needs a lexer[a]	Yes, as far as we can tell, more testing is needed.
Line	Yes (100%)	Yes, only needs a lexer	depends
Token	Low precision	Average portability	Yes, very scalable
Parse Tree	Yes, high but not 100%	No, parser is needed	depends
PDG	Yes, high but not 100%	No, PDG generator is needed	No, graph algorithms used for matching is expensive
AST with suffix tree	Yes, high but not 100%	No, parser is needed	Yes, highly scalable
Metrics	Average precision, false positives can be recorded	No, a parser or PDG generator is needed	Yes, highly scalable

7.4 Computational Complexity

Here we give a statement on the complexity of the new algorithms. Almost all algorithms used by the Exact Code Matcher runs in linear time, i.e. $O(n)$, where n is the number of lines in the program. The exception is Algorithm 5.2. On Line 3 of Algorithm 5.2, it calls a function called Appearance that checks if two tokens has appeared in the same positions in the two programs, this is done inside a for-loop and gives a complexity of $O(n \times m)$, where n is the number of lines as usual and m is the number of tokens in the program (this include literals, keywords and identifiers). Comparing this time complexity with what is documented for other techniques, SPDFA also does well. According to [28], most PDG approaches run in non-polynomial time or worse. Examples include are given in Krinke [17] and Liu *et al* [20]. Tree-based techniques have better complexities of $O(N)$ where N is the number of AST nodes [5]. Most other techniques do not have their time complexity published [28]. Considering the fact that the most costly algorithm used by Exact Code Matcher is has a time complexity of $O(n \times m)$, which is the algorithm that does the actual comparison of programs, we say that SPDFA technique is good enough for applications where portability and accuracy are of paramount importance.

8 Conclusion and Future Work

Program similarity detection is not only useful in picking plagiarised student assignments, it is also used in the software industry to check for software clones during the process of software maintenance [5, 28]. Hence, the need to find more ways of carrying out the task. In this paper we have presented a new technique of detecting similar programs using SPDFAs — a type of DFA. This technique is good at detecting student programs that are apparently dissimilar (because of the comments, identifier choices, and other program texts), but are possibly plagiarised. The technique has its pitfall because it cannot report on partially copied programs. However, the technique is quick enough, and does better in terms of portability and accuracy. The technique can be useful for checking programming assignments with lexical alterations in a large class size. Finally, one exciting aspect of this work is seeing DFAs being applied in the domain of source code plagiarism detection. Advancement and modification of this technique holds better possibilities to the results displayed in this paper. There are some interesting future directions triggered by the results presented in this paper, such as large scale testing of the SPDFAs technique using real student programming assignments, and modification of the comparison algorithm to be able to detect partially copied programs.

References

1. Ade-Ibijola, A.: New Finite Automata Applications in Novice Program Comprehension.Lambert Academic Publishing, Germany, ISBN: 978-3330029057 (2017)

2. Ade-Ibijola, A., Ewert, S., Sanders, I.: Abstracting and narrating novice programs using regular expressions. In: Proceedings of the Annual Conference of the South African Institute for Computer Scientists and Information Technologists. pp. 19–28. ACM (2014)
3. Ade-Ibijola, A., Ewert, S., Sanders, I.: Introducing Code Adviser: A DFA-driven electronic programming tutor. In: Proceedings of the 20th International Conference on the Implementation and Applications of Automata. pp. 307–312. Springer (2015)
4. Ahtiainen, A., Surakka, S., Rahikainen, M.: Plaggie: GNU-licensed source code plagiarism detection engine for Java exercises. In: Proceedings of the 6th Baltic Sea conference on Computing education research. pp. 141–142. ACM (2006)
5. Baxter, I.D., Yahin, A., Moura, L., Sant'Anna, M., Bier, L.: Clone detection using abstract syntax trees. In: Proceedings of the International Conference on Software Maintenance. pp. 368–377. IEEE Computer Society, Washington, DC, USA (1998), http://dl.acm.org/citation.cfm?id=850947.853341
6. Bowyer, K., Hall, L.: Experience using "MOSS" to detect cheating on programming assignments. In: 29th Annual Frontiers in Education Conference. vol. 3, pp. 13B3/18–13B3/22 (November 1999)
7. Brzozowski, J., Tamm, H.: Quotient complexities of atoms of regular languages. In: De-velopments in Language Theory. pp. 50–61. Springer (2012)
8. Cosma, G., Joy, M.: Towards a definition of source-code plagiarism. IEEE Transactions on Education 51(2), 195–200 (2008)
9. DJurić, Z., Gašević, D.: A source code similarity system for plagiarism detection. The Computer Journal p. bxs018 (2012)
10. Hage, J., Rademaker, P., van Vugt, N.: Plagiarism detection for Java: a tool comparison. In: Computer Science Education Research Conference. pp. 33–46. Open Universiteit, Heerlen (2011)
11. Jenkins, T., Helmore, S.: Coursework for cash: the threat from on-line plagiarism. In: 7th Annual Conference of the Higher Education Academy Network for Information and Computer Sciences. pp. 121–126. Citeseer (2006)
12. Johnson, W.L.: Understanding and debugging novice programs. Artificial Intelligence 42(1), 51–97 (1990)
13. Knuth, D.E.: The Art of Computer Programming Vol: 1. Addison-Wesley (1968)
14. Komondoor, R., Horwitz, S.: Tool demonstration: Finding duplicated code using program dependences. In: Proceedings of the 10th European Symposium on Program-ming Languages and Systems. pp. 383–386. Springer-Verlag, London, UK (2001), http://dl.acm.org/citation.cfm?id=645395.651925
15. Komondoor, R., Horwitz, S.: Effective, automatic procedure extraction. In: 11th IEEE International Workshop on Program Comprehension. pp. 33–42. IEEE (2003)
16. Koss, I., Ford, R.: Authorship is continuous: Managing code plagiarism. IEEE Security & Privacy (2), 72–74 (2013)
17. Krinke, J.: Identifying similar code with program dependence graphs. In: 8th Working Conference on Reverse Engineering. pp. 301–309. IEEE (2001)
18. Krinke, J.: Program slicing. Handbook of Software Engineering and Knowledge Engineer-ing 3, 307–332 (2005)
19. Kustanto, C., Liem, I.: Automatic source code plagiarism detection. In: 10th ACIS In-ternational Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing. pp. 481–486 (May 2009)
20. Liu, C., Chen, C., Han, J., Yu, P.S.: GPLAG: Detection of software plagiarism by program dependence graph analysis. In: Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining. pp. 872–881. ACM, New York, USA (2006), http://doi.acm.org/10.1145/1150402.1150522
21. Martin, J.: Introduction to Languages and the Theory of Computation. McGraw-Hill, New York (2003)
22. Maurer, H., Kappe, F., Zaka, B.: Plagiarism - a survey. Journal of Universal Computer Science 12(8), 1050–1084 (2006)
23. Mishne, G., De Rijke, M.: Source code retrieval using conceptual similarity. In: Conference on Computer Assisted Information Retrieval. pp. 539–554. Citeseer (2004)
24. Mössenböck, H.: An example of a Taste program. The compiler generator Coco/R User Manual pp. 30–32 (2010)
25. MSDN: Regular Expression Language–quick reference: Microsoft Developer Net-work Documentation on the .Net framework, version 4.5, regular expressions. http://msdn.microsoft.com/en-us/library/az24scfc.asp, accessed: 2015-05-19
26. Nixon, M.A.: Cheating in cyberspace: Maintaining quality in online education. AACE Journal 12(1), 85–99 (2004)
27. Prechelt, L., Malpohl, G., Philippsen, M.: Finding plagiarisms among a set of programs with JPlag. Journal of Universal Computer Science 8(11), 1016–1038 (2002)
28. Roy, C.K., Cordy, J.R.: A survey on software clone detection research. Tech. rep., Technical Report 541, Queen's University at Kingston (2007)
29. Scanlon, P.M., Neumann, D.R.: Internet plagiarism among college students. Journal of College Student Development 43(3), 374–385 (2002)
30. Wahler, V., Seipel, D., Gudenberg, J.W.v., Fischer, G.: Clone detection in source code by frequent itemset techniques. In: Proceedings of the 4th IEEE International Workshop on Source Code Analysis and Manipulation. pp. 128–135. IEEE Computer Society, Washing-ton, DC, USA (2004), http://dx.doi.org/10.1109/SCAM.2004.5

Using the Spreadsheet Paradigm to Introduce Fundamental Concepts of Programming to Novices

Bennett Kankuzi and Bassey Isong and Lucia Letlonkane

Department of Computer Science, North-West University, Mafikeng Campus,
P/Bag X2046, Mmabatho 2735, South Africa
bennett.kankuzi@nwu.ac.za

Abstract. In this paper, we explore the potential of introducing fundamental concepts of programming to first year students using spreadsheets. Many students find programming to be difficult to learn as the concepts are new and unfamiliar. On the other hand, spreadsheets have a relatively easy to use interface that provides instant viewing of results of data manipulation. Fundamental concepts of programming can then be indirectly introduced using their spreadsheet equivalents before the students start their formal introductory programming class. This paper specifically focusses on how fundamental programming concepts can be mapped to their spreadsheet equivalents.

Keywords: Teaching Programming, Novice Programmers, Spreadsheets

1 Introduction

Many students find it difficult to learn programming as the concepts are new and unfamiliar [6, 8, 17]. Students have to master the art of problem solving using a programming language while at the same time they have to learn the syntactic and semantic details of a programming language [6, 17]. Compounding this problem is the fact that many programming languages currently being used to teach students how to program, were not primarily meant for teaching programming [8]. As such, many students are turned off from programming and ultimately view Computer Science in general as hard. It is thus imperative that techniques and ways are found to improve programming education as programming is a fundamental subject that acts as a rite of passage to Computer Science [6, 8, 13].

Various researchers have proposed ways in which programming education can be improved in order to address the challenges in introducing programming to students. It is, however, generally agreed that before students take a formal introductory programming class, they should first be introduced to sound problem solving skills [2, 6]. At the same time, it is important to prepare students indirectly for the fundamental concepts of programming that they are to encounter in a formal introductory programming class [21]. A balance is therefore needed that problem solving skills are inculcated while at the same time indirectly introducing fundamental concepts of programming. There is, therefore, need for programming environments that can help achieve this balance. Brusilovsky et al. [3] recommend simple visual programming environments for novices as they protect novices from cognitive load.

Spreadsheets are popular end-user visual programming environments [1]. Spreadsheets are intuitive and easy to use as the visual interface provides instant viewing of results of data manipulation [14]. As such, spreadsheet users who are normally not professional programmers can solve problems with little investment of time and effort as long as they have expertise in the problem domain [15]. This provides for an opportunity to emphasize on problem solving skills to beginning programmers as long as the study problems can be related to by the students. On the other hand, spreadsheet development is actually functional programming as spreadsheets are first-order functional programming languages [1]. The beauty of spreadsheets, however, is that spreadsheet developers do not have to think of themselves as doing programming [1]. Yet, as they solve their problems using spreadsheets, they indirectly follow some fundamental concepts of traditional programming. For example, assignment commands, data values, data types, variables, functions and control structures are some fundamental concepts of tra-

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

ditional programming languages [11, 20] that spreadsheet developers indirectly use as they develop their spreadsheet programs. In spreadsheets, there are equivalent concepts such as assignment through formulas, data values, data types, cell references, functions and decision constructs. Spreadsheet programming, although different from traditional programming, can therefore provide a platform in which novice programmers can be introduced to problem solving skills while at the same time introducing them to fundamental aspects of programming without them realizing that they are learning fundamental concepts of traditional programming.

We are thus proposing that before students take a formal introductory programming course, they should first be taught spreadsheet programming where fundamental programming concepts are indirectly introduced to them through problem solving using spreadsheets. This could be done during the traditional "Introduction to Computer Science" course where software applications such as word processors and spreadsheets are taught to students. By solving real-world problems using spreadsheets, students can be introduced to computational thinking through abstracting real-world problems into corresponding spreadsheet models which consist of label cells, data cells and formula cells.

In this paper, we focus on how fundamental concepts of traditional programming can be mapped to their equivalents in spreadsheets. In particular, the paper has been structured as follows: in Sect. 2, we discuss related literature. In Sect. 3, we discuss our proposed spreadsheet approach on how fundamental aspects of traditional programming can be mapped to their spreadsheet equivalents. We conclude in Sect. 4 by also discussing the limitations of the proposed spreadsheet approach.

2 Related Work

Other researchers have also explored the potential of using alternative techniques as a way to address the challenges students face when being introduced to programming using traditional programming languages.

Some researchers have suggested the use of pseudo-code instead of proper language constructs when introducing programming to students [17]. Mini-languages have also been suggested as a way to introduce students to programming [4]. Some have also proposed the use of drag and drop actor based visual programming environments such as Alice [16]. However, one disadvantage of these approaches is that students find it difficult to transition to traditional programming languages as they now have to grapple with learning the syntactical details of that language, a feature which is not present in these approaches [16].

Shakelford et al. [18] proposed that before first year students join their first introductory programming class, they should be enrolled in an "Introduction to Computing" course. In this course, students could learn how to use application-oriented software packages such as spreadsheets, databases and web-browsers in solving computation problems [18]. In doing so, students could develop computational problem solving skills that would help them in a formal introductory programming class [18]. In our case, we are specifically proposing the need for students to be first taught spreadsheet programming before they undertake a formal introductory programming course.

Warren [22] also explored the potential of using scripting languages such as JavaScript, instead of traditional programming languages such as C++ and Java, to teach programming to students. Scripting languages are touted to be easy to use and provide a minimal set of features that are not distracting for novice programmers [23]. However, scripting programming language environments might be intimidating to novices.

Martin [12] proposed introducing non-programmers to programming by integrating spreadsheet development with developing programs using Visual Basic for Applications, which is a programming environment extension to the Microsoft Excel spreadsheet system. This is in contrast to introducing programming using stand-alone Visual Basic [12]. In our case, we are not proposing integrating a programming environment to the spreadsheet system. Instead, students should be indirectly introduced to fundamentals of programming using the basic spreadsheet environment.

Lovaszov et al. [10] also proposed the use of spreadsheets as an alternative way of introducing programming to students. However, they did not put their proposal into practice on novice programmers. Instead, they used spreadsheets to demonstrate to trainee Computer Science teachers (already skilled programmers), on how spreadsheets can be used to simulate, side by side, a solution to an algorithm [10]. As students experiment with data values in a spreadsheet solution, they also relate how a particular algorithm works, hence they can easily understand how a particular programming concept works [10]. The rationale of introducing this approach to trainee Computer Science teachers was to demonstrate spreadsheet programming as one alternative way of introducing programming concepts to students, other than the traditional way of introducing programming through general-purpose programming languages. In our case, we are proposing the use of spreadsheets to introduce programming concepts to novice programmers themselves.

Warren [23] also proposed that as much as possible, fundamental programming concepts should be introduced through programming environments such as spreadsheets as computations in those environments are easier to comprehend. Warren [23] also argues that in spreadsheets, the system status is visible as the whole computation is displayed and there is no time lapse waiting for a computation to be done. Warren [23] therefore recommends that spreadsheets can be used to introduce fundamental programming concepts such as data types and function use. In this paper, we are also advocating for the same. However, we are specifically focussing on how fundamental programming concepts can be mapped to their spreadsheet equivalents.

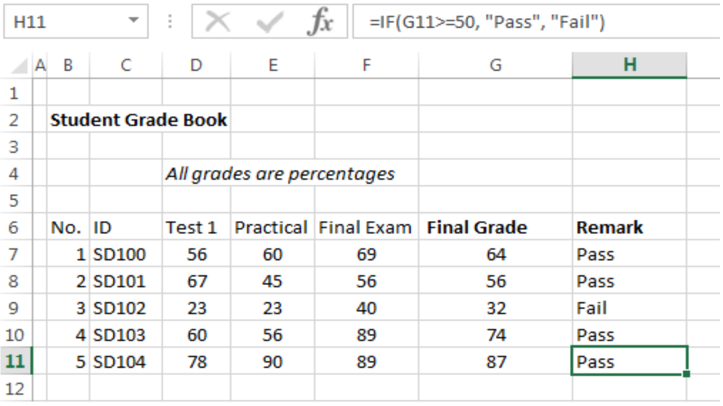
Tahy [21] also proposed using spreadsheets indirectly to teach programming as way of promoting digital literacy. In this case, it is assumed that by learning to program, though indirectly through spreadsheets, digital literacy can be improved in a society. In this proposed technique, the teaching of programming is embedded in the teaching of a spreadsheet application such that fundamental programming concepts such as data types and data structures are introduced in the course of solving a problem using a spreadsheet application [21]. Tahy [21] also recommends her proposed technique as a way of indirectly introducing programming concepts to students. In this paper, we are also proposing using spreadsheet concepts to indirectly introduce students to fundamental programming concepts. In particular, we focus on how fundamental programming concepts can specifically be mapped to their spreadsheet equivalents.

3 The Proposed Spreadsheet-Based Approach

Values, data types, input and output data, variables, control structures, functional abstractions, data structures and syntax are the common fundamental concepts of many traditional programming languages [11, 20]. In this section, we illustrate how these fundamental concepts of programming can be mapped to their spreadsheet equivalents.

3.1 Values and Data Types

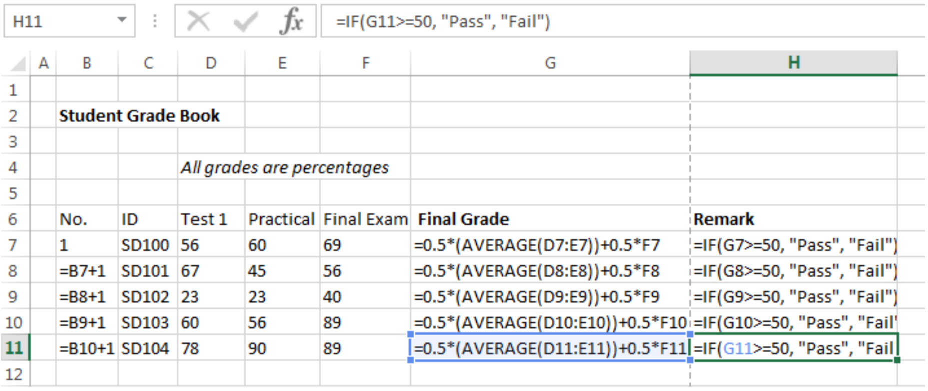
A value may be defined as any data that may be processed by a computer pro-gram [20]. For example, a number such as 10, a character such as ‘K’, a string such as “John”, are values. Each value, however, belongs to a type, where a type is simply a set of possible values that are related by the fact that they can be processed by similar operations [20]. Hence, programming languages support the concept of data types. In a spreadsheet, which is essentially a grid of columns and rows that form cells, one can enter either labels, numeric values and formulas in cells. Although not strongly-typed as traditional programming languages, spreadsheets allow numeric values and formulas to be formatted to conform to a particular data type. In the spreadsheet in Fig. 1, cells in row 6 are labels, while contents of cell range from C7 to F11 are values. On the other hand, cells from G7 to H11, are formulas. The formula view of the spreadsheet is also presented in Fig. 2.



H11 =IF(G11>=50, "Pass", "Fail")

	A	B	C	D	E	F	G	H
1								
2		**Student Grade Book**						
3								
4				*All grades are percentages*				
5								
6		No.	ID	Test 1	Practical	Final Exam	**Final Grade**	**Remark**
7		1	SD100	56	60	69	64	Pass
8		2	SD101	67	45	56	56	Pass
9		3	SD102	23	23	40	32	Fail
10		4	SD103	60	56	89	74	Pass
11		5	SD104	78	90	89	87	Pass
12								

Fig. 1. An example Student Grade Book spreadsheet where student grades are captured. Continuous assessment consists of “Test 1” and “Practical”. “Final Grade” is calculated from 50% of the average of continuous assessment and 50% of the “Final Exam”. If a student's final grade is greater or equal to 50%, a remark “Pass” is recorded, otherwise a “Fail” is recorded.



H11 =IF(G11>=50, "Pass", "Fail")

	A	B	C	D	E	F	G	H
1								
2		**Student Grade Book**						
3								
4				*All grades are percentages*				
5								
6		No.	ID	Test 1	Practical	Final Exam	**Final Grade**	**Remark**
7		1	SD100	56	60	69	=0.5*(AVERAGE(D7:E7))+0.5*F7	=IF(G7>=50, "Pass", "Fail")
8		=B7+1	SD101	67	45	56	=0.5*(AVERAGE(D8:E8))+0.5*F8	=IF(G8>=50, "Pass", "Fail")
9		=B8+1	SD102	23	23	40	=0.5*(AVERAGE(D9:E9))+0.5*F9	=IF(G9>=50, "Pass", "Fail")
10		=B9+1	SD103	60	56	89	=0.5*(AVERAGE(D10:E10))+0.5*F10	=IF(G10>=50, "Pass", "Fail'
11		=B10+1	SD104	78	90	89	=0.5*(AVERAGE(D11:E11))+0.5*F11	=IF(G11>=50, "Pass", "Fail
12								

Fig. 2. A formula view of the example Student Grade Book spreadsheet.

Spreadsheet systems such as Microsoft Excel allow different data types such as number, currency, accounting, date, time, fraction and text. In the example Student Grade Book spreadsheet in Fig. 1, data in the “Final Grade” column has been formatted to the data type number with zero decimal places.

3.2 Input and Output Data

Data in traditional programs may also be classified as input data and output data. Input data is raw data that is not yet processed while output data is data that is the result of computation by a computer program. In spreadsheets, users input data in some cells and this data is processed by being referenced in formula cells which subsequently produce output data. For example, in the spreadsheet in Fig. 1, cells D7, E7 and F7 contain input data while the data in cell G7 is output data. Students can therefore become aware of the concept of input and output data in programs through working with spreadsheets.

3.3 Variables

A variable in traditional programming languages may be defined as a storage location and associated symbolic name which contains some value [20]. The equivalent of a variable in a spreadsheet is a cell. Cells can be referenced either using the popular A1 notation or the R1C1 notation. For example, in the spreadsheet in Fig. 1, cell C7 holds the text value “SD100” while cell D7 holds the numeric value 56. On the other hand, cell H11 has a formula =IF(G11 >= 50, “Pass”, “Fail”).

Variables in programming languages can also be assigned to expressions to be evaluated. The expressions in turn can also contain other variables and values. In spreadsheets, formulas are used to assign expressions that need to be evaluated. In spreadsheet systems such as Microsoft Excel, the assignment operator (“=”) denotes the beginning of a formula expression. The expressions in formulas can also reference other cells. For example, in the formula view of the example Stu-dent Grade Book spreadsheet in Fig. 2, cell B8 has a formula =B7+1 (referencing cell B7) while cell G7 has the formula =0.5*(AVERAGE(D7:E7))+0.5*F7 (and thus referencing cells D7, E7 and F7). The formula expressions are also evaluated in a particular order. Just as in the formula in cell G7, the order of evaluation has been guided through the parentheses. In general, just as variable assignment in traditional programming languages, the form of a formula is given by **= operand *operator* operand** where the operands can be numbers, cell references, or functions while operators can be arithmetic, comparison, text concatenation, and reference.

3.4 Control Structures

A control structure in traditional programming languages is a block of a program that analyzes expressions and chooses a direction of execution based on the resulting boolean values [20]. Program statements are normally executed sequentially (one after the other) but decision and repetition expressions are used to alter the flow of execution of a program. Spreadsheets also support decision expressions and to a limited extent also support repetition structures. For example, the decision spreadsheet formula =IF(G11 >= 50, “Pass”, “Fail") is equivalent to the Java code

```
if (finalGrade >= 50)
    System.out.println(“Pass”);
else
    System.out.println(“Fail”);
```

Spreadsheet systems such as Microsoft Excel are Turing complete [7] hence support looping or repetition. However, in Microsoft Excel, this feature is disabled by default since it involves producing cycling data flows (a cell referencing itself) which normally is alerted as an error in a spreadsheet. Thus it is possible to introduce the concept of control structures using spreadsheets, particularly through decision formulas and limitedly through looping formulas.

3.5 Functional Abstraction

A function in traditional programming languages may be defined as a group of program statements that perform a particular task [20]. As such, a function may be called from time to time from other parts of a program to perform a particular task provided appropriate task parameters are given. In spreadsheets, functions may not necessarily appear to the user as in traditional programming languages. Nevertheless, spreadsheets provide numerous already defined functions which can be called in formulas. For example, in the formula view of the example Grade Book spreadsheet given in Fig. 2, the formula in cell G7 is given by =0.5*(AVERAGE(D7:E7))+0.5*F7 whereby AVERAGE is a system defined function that calculates the average of the numbers in a given range, in this case the cell range D7 to E7. The concept of a function can therefore be indirectly introduced to beginning programmers through spreadsheets hence preparing them for a more formal introductory programming course.

3.6 Data Structures

A data structure may be defined as a way of storing and organizing data in such a way that it can be used efficiently [20]. Arrays, lists and maps are some of the common data structures available in many programming languages. It is thus imperative that a novice programmer be introduced to these very important concepts at the foundation stage. But again, these concepts can be intimidating to beginning programmers. A spreadsheet, on the other hand, though with an easy to use interface, can also be viewed as a massive data structure that is represented as a grid of columns and rows which form cells. And as already discussed above, spreadsheet cells can hold values of different data types. In spreadsheets, a range of cells containing values of the same data type can be considered as an equivalent of an array in traditional programming languages. On the other hand, a range of cells containing values of different data types can be considered as an equivalent of a list in traditional programming languages. Sorting and searching are common operations on data structures and spreadsheets also offer other sorting and searching functions over various cell ranges. For very large spreadsheets, spreadsheet systems such as Microsoft Excel, provide search functions such as lookup functions and pivot tables to improve efficiency of search over particular cell ranges. Thus through sorting and searching over particular cell ranges in spreadsheets, one can indirectly demonstrate the concept of data structures and their associated operations.

3.7 Syntax

The syntax of a programming language is the set of rules that define the combinations of symbols that are considered to be correctly structured programs in that language [19]. The syntax of a programming language, therefore, denotes how variables, data types, control structures, functions and other language constructs can be woven together to form programs which are considered legal by the programming language. The ability to master a syntax of a programming language is one of the required skills of a programmer [5, 8]. Novice programmers, should therefore be trained in the art of mastering syntax of a programming language. Formulating a problem solution in spreadsheets also requires one to follow syntactical rules of spreadsheets such as how to specify formulas. Thus by practising to express solutions in spreadsheets, novice programmers are prepared to master a skill that will help them to follow the syntactical rigour of a traditional programming language.

4 Conclusion

We have demonstrated that many fundamental programming concepts have their equivalents in spreadsheets. A summary of the mappings is given in Fig. 3. We are thus proposing that before students take a formal introductory programming course, they should first be taught spreadsheet programming where fundamental programming concepts, as above, are indirectly introduced to them through problem solving using spreadsheets. We, however, also take note of the limitations of this proposed approach. First, most traditional programming languages used to introduce students to programming are imperative in nature, hence program statements are given one after the other and correspondingly executed sequentially. On the other hand, the spreadsheet paradigm is not imperative. Data and formulas are entered in cells not necessarily in particular order and the spreadsheet engine calculates and propagates changes to spreadsheet state accordingly. Nevertheless, we find this not detrimental to our proposed approach because what we are emphasizing are the equivalent fundamental concepts of programming as applied in spreadsheets and not necessarily how execution sequence is done. Moreover, declarative programming languages such as spreadsheets focus on "what" must be accomplished in terms of the problem domain and rather than "how" to accomplish a solution to a problem as an explicit sequence of programming language primitives [9]. This could be advantageous to novices as they can just concentrate on the logic of the task at hand [9] while indirectly learning programming constructs.

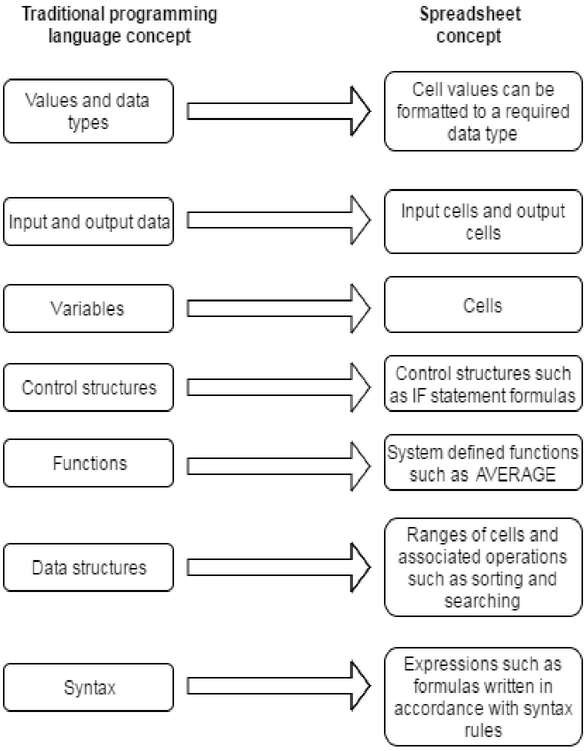

Traditional programming language concept
Spreadsheet concept
Values and data types
Cell values can be formatted to a required data type
Input and output data
Input cells and output cells
Variables
Cells
Control structures
Control structures such as IF statement formulas
Functions
System defined functions such as AVERAGE
Data structures
Ranges of cells and associated operations such as sorting and searching
Syntax
Expressions such as formulas written in accordance with syntax rules

Fig. 3. A summary of the mapping of some traditional programming language concepts to their spreadsheet equivalents.

Second, spreadsheets do not properly support looping control structures. This could be motivation to explain to students why for some problems, we need to program using traditional programming languages. Third, spreadsheets are also well known to be error prone [1]. The spreadsheet programming course therefore also provides an opportunity for students to be sensitized of this vulnerability of spreadsheets. The fourth limitation of our approach is that we have also not yet conducted empirical studies on the efficacy of our approach. This will be part of our future work.

References

1. Abraham, R., Burnett, M., Erwig, M.: Spreadsheet programming. Wiley Encyclopedia of Computer Science and Engineering (2008).

2. Beaubouef, T., Mason, J.: Why the high attrition rate for computer science students: some thoughts and observations. ACM SIGCSE Bulletin 37(2), 103-106 (2005).
3. Brusilovsky, P., Kouchnirenko, A., Miller, P., Tomek, I.: Teaching programming to novices: A review of approaches and tools. (1994).
4. Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., Miller, P.: Mini-languages: a way to learn programming principles. Education and Information Technologies 2(1), 65-83 (1997).
5. Denny, P., Luxton-Reilly, A., Tempero, E., Hendrickx, J.: Understanding the syntax barrier for novices. In: Proceedings of the 16th annual Joint Conference on Innovation and Technology in Computer Science Education. pp. 208-212. ACM (2011).
6. Gomes, A., Mendes, A.J.: Learning to program-difficulties and solutions. In: International Conference on Engineering Education (ICEE). (2007).
7. Hermans, Felienne: Excel Turing Machine (2013), http://www.felienne.com/archives/2974, last accessed 2017/05/15.
8. Jenkins, T.: On the difficulty of learning to program. In: Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences. vol. 4, pp. 53-58 (2002).
9. Lloyd, J.W.: Practical Advantages of Declarative Programming. In: Joint Conference on Declarative Programming, GULP-PRODE'94. pp. 18-30 (1994).
10. Lovaszova, G., Hvorecky, J.: On programming and spreadsheet calculations. Spreadsheets in Education (eJSiE) 1(1), 3 (2003).
11. Lubitz, W.D.: Rethinking the first year programming course. Proceedings of the Canadian Engineering Education Association (2011).
12. Martin, A.: An integrated introduction to spreadsheet and programming skills for operational research students. Journal of the Operational Research Society 51(12), 1399-1408 (2000).
13. McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y.B.D., Laxer, C., Thomas, L., Utting, I., Wilusz, T.: A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. ACM SIGCSE Bulletin 33(4), 125-180 (2001).
14. Nardi, B.A., Miller, J.R.: The spreadsheet interface: A basis for end user programming. Hewlett-Packard Laboratories (1990).
15. Nardi, B.A., Miller, J.R.: Twinkling lights and nested loops: distributed problem solving and spreadsheet development. International Journal of Man-Machine Studies 34(2), 161-184 (1991).
16. Powers, K., Ecott, S., Hirshfield, L.M.: Through the looking glass: teaching CS0 with Alice. ACM SIGCSE Bulletin 39(1), 213-217 (2007).
17. Robins, A., Rountree, J., Rountree, N.: Learning and teaching programming: A review and discussion. Computer science education 13(2), 137-172 (2003).
18. Shackelford, R.L., LeBlanc, R.: Introducing computer science fundamentals before programming. In: Frontiers in Education Conference, 1997. 27th Annual Conference. Teaching and Learning in an Era of Change. Proceedings. vol. 1, pp. 285-289. IEEE (1997).
19. Slonneger, K., Kurtz, B.L.: Formal syntax and semantics of programming languages, vol. 340. Addison-Wesley Reading (1995).
20. Strachey, C.: Fundamental concepts in programming languages. Higher-order and symbolic computation 13(1-2), 11-49 (2000).
21. Tahy, Z.S.: How to teach programming indirectly-using spreadsheet application. Acta Didactica Napocensia 9(1), 15 (2016).
22. Warren, P.: Teaching programming using scripting languages. Journal of Computing Sciences in Colleges 17(2), 205-216 (2001).
23. Warren, P.: Learning to program: spreadsheets, scripting and HCI. In: Proceedings of the Sixth Australasian Conference on Computing Education-Volume 30. pp. 327-333. Australian Computer Society, Inc. (2004).

Critical Reflection on a Data Warehouse/Business Intelligence Course: Does It Prepare Students to be Successful Practitioners?

Carin Venter

North-West University, Vanderbijlpark, South-Africa
Carin.Venter@nwu.ac.za

Abstract. This paper conceptually reflects on an elective postgraduate data warehousing/business intelligence course offered at a university; the aim is to determine whether it sufficiently prepares students to become successful practitioners, in this field, upon entering industry. The reflection is guided by a critical systems thinking approach, i.e. critical social heuristics. The researcher reflects on: the basis of motivation; the basis of power; the sources of knowledge; and the sources of legitimation that are relevant to this course in terms of the 'as-is', i.e. actual scenario, and the 'ought to be', i.e. the ideal scenario. The researcher reflects on probable shortcomings in the course and whether they hinder students' abilities to develop successful data warehouse/business intelligence systems; she also suggests improvements. She then proposes next steps to empirically test the findings.

Keywords: Data Warehousing; Business Intelligence; Critical Systems Thinking; Critical Social Heuristics

1 Introduction

The quality of information determines the quality of decisions in organisations; it ultimately determines an organisation's competitive advantage [32]. The aim of business intelligence (BI) is to improve strategic decisions in organisations. BI is enabled by a technological platform, i.e. a data warehouse (DW). Organisations implement expensive data warehouse/business intelligence (DW/BI) systems to improve decision making capabilities [46]. These DW/BI systems are designed and developed by DW/BI practitioners.

Future DW/BI practitioners must acquire specific skills to be ready for industry. Industry require practitioners that can develop efficacious systems that end-users accept and use. However, high failure rates indicate, at the least, a partial inability of current practitioners to develop successful systems [7, 10, 11, 15]. A lot of current practitioners (including those working in DW/BI) received tuition at an academic institution [27]. So, there is an apparent gap between students' academic preparation and what is required from them as practitioners in industry [37, 45].

Academic institutions must continuously reflect on what is required from these practitioners, in order to respond timeously to create the next generation workforce [45]. It is not possible to teach students all the latest technological trends, especially in ever-evolving fields such as IT and DW/BI. Yet, it remains crucial for academic institutions to keep abreast of key new developments and respond timeously to relevant requirements from industry. Academics should, for example, attempt to determine reasons behind current failures in the fields that they teach, and then evaluate alternatives to improve upon the status quo.

This paper conceptually explores shortcomings in a postgraduate DW/BI course taught to students at a South African university; the researcher also reflects on possible causes of DW/BI failures in industry. The researcher reflects on new developments in the DW/BI field, and whether these may overcome identified shortcomings and improve DW/BI success; she also comments on probable enrichment of the current DW/BI course through these.

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

The researcher uses a holistic research approach from Checkland and Holwell [4] to guide the reflection in this study; they suggest that an identified area of concern be investigated using a methodology that embodies a particular philosophical framework, i.e. a linked set of ideas. The researcher applies critical social heuristics (CSH), which embodies ideas from the critical systems thinking (CST) paradigm, to guide her reflection process.

The paper is organised as follows: The structure of this study is discussed in Section 2. Section 3 gives an overview of the key concepts in the study. In Section 4 the researcher reflects on possible reasons behind DW/BI failures, as well as new developments in the DW/BI field to potentially overcome them. Section 5 discusses future research. Lastly, a summary is given in Section 6.

2 The Structure of the Research

This study is structured according to the work of Checkland and Holwell [4] – they argue that holistic and reflective research entails three elements: a methodology (M) that embodies a particular linked set of ideas, i.e. a framework (F), that can be applied to investigate a specific instance of an identified area of concern (A); this is referred to as the FMA framework. The FMA framework is illustrated in **Figure 1** below. Application of the FMA framework in this study is discussed next.

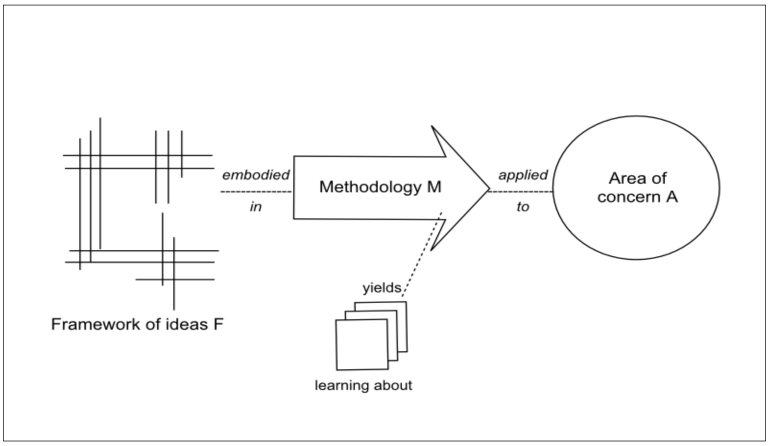


Figure 1. FMA Illustration: Elements in Research [4]

The Area of Concern. An area of concern (A) constitutes a "real-world situation which seems relevant to research themes which [a researcher] regards as significant" [4]. The theme for this study is a perceived gap between students' academic preparation at universities and industry's requirements for students when they become part of the active workforce; this conceptual study focuses specifically on an elective postgraduate DW/BI course taught at a university. The DW/BI course aims to prepare students to become successful DW/BI practitioners; these students must be taught applicable content and DW/BI methodologies to enable them to become successful practitioners that design and develop efficacious DW/BI systems for end-users. To achieve this, courses should timeously incorporate relevant new developments; academics should thus identify key new developments in fields of study in order to improve curriculums.

In this study, the real-world situation of this research theme is as follows: The researcher is concerned about high failure rates in DW/BI; she explores reasons behind it, and key developments in the field that may improve success rates of DW/BI systems, to identify how a postgraduate DW/BI course currently offered at a university can be improved. The researcher currently teaches this course at one of the campuses of a university. She also has industry experience in the field if DW/BI. One of her research focus areas is the improvement of DW/BI success – refer for example to a study that reflects on requirements analysis for successful DW/BI using CSH [42].

The Framework of Ideas. Reflective practice necessitates the inclusion of practices to identify and explore relevant assumptions/questions about the problem being investigated, as well as the kind of change required to bring about improvement [40]. In this study, the identified area of concern is reflected upon within the holistic framework (F) of the critical systems thinking (CST) paradigm. Refer to Section 3.2 for a discussion on CST.

The Methodology. Critical social heuristics (CSH) is a reflective methodology positioned in the CST paradigm; CSH enables reflection on the (problematical) status quo in order to ultimately enable improvement [41]. CSH is applied as the methodology (M) to guide the reflection in this study. Refer to Section 3.3 for a discussion on CSH.

3 Key Concepts in the Study

The key concepts in this study are data warehousing/business intelligence, as well as critical systems thinking and critical social heuristics. These are discussed next to create a shared understanding.

3.1 Data Warehousing/Business Intelligence

The ultimate goal of business intelligence (BI) is to improve decision making. The technological infrastructure employed to achieve this is the data warehouse (DW). The terms DW and BI are often used interchangeably in literature and by practitioners; however, they are interlinked and should not be used independent of each other. Inmon [16] defines a DW as a *"subject oriented, integrated, nonvolatile and time variant"* data source that is to be used for *"management's decision processes"* and says that management's decisions are supported by business intelligence. Kimball [23] also refers to a DW as *"the platform for all forms of business intelligence*". He defines business intelligence as a "generic term to describe leveraging the organization's internal and external information assets to support improved business decision making" [22]. As per the definition of Kimball [22], the combined phrase "*data warehouse/business intelligence (DW/BI)*" will be adopted in this paper to refer to "the complete end-to-end system" that supports decision making; this reinforces dependency between these concepts.

3.2 Critical Systems Thinking

The aim of CST is to facilitate social improvement through social and critical awareness, complementarism and a dedication to emancipation [8]. CST "seeks to promote a transformational as opposed to an affirmative account of the social world" [6]. The CST paradigm combines philosophy with social science; it build on the work of critical social theorists such as Kant and Habermas [1, 20]. Their work are concerned with emancipation of people from societal repression and the possibility of rational consensus through appropriate communication [29].

Researchers in the CST paradigm appreciate that the value of social science resides in its potential to develop conditions to benefit human through application of critical powers of reason, where the aim is to advance emancipatory change and subsequently liberate people from oppressive social structures, such as unnecessarily restrictive identity formations, assumptions, traditions and power relations; such oppressive social structures restrain opportunities for autonomy and the clarification of genuine needs, and ultimately hinders greater and lasting gratification amongst individuals and collective groups [1]. The CSH methodology was developed to operationalise notions of the CST paradigm [40] – it is discussed next.

3.3 Critical Social Heuristics

The CSH methodology enables reflection to critique in order to recognise underpinning boundary judgements and associated normative implications of an area of investigation; it also enables reflection to ensure rational application of improvement actions within justified boundaries [38]. CSH enables problem solvers to holistically determine what *is/has been* done (the *as-is* scenario), as well as what *ought to be* done (the *ideal* scenario) [8].

Ulrich [40] asserts that CSH is based on practical philosophy. It builds on the work of philosophers such as Kant and Habermas that were concerned with the ideal concept of rationality, i.e. rational identification and justification of the normative content of actions [40]. However, to make CSH practical, rather than theoretical, it also embodies ideas from the field of systems thinking where systems are regarded as social constructs in society; for example, a system may be a problematical social context to be reflected upon, with the aim to improve it and emancipate involved and/or affected stakeholders [40].

Firstly, CSH aims to identify boundary judgements of entities referred to as *systems* in social reality; such a social system (S) is a problematical social context that is bounded by two distinct groups of social actors, i.e. those that are *involved* in its design as well as those that are merely *affected* by its design. The *involved* and *affected* stakeholders, and the system's *environment*, define the normative content of S [40] – illustrated in **Figure 1** below.

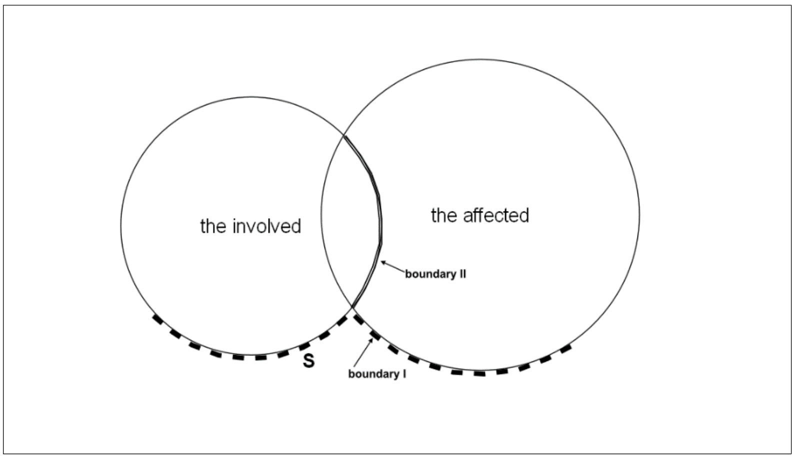


Figure 2. Basic kinds of boundary judgements for system (S) [40]

Secondly, to determine the boundary judgements of S for the *involved* actors, Ulrich [40] employs Churchman's conceptualisation of inquiring systems where a purposeful social system has the following nine conditions/ characteristics: it is teleological, i.e. it exists to serve the *purpose* that it was created for; it has a *measure of performance*; it exists to serve a *client*; it has teleological *components* that coproduce its expected performance; it has a clearly defined (bordered) *environment*; it has a *decision maker* that can change the system; it has a designer (*planner*) with the intention to maximise the system's value for its client; it is *implemented* in its environment; and the system is assumed (*guaranteed*) to be stable [40]. Accordingly, the nine conditions are applied to create nine categories to trace the normative content of the design of S for the involved stakeholders to determine: the sources that *motivate* the system, i.e. its *client*, *purpose* and *performance (improvement) measure*; the sources that *control* the system, i.e. its *decision maker*, *components* and *environment*; and the sources that have relevant *knowledge/expertise* regarding the system, i.e. its *planner*, *environment* and *guarantor* [40].

Thirdly, Ulrich [40] reasons that the normative content of a social system's design can only be legitimately reflected upon when the views of those that are *affected* it (yet have not been involved during its design) have been incorporated. So, he added a self-reflective (*legitimising*) dimension with three additional categories for those that are *affected* by S, i.e. a *witness* that is concerned with the *emancipation* of those that are affected (but not involved); the witness is also concerned with the fundamentally different (conflicting) *worldviews* of the involved and affected stakeholders [40].

Lastly, Ulrich [39, 40] applied the twelve categories described above, and devised twelve boundary questions to determine: the boundary judgements and normative content of a social system; when asked in the 'is-mode', the answers reveal the *as-is* scenario and when asked in the 'ought to-mode' the answers reveal the *ideal* scenario:

- "Who is (ought to be) the **client** or beneficiary? That is, whose interests are (should be) served?"
- "What is (ought to be) the **purpose**? That is, what are (should be) the consequences?"
- "What is (ought to be) the **measure of improvement** or measure of success? That is, how can (should) we determine the consequences, taken together, constitute an improvement?"
- "Who is (ought to be) the **decision maker**? That is, who is (should be) in a position to change the measure of improvement?"
- "What **resources** and other conditions of success are (ought to be) controlled by the decision maker? That is, what conditions of success can (should) those involved control?"
- "What conditions of success are (ought to be) part of the **decision environment**? That is, what conditions can (should) the decision-maker *not* control (e.g. from the viewpoint of those not involved)?"
- "Who is (ought to be) considered a **professional** or further **expert**? That is, who (should be) involved as competent provider of experience and expertise?"
- "What kind of **expertise** is (ought to be) consulted? That is, what counts (should count) as relevant knowledge?"
- "What or who is (ought to be) assumed to be the ***guarantor*** of success? That is, where do (should) those involved seek some guarantee that improvement will be achieved – for example, consensus among experts, the involvement of stakeholders, the experience and intuition of those involved, political support?"
- "Who is (ought to be) **witness** to the interests of those affected but not involved? That is, who is (should be) treated as a legitimate stakeholder, and who argues (should argue) the case of stakeholders who cannot speak for themselves, including future generations and non-humans?"

- “What secures (ought to secure) the **emancipation** of those affected from the premises/promises of those involved? That is, where does (should) legitimacy lie?”
- “What **worldview** is (ought to be) determining? That is, what different visions of ‘improvement’ are (should be) considered and how are they (should they be) reconciled?”

The twelve boundary questions can be grouped into four main categories that enable reflection to determine: what aspects of a situation are to be considered relevant, who should be involved in determining it, who are affected by it and how to handle conflicting views amongst relevant stakeholders; these four main categories are descriptive of the normative content of a social system in terms of its basis of *motivation*, its basis of *power*, its sources of *knowledge*, and its basis of *legitimacy* [40]: First, the basis of *motivation* indicates the *clients/beneficiaries* that are the sources of motivation of the system to be designed; they are involved during system design, concerned with the *purpose* of the system and interested in possible *improvements* by means of the system. Second, the basis of *power* indicates the *decision makers* as the sources of control; they are involved during system design to determine sources of control within (the *components*) as well as beyond (the *environment*) of the system and account for the way in which system improvements depend on both. Third, the basis of *knowledge* indicates the *planners* that are the *sources of knowledge*, experience and/or skill; they are the implementers of the system that should aim to *guarantee* its success. Fourth, the basis of *legitimacy* indicates the *witnesses* to those that are affected by the system but would not usually be involved in its design or implementation; they are a potentially large group that hold the three involved stakeholders mentioned above ethically responsible and may require *emancipation*, yet may have conflicting *worldviews* amongst them that need to be reconciled.

4 Reflection on the DW/BI Course

In this section, the researcher applies the twelve boundary questions to conceptually reflect on an elective DW/BI course offered at a university in the ‘is-mode’ (the *as-is* scenario) and attempt to identify shortcomings; the researcher also reflects on key developments in the field, as per the literature, in the ‘ought to-mode’ (the *ideal* scenario) to determine possible improvements. The course is part of a postgraduate programme offered to full- and part-time students at the university; it is offered part-time to also allow working/employed students to partake. The assumption is that students that take the DW/BI elective course are potentially interested to become successful DW/BI practitioners; or work in the DW/BI field and want to improve their skills so as to enhance their careers. The discussion is structured according to the four main categories that encapsulate the boundary questions.

4.1 The Basis of Motivation: the Client/Beneficiary, Purpose and Measure of Improvement/Success

The question regarding the client/beneficiary in the ‘is-mode’ reveals whose interests are currently being served by the course; reflection in the ‘ought to-mode’ reveals whose interests should be served by the course. Then, the question regarding the purpose in the ‘is-mode’ reveals what motivates the design (and existence) of the course; reflection in the ‘ought to-mode’ reveals what should motivate the course design and existence. Lastly, the question regarding the measure of improvement/success in the ‘is-mode’ reveals how success is being measured for the course; reflection in the ‘ought to-mode’ reveals how success should be measured.

The interests of the students and industry ought to be served in that the course content must be relevant; students should be able to confidently enter the job market upon completion, or enhance their careers. Students must be enabled to become successful and productive DW/BI practitioners in the industry.

The DW/BI course is currently offered as part of a postgraduate (Honours) degree at a university. It aims is to teach students, as the current/next generation workforce, DW/BI principles; at the end of the course students must be able to design and develop a DW/BI system inclusive of a DW, a browser, online analytical processing (OLAP) cubes, a BI dashboard as well as suitable technical and end-user documentation. They must also be able to query the DW and generate BI reports. The students must use dimensional modelling principles, as per the Kimball lifecycle approach [22], to design and develop the DW/BI system; the Kimball approach is discussed as a source of knowledge in Section 4.3.

The Kimball approach and dimensional data modelling is currently regarded as the ‘gold’ standard in the industry; however, there are new developments in the field such as data vault modelling that is growing in popularity [25, 26]; data vault modelling is thus also discussed as a possible source of knowledge in Section 4.3.

Success (or failure) of students are currently measured by whether they have acquired sufficient theoretical and practical knowledge to pass theoretical and practical examinations, as per the requirements of The South African Qualifications Authority (SAQA) [33]. The course has a high pass rate. However, the success of the course ought to include an additional measure; it ought to be reflected in the number of students entering the workforce as DW/BI practitioners that can design/implement good and practical DW/BI systems, as per the requirements of end-users (i.e. that are accepted and used by end-users). A third stakeholder group can thus also be identified, i.e. the (affected) end-users of DW/BI systems – they should receive DW/BI systems that fulfil their requirements. End-users and end-users' business requirements, as sources of legitimisation, are discussed in Section 4.4.

4.2 The Basis of Power: the Decision Maker, Resources and Conditions of Success

The question regarding the decision maker in the 'is-mode' reveals who is in a position to change the measure of improvement/success for the course; reflection in the 'ought to-mode' reveals who should be in a position to change the measurement of improvement/success. Then, the question regarding the resources/success conditions in the 'is-mode' reveals what resources/success conditions are being controlled by the identified decision maker of the course; reflection in the 'ought to-mode' reveals the resources/success conditions that should be controlled by the decision maker. Similarly, the question regarding the environment in the 'is-mode' reveals the conditions that the course's decision maker cannot control; reflection in the 'ought to-mode' reveals the conditions that the decision maker should not control from the perspective of those that are not involved.

The course adheres to the criteria for an Honours year level course, i.e. National Level 8, as defined by The South African Qualifications Authority (SAQA) [33]. SAQA is a governing body and should be appreciated as such; the existing performance measures in terms of theoretical and practical examinations will therefore remain as-is.

It is, however, also important to be cognisant that failure rates of software and information technology (IT) systems, such as DW/BI, are very high [7, 10, 11, 15]. Academic institutions do have control over the course content; the content of the course may therefore be enriched to improve future DW/BI success. The course designer should, though, be mindful that the course is a 24-credit module; this implies that students should spend a total of 240 study hours in a period of one year of study. New content should not readily be added without taking into account the time that the student will be expected to spend learning added content.

4.3 The Sources of Knowledge: the Professional/Expert, Relevant Expertise and Guarantor

The question regarding the professional/expert in the 'is-mode' reveals who are/were involved to design and implementation the course; reflection in the 'ought to-mode' reveals who should be involved. Then, the question regarding the kind of expertise consulted in the 'is-mode' reveals the kind of expertise that is/was consulted during the design and implementation of the course, and what count/counted as knowledge during the course design and implementation; reflection in the 'ought to-mode' reveals what should count as knowledge. Lastly, the question regarding the guarantor in the 'is-mode' reveals where those that are/were involved in the course design seek guarantee for the promised improvement (refer also the discussion on the measure of improvement/success earlier in this section); reflection in the 'ought to-mode' reveals where they should seek guarantee.

The DW/BI course is based on the Kimball lifecycle approach, as explained in the book "The Data Warehouse Lifecycle Toolkit" [22]; this is also the prescribed text book for the course. However, there are three main 'schools of thought' that are applicable in the field of DW/BI, i.e. the Kimball lifecycle approach [23]; Inmon's corporate information factory (CIF) [19]; and Linstedts' data vault model [25]. The researcher reflects on these three approaches next in terms of the basis of motivation that was defined for the course – refer also to Section 4.1.

The Kimball Approach. The Kimball approach uses dimensional modelling, rather than data vault models or the more traditional relational modelling, to create the DW. Dimensional models are also referred to as star-schemas; they have a star-like structure where a single 'fact table' is surrounded by many 'dimension tables' [22]. Fact tables hold transaction-specific data, e.g. the transaction amount and invoice number, as well as foreign keys from associated dimension tables for reference to descriptors of transaction data. Dimension tables contain descriptive detail on, for example, a customer, product and all potential date/time entries; dimension tables hold all the descriptive data related to the transaction records to give context [22].

Kimball [22] argues that dimensional data models are "the logical foundation for all OLAP system" since they are "structured to deliver maximum query performance and ease of use"; star-schemas "proved to be understandable, predictable, extendable, and highly responsive to ad hoc demands"; and provide "powerful indexing approaches and aggregations". The advantages of dimensional modelling is acknowledged by industry; hence, it is often used by industry to model the DW [12]. However, the dimensional modelling technique also have some shortcomings e.g. data redundancy, large data sets that are sometimes difficult to maintain in changing source environments, significant amount of time spent to extract, transform and load (ETL) data from source to target as well as lack of data traceability from source to target [12].

The Kimball approach is 'richer' in terms of the explicit inclusion of business requirements prior to the development of infrastructure; it includes activities to gather business requirements, as well as design and develop the DW architecture [22]. In the Kimball approach the DW/BI system is developed in three parallel tracks, i.e. the technical track, the data track and the BI application track – this is illustrated in **Figure 3** below.

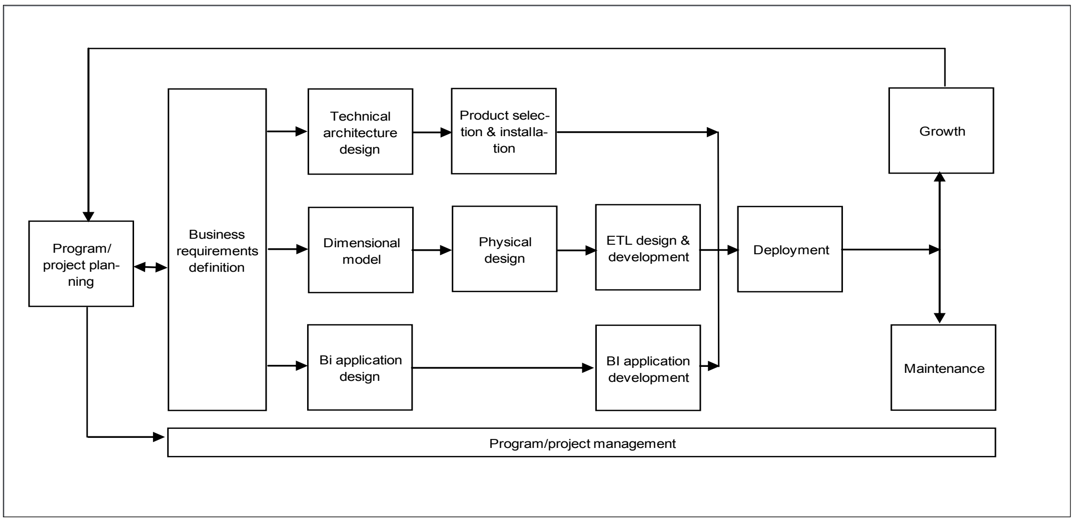

Growth
Technical architecture design
Product selection & installation
Program/ project planning
Business requirements definition
Dimensional model
Physical design
ETL design & development
Deployment
Bi application design
BI application development
Maintenance
Program/project management

Figure 3. The Kimball Lifecycle Diagram [23]

Software development success, in general, is driven by the degree to which the designer/developer understands the end-users' business requirements [24]. The principal problem areas in software development, thus the reasons for unsuitably produced software, are the inappropriate specification and management of customer's requirements i.e. requirements do not reflect real customer needs, inconsistent and/or incomplete requirements, expensive late changes to requirements as well as misunderstandings between customers, requirements analysts and software developers [34]. Designers/developers therefore require an accurate understanding of end-users' requirements to be able to deliver quality systems that are acceptable to them [13, 28]. Keating *et al.* [21] argue that, since user adoption is a crucial indication of success, appropriate collection of end-users' requirements is the most critical part of development. Appropriate requirements collection is necessary to avoid that "those specifications were based upon the designers' ignorance before they started the job" [30].

Since requirements are gathered prior to design and development, the Kimball approach should enable development of DW/BI that are acceptable to end-users. Kimball and Ross [23] state that "understanding the business requirements is the most fundamental and far reaching" aspect of DW/BI development. They explain that business requirements include key performance indicators that drive organisational decisions; user communities' abilities to use applications such as spreadsheets and reporting tools; and user expectations in terms of implementation time horizons. They suggest "focusing on what business users do today…want to do in the future" and recognise that users' future requirements may be different from what they currently have [23].

Still, the requirements gathering process that the Kimball lifecycle prescribes seems to presume that users are able to accurately conceptualise future requirements and that it can be discovered through interviews [22, 23]. Contrariwise, BI users "operate in a mode of discovery…cannot tell what the information needs are until they see the possibilities" [17]; this implies that an evolutionary and agile design and development approach may yield more success. Kimball and Ross [23] recommend an incremental, agile BI development approach where business requirements are elicited "by focusing on what business users do… or want to do…" and to "give the users what they want". Still, the Kimball approach – as portrayed in **Figure 3** – is not an incremental or agile process; it is a sequential development approach that does not allow the designer/developer to cycle back and refine business requirements during the design/development stages.

Appropriate requirements specification involves more than merely designing the automation of current functions; implemented software must yield new functionalities and ultimately improvements in current work processes [31]. The introduction of a new system (such as DW/BI) will result in an adjusted social context for its users since it "stirs things up, introduces uncertainties, gets people perceiving their world in a new way" [4]. These systems are thus intricately part of the social structures where they are applied; design and development cannot happen in isolation but must be explored as part of the bigger social (organisational) system that it will ultimately belong to and in terms of the individuals that it will impact upon [30]. Appropriate requirements must therefore ultimately incorporate the human, social and organisational dimensions of the organisation, in addition to capturing technical requirements. Gathering of end-users' business requirements is further discussed as a source of legitimisation in Section 4.4.

The Corporate Information Factory. Inmon's CIF originally employed (the more traditional) relational and normalised entity-relationship (ER) modelling to model data with minimal redundancy [18]. Students are taught ER modelling when they learn basic data base skills – this a pre-requisite for the DW/BI course – and also in a (complementary) advanced data base course that is compulsory for the students to take with the elective DW/BI course. ER modelling, as a possible source for improvement, is therefore not further discussed in this paper. Furthermore, more recently, Inmon promoted the use of data vault modelling in the CIF [26]. Data vault modelling is discussed later in this paper.

With regard to the collection of business requirements, the CIF is claimed to be structured around the business and allows business users to extract centrally stored information [19]. However, the process does not explicitly make provision for business requirements to be gathered; it is also not clear exactly how the CIF supports business, other than incorporating all the data produced by business users into the DW and providing users with BI reports – the basic structure of the CIF is depicted in **Figure 4** below. The CIF is therefore also not further explored in terms potentially improving upon the requirements gathering approach of Kimball.

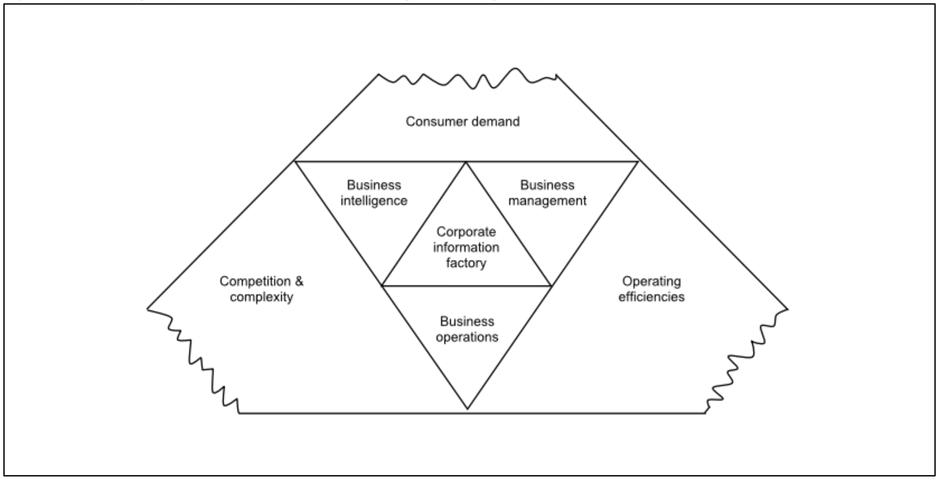


Figure 4. Basic structure of the CIF [19]

Data Vault Modelling. Data vault modelling, similar to dimensional modelling, is fundamentally different than classical ER modelling; According to Linstedt [26] data vault modelling separates the context of entities from their associations resulting from business rules. He explains data vault modelling as follows: Data are stored in 'hub tables' and 'satellite tables'; 'link tables' are used to link hubs and/or satellites together. Hub tables contain business keys (usually as natural keys) that are typically known to business users e.g. referring to a specific department such as the 'Engineering' or 'Finance' department [26]. Link tables indicate where business keys intersect and represent transactions or hierarchies; link tables do not store details, but merely represent relationships between tables. Satellite tables store descriptive data providing context to the hub and link tables [26]. Satellite tables differ from dimension tables in dimensional modelling as they may also represent detailed transaction entries, in addition to being descriptors of contextual data. All data vault tables also include fields indicating the source and load time of the record from its source [26]. So, data vault modelling is the only modelling technique that also incorporates data traceability, since the source of each data element is tracked in its target table. It also includes a metrics repository that collects statistical information regarding the DW growth/ usage patterns to optimise usage of hardware [25]. Data are stored in distributed base systems; hence, large data sets are more flexible [26].

Some of the dimensional modelling shortcomings (noted earlier in this section) are addressed by data vault modelling: Data stored in warehouses modelled using the data vault method are fully traceable; less redundant data are also stored in data vaults due to the many-to-many relationships between tables; and changes/evolutions in source systems can easily be incorporated since the data base structure is largely distributed [12]. Incorporation of these technical design principles can add value to enrich students' understanding of data modelling techniques.

With regard to business requirements, the gathering of business requirements is not explicitly included in the data vault method – refer to **Figure 5** below for a high-level illustration of a data vault. Data vaults are therefore also not further explored in terms of potentially improving upon the requirements gathering approach of Kimball.

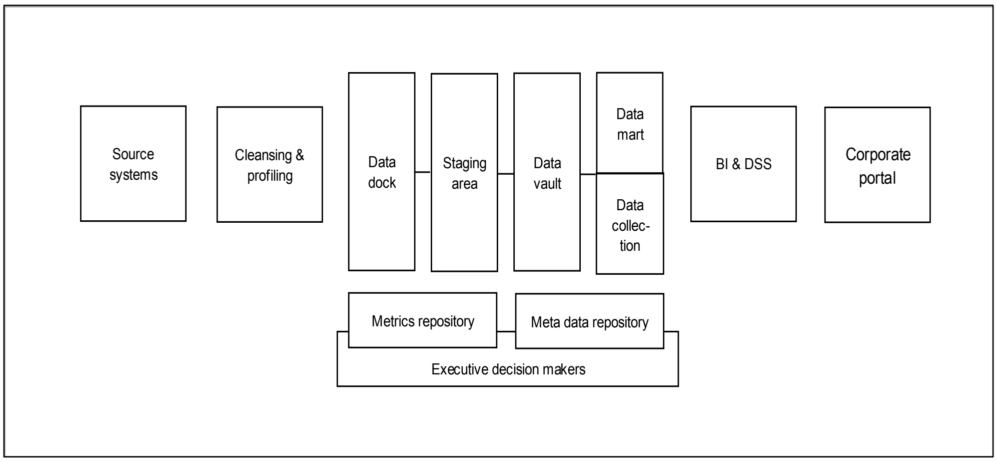


Figure 5. Linstedt's data vault method [25]

4.4 The Basis of Legitimacy: the Witness, Emancipation and Worldview

The question regarding the witness in the 'is-mode' reveals with whom legitimacy lies in terms of those affected by the premises/promises of those involved, i.e. who speaks/spoke for those that cannot speak/have spoken for themselves; reflection in the 'ought to-mode' reveals with whom legitimacy should lie. Then, **t**he question regarding the emancipation of those *affected* by the promises of the *involved* in the 'is-mode' reveals where legitimacy lies in terms of those affected by the premises/promises of those involved; reflection in the 'ought to-mode' reveals where legitimacy should lie. This question should firstly determine *what* constitutes emancipation in order to determine *how* to measure success. Lastly, the question regarding the worldview in the 'is-mode' reveals the different notions of improvement that are/were considered and reconciled; reflection in the 'ought to-mode' reveals the notions of improvement that should be considered and reconciled.

The term 'emancipation' may appear to be fairly grandiose; however, it may also be applied on a relatively small scale as *"microemancipation"* focusing on concrete activities and techniques that serve as vehicles for liberation where "processes of emancipation are understood to be uncertain, contradictory, ambiguous, and precarious" and "loopholes" can be found where power techniques are operational [1]. In such a case the emancipatory idea is then not portrayed as one large grandiose project, but rather as a group of smaller projects, where each smaller project is "limited in terms of space, time, and success" [1]. So, for the purpose of this (initial/conceptual) study, the researcher defines emancipation of students as: enabling students to develop efficacious DW/BI systems that are acceptable to (affected) end-users i.e. that incorporate end-users' specific business requirements.

Some technical weaknesses were identified in the current (dimensional modelling) approach that is taught to students – refer to Section 4.3; however, these do not seem significant enough to make fundamental differences in future DW/BI success rates as IT systems (including DW/BI) rarely fail due to technical infeasibility [2, 5]. It may be marginally beneficial to enrich the DW/BI course and also incorporate elements from data vault modelling. Dimensional modelling is still regarded as the 'gold standard' by industries and data vault modelling is not yet as widely used; however, this may change in future and students that also know about data vault modelling can then benefit from their knowledge. One must, however, caution not to overburden students with additional methodologies to such an extent that they cannot cope with the workload in the course – refer also to Section 4.2.

It is noteworthy that another (more significant) weakness was also identified: the main problem with failed DW/BI is not due to technical infeasibility; rather, these systems fail as a result of limited business benefits being realised and low user acceptance when end-users feel that their requirements have not been met [2, 5]. Implemented systems "may be technically appropriate but culturally/organisationally infeasible or fail to meet user needs" [5] and "in many of the information systems failures that have occurred, the conclusion has been placed squarely on human and organisational factors rather than technical ones" [2]. So, business benefits fail to realise and gathering of business requirements seem to be ineffective, as it apparently fails to effectively capture these human, cultural and organisational dimensions of DW/BI systems.

Business requirements do not reflect organisational improvement; collection approaches typically simply capture requirements such as mere automation of exiting business processes/functionality [9, 17]. Business users tend to restrict themselves within the performance limitations of current systems and hence also only current information; they request only automation of current information when questioned regarding potential actions for improvement [9]. In the Kimball approach, requirements are also gathered mostly around existing business processes and data sources [22]. The output of the requirements gathering process is a "cross-organisational understanding of the present state of data…" in the form of an "enterprise data warehouse business matrix" that summarises the business processes (in the rows) and associated data sources (in the columns) that are then captured as dimension and fact tables [22] – refer to Section 4.3 for a discussion on table structures in dimensional modelling. **Figure 6** below illustrates a two-dimensional business matrix.

	Date	Policy holder	Coverage	Policy	Claim	Payee
Underwriting transactions	X	X	X		X	
Policy premium billing	X	X	X	X		
Agent commissions	X		X	X		
Claims transactions	X			X	X	X

Figure 6: Example of a business matrix

End-users of DW/BI systems "operate in a mode of discovery…cannot tell what the information needs are until they see the possibilities" [17]. Similarly, evolutionary and agile development methods where requirements are continuously gathered throughout the design and development process seem to deliver more successful software artefacts [14, 35, 36]. However, DW/BI systems development projects are large and complex interventions [46]; continuous incorporation of changing requirements during the design and development process may result in poorly structured artefacts that are difficult to maintain [35, 36]. Upon reflection on the relatively higher rates of success of evolutionary and agile development approaches [14, 35, 36], where requirements are gathered by iteratively presenting small (evolving) portions or prototypes to end-users for review/comment, the question arises whether users are able to accurately portray what they *want* from a software system without critical reflection on what it is that they actually *need* from a software artefact, i.e. to *improve,* even beyond what is imaginable, rather than merely *automate* what is easily observable. It could be that, when end-users are exposed to (evolving) portions of a system, it enables them to critically reflect, to imagine using it and benefiting from it in ways that improve their organisational context. If this process of critical reflection can be simulated so that end-users can imagine using a system, and benefiting from it, prior to the design and development phases thereof, DW/BI practitioners can be emancipated to design and develop efficacious DW/BI systems that realise business benefits for end-users, i.e. systems that end-users accept and use.

The fundamental shortcoming that must therefore be addressed when teaching students to become successful DW/BI practitioners that can develop successful (acceptable) DW/BI systems that realise business benefits for end-users, is to teach them how to effectively gather business requirements that constitute organisational improvement, rather than mere automation, i.e. business requirements that also incorporate the human, cultural and organisational dimensions of the organisation into the DW/BI system. To find such an alternative approach may, however, be challenging. Alternative approaches that claim to incorporate the human, cultural and organisational dimensions of IT systems are, for example, ethnographic techniques, as suggested by Viller and Sommerville [44]; an approach named "coherence" that was also suggested Viller and Sommerville [43] and soft systems methodology from Checkland [3]. The effectiveness of these alternative approaches must, however, still be determined; use thereof is not widely documented, especially in the field of DW/BI.

5 Future Research

This study entailed a conceptual reflection on an elective DW/BI course taught to postgraduate students. The assumption is that students that choose to take this course are interested to potentially become DW/BI practioners; or are already working in this field and want to improve their skills so as to advance their careers. This assumption will be tested in a next iteration of this research. Also, the views of current students as well as previous (alumni) DW/BI students that work in the DW/BI field will be determined and incorporated. Alternative requirements gathering approaches that have proved to be more successful are also to be identified. Then, the next step is to test findings empirically: findings is to be incorporated in the DW/BI course and the outcome of these changes is to be tested longitudally to determine success.

6 Summary

This paper conceptually reflects on a postgraduate DW/BI course offered a South African university to determine whether it prepares students sufficiently to become successful DW/BI practitioners. The course is reflected upon according to the four main categories of Ulrich's CSH methodology, i.e. its basis of motivation; its basis of power; its sources of knowledge; and its basis of legitimacy [40].

The course is assumed to be motivated by: students wanting to become successful DW/BI practitioners; industry wanting to employ competent DW/BI practitioners; as well as users wanting usable DW/BI systems. Students are taught DW/BI principles using the Kimball lifecycle approach [22]. Success (or failure) of students are measured through theoretical and practical examinations, as per SAQA [33] requirements. However, success ought to also include reflection on the number of successful systems that these students are able to develop that are accepted by users. Affected users, and gathering of business requirements from these users, are therefore also reflected upon as a source of legitimation.

The three 'schools of thought' in the DW/BI field are discussed as sources of knowledge. The Kimball approach [22] is still regarded as the 'gold standard' in industry and delivers technically good systems. A few shortcomings have been identified – these can be overcome by complementing teaching about dimensional modelling with data vault modelling; however, the additional workload should not extend beyond the limitations of a SAQA National Level 8 course [33]. Also, the main cause identified for DW/BI failure is not technical infeasibility [2, 5].

The most significant reasons for failure are that users feel that DW/BI systems do not realise business benefits and that their business requirements are not met; to add to the complexity of the problem, business users also have difficulty to clearly articulate business requirements that embrace the system's human, cultural and organisational dimensions and, as such, will result in organisational improvement [9]. In short, business requirements do not embody organisations' human, cultural and organisational dimensions and, thus, fail to bring about organisational improvement [2, 5].

Alternative requirements gathering approaches have been identified from literature [3, 43, 44]; however, the effectiveness of these approaches must be determined before they can be incorporated into the course curriculum. It is, however, crucial to identify such an alternative approach as that will enable teaching students to become more efficacious DW/BI practitioners and increase success rates of DW/BI systems. Such an approach is then to be incorporated into the curriculum and tested for effectiveness.

References

1. Alvesson, M. and H. Willmott, *ON THE IDEA OF EMANCIPATION IN MANAGEMENT AND ORGANIZATION STUDIES.* Academy of Management Review, 1992. **17**(3): p. 432-464.
2. Avison, D. and G. Fitzgerald, *Information systems development: methodologies, techniques & tools*. 2006, London: McGraw-Hill.
3. Checkland, P., *Systems thinking, systems practice*. 1981, Chichester: Wiley.
4. Checkland, P. and S. Holwell, *Information, systems, and information systems: making sense of the field*. 1998, Chichester: Wiley.
5. Clegg, B. and D. Shaw, *Using process-oriented holonic (PrOH) modelling to increase understanding of information systems.* Information Systems Journal, 2008. **18**(5): p. 447-477.
6. Delanty, G., *Varieties of critique in sociological theory and their methodological implications for social research.* Irish Journal of Sociology, 2011. **19**(1): p. 68-92.
7. Dresner Advisory Services, L. *Wisdom of the Crowd^{TM} Business Intelligence Market Study $^{(R)}$*. 2012 Accessed: 15 May 2014; Available from: www.yellowfinbi.com/Document.i4?DocumentId=159663.
8. Flood, R.L. and M.C. Jackson, *Creative Problem Solving: Total Systems Intervention*. 1991, Chichester: Wiley.
9. Gardner, S.R., *Building the data warehouse.* Communications of the ACM, 1998. **41**(9): p. 52-60.
10. Gartner. *Business Intelligence Adoption Trends, 2011*. 2011 Accessed: 15 May 2013; Available from: http://my.gartner.com/portal/server.pt?open=512&objID=256&mode=2&PageID=2350940&ref=WorkGroupWidgetContent&resId=1852916.

11. Gartner. *Gartner Says Business Intelligence/Analytics is Top Area for CFO Technology Investment Through 2014*. 2013 Accessed: 6 November 2013; Available from: http://www.gartner.com/newsroom/id/2488616.
12. Goede, R., *Preparing Data Warehousing students to be responsible practitioners: Complementing dimensional modelling teaching with data vault modelling.* Annual International Conference on Computer Science Education: Innovation & Technology, 2016: p. 65-72.
13. Green, R., T. Mazzuchi, and S. Sarkani, *Communication and quality in distributed agile development: An empirical case study.* World Academy of Science, Engineering and Technology, 2010. **61**: p. 322-328.
14. Hina Gull, A., B. Farooque Azam, C. Wasi Haider Butt, and D. Sardar Zafar Iqbal, *A new divide & conquer software process model.* World Academy of Science, Engineering and Technology, 2011. **60**: p. 255-260.
15. Hwang, M.I. and X. Hongjiang, *The Effect of Implementation Factors on Data Warehousing Success: An Exploratory Study.* Journal of Information, Information Technology & Organizations, 2007. **2**: p. 1-14.
16. Inmon, B., *DW 2.0.* DM Review, 2006. **16**(4): p. 8-25.
17. Inmon, B., *World-Class Business Intelligence.* DM Review, 2005. **15**(2): p. 60-61.
18. Inmon, W.H., *Building the data warehouse*. 1996, New York: Wiley.
19. Inmon, W.H., C. Imhoff, and R. Sousa, *Corporate information factory*. 2001, New York: Wiley
20. Jackson, M.C., *Critical systems thinking and practice.* European Journal of Operational Research, 2001. **128**(2): p. 233-244.
21. Keating, C.B., J.J. Padilla, and K. Adams, *System of systems engineering requirements: Challenges and guidelines.* EMJ - Engineering Management Journal, 2008. **20**(4): p. 24-31.
22. Kimball, R., *The data warehouse lifecycle toolkit*. 2008, Indianapolis: Wiley.
23. Kimball, R. and M. Ross, *The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence*. 2010, Indianapolis: Wiley. 744.
24. Leffingwell, D., *Understanding User Needs.* Software Development, 1997. **5**(8): p. 51-55.
25. Linstedt, D., *Method and system of data warehousing and building business intelligence using a data storage model*. 2002, Google Patents.
26. Linstedt, D., *Super Charge Your Data Warehouse: Invaluable Data Modeling Rules to Implement Your Data Vault*. 2011: Createspace.
27. Mahalepa, T., *Developing guidelines for business intelligence modules in information technology programmes at universities using critical systems heuristics*. 2016, North-West University.
28. Maté, A., J. Trujillo, and X. Franch, *Adding semantic modules to improve goal-oriented analysis of data warehouses using I-star.* The Journal of Systems & Software, 2014. **88**: p. 102-111.
29. Mingers, J., *Towards an appropriate social theory for applied systems thinking: critical theory and soft systems methodology.* Journal of Applied Systems Analysis, 1980. **7**: p. 41-50.
30. Naur, P. and B. Randell, eds. *Software Engineering: Report on a Conference, Garmisch, Germany*. 1969, Scientific Affairs Division, NATO: Brussels.
31. Newman, W.M. and M.G. Lamming, *Interactive system design*. 1995, Essex: Addison-Wesley.
32. Popovič, A., J. Jaklič, R. Hackney, and P.S. Coelho, *Towards business intelligence systems success: Effects of maturity and culture on analytical decision making.* Decision Support Systems, 2012. **54**(1): p. 729-739.
33. SAQA *The South African Qualifications Authority: Level Descriptors for the South African National Qualifications Framework*. 2012.
34. Sawyer, P., I. Sommerville, and S. Viller, *Requirements process improvement through the phased introduction of good practice.* Software Process: Improvement & Practice, 1997. **3**(1): p. 19-34.
35. Sommerville, I., *Software engineering*. 9th ed. 2011, New York: Addison-Wesley.
36. Sommerville, I., *Software process models.* ACM Computing Surveys, 1996. **28**(1): p. 263-271.
37. Trauth, E.M., D.W. Farwell, and D. Lee, *The IS Expectation Gap: Industry Expectations versus Academic Preparation.* MIS Quarterly, 1993. **17**(3): p. 293-307.
38. Ulrich, W., *Beyond methodology choice: Critical systems thinking as critically systemic discourse.* Journal of the Operational Research Society, 2003. **54**(4): p. 325-342.
39. Ulrich, W. *A Brief Introduction to Critical Systems Heuristics (CSH)*. 2005 Accessed: 24 February 2014; Available from: http://wulrich.com/downloads/ulrich_2005f.pdf.
40. Ulrich, W., *Critical Heuristics of Social Planning*. 1983, Bern: Haupt.
41. Ulrich, W., *Critical systems thinking*, in *Encyclopedia of operations research and management science*, S.I. Gass and M.C. Fu, Editors. 2013, New York: Springer. p. 314-326.
42. Venter, C. and R. Goede, *The use of critical systems heuristics to surface and reconcile users' conflicting visions for a business intelligence system.* Systemic Practice and Action Research, 2016.
43. Viller, S. and I. Sommerville, *Coherence: An approach to representing ethnographic analyses in systems design.* Human-Computer Interaction, 1999. **14**(1): p. 9-41.
44. Viller, S. and I. Sommerville, *Ethnographically informed analysis for software engineers.* International Journal of Human Computer Studies, 2000. **53**(1): p. 169-196.

45. Wixom, B., T. Ariyachandra, D. Douglas, M. Goul, and B. Gupta, *The Current State of Business Intelligence in Academia: The Arrival of Big Data.* Communications of the Association for Information Systems, 2014. **34**: p. 1-13.
46. Yeoh, W. and A. Koronios, *Critical success factors for business intelligence systems.* Journal of Computer Information Systems, 2010. **50**(3): p. 23-32.

Students' Over-Estimation of Their Academic Ability: A Case Study in Undergraduate Computer Science

Vreda Pieterse, Hein S. Venter and Stefan Gruner

Department of Computer Science, University of Pretoria, South Africa
{vpieterse | hventer | sgruner}@cs.up.ac.za

Abstract. In this paper we show how a large cohort of second-year university students in Operating Systems repeatedly over-estimated their own abilities in tests of predictive self assessment. We conjecture that this persistent discrepancy between students' reality and their optimistically exaggerated self-image is symptomatic for the mass-participation in university education nowadays, and contributes considerably to the grievances notoriously experienced by many tertiary-level students and lecturers of Computer Science in these days. Awareness of this cognitive discrepancy typical of the upcoming generation might help us to mitigate at least a few of the academic problems from which difficult courses with persistently high failure rates, such as Operating Systems, are suffering in the mass-participation universities of nowadays.

Keywords: Computer science education, Academic performance, Struggling students, Self estimation, Improvement of preparedness.

"Know Yourself!"
– Oracle of Delphi

1 Introduction

'Success' of an academic institution as a whole is often measured on the basis of its students' pass rates, whereby it has often been argued that the standards of teaching and contents within a given curriculum play a vital role in those pass rates. For the purpose of this paper, however, we assume that the teaching and curriculum standards in our context are at least in principle acceptable,[1] although the specific course (Operating Systems) which we investigated had a long track record of low pass rates since many years. This perceived difficulty of our course and its persistently high failure rate have motivated us to conduct our investigation.

The case study described and discussed in this paper is based on the presumption that above-mentioned difficulties are somehow 'connected' with a far-ranging 'transformation' of the whole society, including its entire academic system, since the late 19th century (until today). The 'classical' university of the 19th century catered for a very small proportion, namely the *Bildungsbürgertum*,[2] of a country's population. This old situation is sketched qualitatively (without exact numbers) by the purple area in **Figure 1**; for quantitative details see [9][32]. Since the second half of the 20th century, and especially since the 1970s (after the wide-spread social movements of 1968), the number of study beginners at universities world-wide has been rising rapidly for a combination of both *political* and *economic* reasons:

- In the political sphere, social-democratic and left-wing welfare state ideas have been strongly on the rise across the globe, including their promises of equal opportunities for everybody, which fueled *new ambitions* amongst many capable and intelligent young members from the traditional working- and farming-classes.
- In the sphere of economics, advanced means and methods of production by *automation* have made many *traditional crafts and jobs almost obsolete*, in particular those ones for which low levels of education had been for centuries sufficient (e.g.: farm-hands, fishermen, wood choppers in the forests, and the like). To people from those kind of social backgrounds the massively automated economy can hardly offer any work opportunities any more, which threatens those people's livelihoods and forces them to seek new opportunities (through higher education) towards 'white-collar' jobs in indoor offices.

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

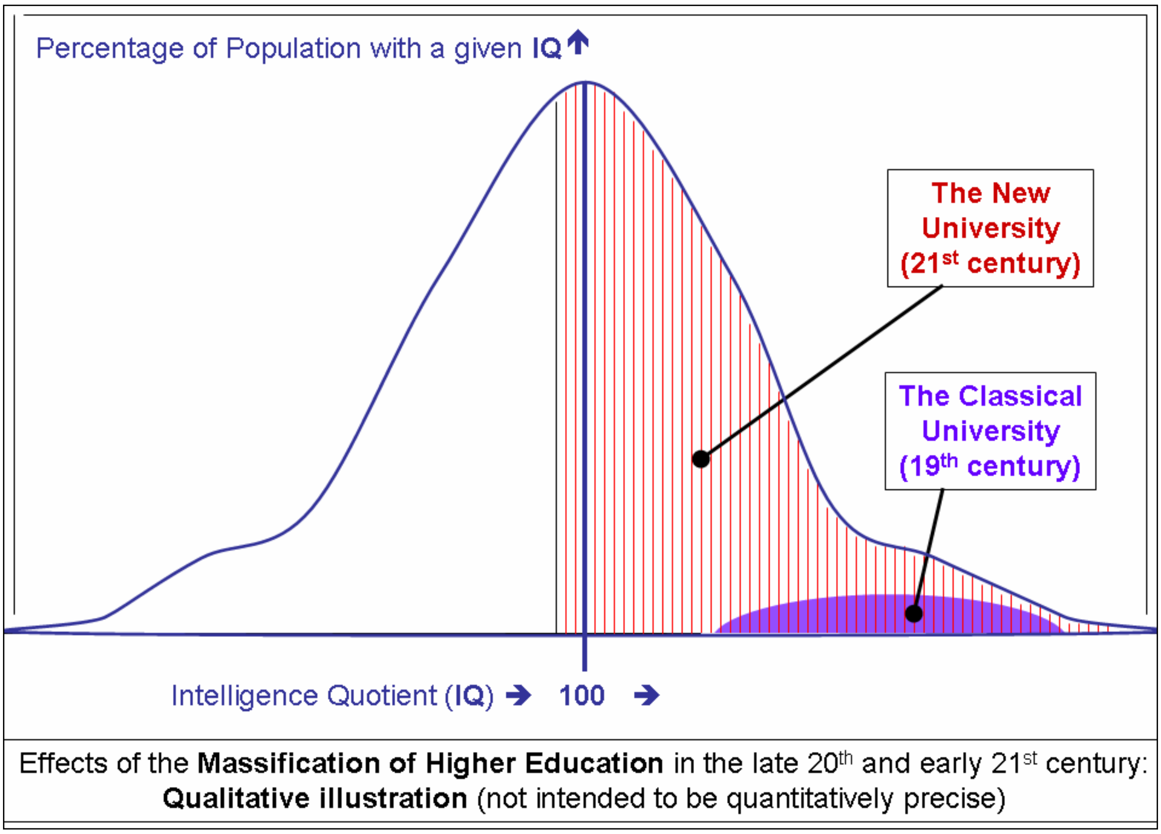


Figure 1: The Law of the Gauβ'ian Bell as Source of Problems in Higher Education.

Such 'rising of the masses' was already diagnosed around the year 1930 by the visionary Spanish philosopher and political intellectual José Ortega y Gasset [21][26]. However, due to the notorious Gauβ'ian bell curve distribution of natural intelligence (see **Figure 1**) across the entire population of every nation [19], the only 'place' from where large masses of students can enter the 'new university' of the 21st century is the 'bulk' of the bell curve where the population's IQ levels are near-average, considerably below the level of exceptional scholarly excellence. This long-term development has some inevitable consequences:

- *If* the 'intellectual level' of tertiary education shall nowadays remain as high as it had been in the classical university of the 19th and early 20th century (purple spot in figure **Figure 1**), then the failure rates among students 'located' near the 'bulk' of the 'bell' must inevitably be high.
- *If*, on the other hand, high pass rates are politically and economically desired for the well-being of the contemporary welfare state, then the 'intellectual level' of nowadays tertiary education cannot remain as high and difficult any more as it had been in the 'classical' university.

In other words: the currently fashionable 'parlance' among many contemporary university managers (provosts, deans and rectors) about keeping up a high standard and at the same time having a high pass rate expresses politically (21st century) and academically (19th century) *conflicting* 'wishful thinking', where at best some semi-satisfying compromise somewhere on the middle ground between those two mutually exclusive extremes can be found; the 'law of the bell' is strict and does not leave much room for miracles. All this is particularly true for our inherently difficult discipline of Computer Science as well, with which large numbers of enrolled students can hardly cope, and for the *scientific-ness* of which the large majority of merely 'skill'-seeking students does no longer have any appreciation [17]. Even worse in this context, as recently confirmed by Suleman, is that large numbers of Computer Science students nowadays *"do not read and do not wish to read"* [29](p.92) any more.

Though it would be good to preserve some attributes of the 'classical' university, the reality of nowadays requires us to think innovatively, to adapt ourselves to our new environment, and to support as many as possible of the new generation of students for the sake of our contemporary society as a whole. The small case study, which we present in this paper, is an attempt at gaining some deeper understanding of this new reality. We want to characterise some of its interesting 'features' in order to be able to find (if possible) some reasonable 'middle ground' on which more students can improve their personal qualifications without the need for universities to substantially lower their institutional academic quality standards.

For the purpose of our investigation the context of it was carefully chosen. As mentioned above, since many generations of students our Operating Systems course (within which we have conducted our case study) has been dreaded and feared by many students as 'very difficult' (no matter by which individual lecturer with which particular textbook it had been taught over the years); see [25] for comparison. Word-of-mouth on campus, as well as student-written 'blogs' on the Internet, spread out the usual anecdotes, and since many years the course has appeared in the university's annual 'rank list' of those courses with uncomfortably high failure rates (alongside, for example: Inorganic Chemistry). Some particularly unfortunate students in Operating Systems needed several years of consecutive re-attempts before they eventually passed with mediocre marks. In many of those cases the failing students themselves were painfully 'surprised' by their failures, having (wrongly) believed for too long time that their own success would be almost guaranteed – according to a wide-spread belief that failing students are almost always "the other students", not "me". Our Operating Systems course, with its enrollment number of more than 350 students, was also large enough to 'lift' the significance of our findings at least a little bit above the level of mere 'anecdotal evidence'.

For lecturers and tutors of Computer Science at other universities the findings of our investigation should be reasonably interesting, because the persistent discrepancy between many students' exaggerated subjective self-image and their rather harsh objective reality in courses as difficult as Operating Systems [25] is a permanent source of grievance in contemporary academic (Computer Science) environments. The *origins* (or 'root causes') of those exaggerated self-images found among our students, however, are *not* included in the research scope of this paper any more: they are left to be investigated in 'future work'.

2 Related Work

According to Ghazvini, 'self-concept' is a multifaceted construct consisting of knowledge and attributes that people use to describe themselves [15]. Academic self-concept, self-esteem and self-efficacy are some of the prominent aspects of self-concept. Self-esteem is connected with social competence whilst self-efficacy is the belief in one's ability to carry out a given task or assignment successfully. It is generally accepted that self-efficacy improves students' achievements [4], which in turn bolsters both self-esteem and academic self-concept. Successful education is believed to be related to how effective an educator is in strengthening the self-efficacy of the learners [1][2].

Numerous studies have been conducted about people's insight into their own abilities [38], as well as about students' perception of self and what it does to their cognitive and psychological well-being [38]. In [6] it was found that the correlation between self-perceptions and performance was significantly higher among high-ability children than among low-ability children, which suggests that accurate academic self-concept is indicative of higher ability. Similar observations were described in [22][23], according to which bottom-quartile participants were nearly four times more 'mis-calibrated' than their top-quartile counterparts, and according to which top-quartile participants tended to under-estimate their performance; see also [12][20] for comparison. This phenomenon was *aphoristically* expressed already by Bertrand Russell more than 60 years ago: *"One of the painful things about our time is that those who feel certainty are stupid, and those with imagination and understanding are filled with doubt and indecision"* [27].

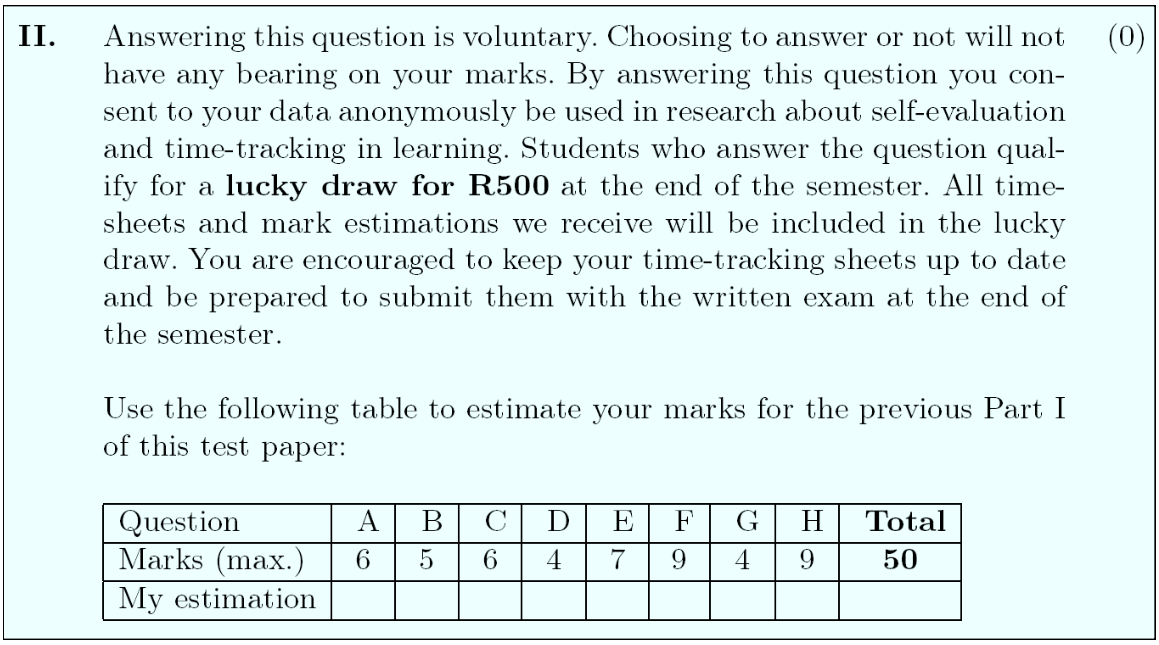
Ghazvini also asserted that accurate academic self-concept predicts ability and performance in literature and mathematics [15]. Moreover it has been claimed that a healthy self-concept is valuable in any profession [28]. While it is essential for academic growth [5][31], there is on the other hand some danger in having an inflated or otherwise unrealistic perception of one's own ability, e.g.: pursuing careers for which one is objectively under-qualified [38]. In [10] it was postulated that, on average, people tend to believe that they themselves would be *above average* in skill; they are thus likely to reach judgments with too much confidence – a consequence that defies the laws of descriptive statistics. The authors of [10] point to the 'double curse' from which incompetent individuals have to suffer: their deficits cause them to make errors and also prevent them from gaining insight into their errors. The authors of that paper have expressed their hope that 'up-skilling' might help those unfortunate individuals to recognize the limitations of their abilities.

The inclination of people towards a heightened sense of egotism associated with over-estimation of their own abilities is believed to be more prevalent in the upcoming generations of nowadays than in previous generations of the not-so-distant past [35][37]. In related research by Twenge and Campbell [34][36] on survey responses provided by 37'000 college students, about ≈30% (per cent) of those students are reported to believe that they 'ought to' obtain good marks simply for paying their tuition fees – thereby not understanding (or not wanting to understand) that with tuition fees 'invested' into an 'education market' one can only 'buy' a learning-opportunity, *not* an actual learning-result. In [33], on the other hand, it was critically pointed out that case studies which generally confirm a universal rise of narcissism among the youth may not be statistically (or otherwise methodologically) sound.[3]

3 Investigation: Motivation and Design

Wise planning and dedication shown by a university student is often a catalyst for personal success [3]. However, how do we measure whether a student's amount of planning and dedication is sufficient for success? In order to guide our students to appreciate the value of effective time management we instructed them to read an article about time management on a website called 'Good Luck Exams' by Rebecca Ezekiel.[4] We also encouraged our students to keep track of the time they spend (individually) on our course, and provided them with pre-designed personal time-tracking sheets (including items like 'background reading time', 'practical programming time', 'exam preparation time', and so on) for this purpose. We pointed out that, when they do this time-tracking exercise while being honest to themselves (i.e.: no cheating with hours and half-hours), the time-tracking sheet would help them to 'balance' their 'time budgets' more efficiently towards success. Our students were not compelled to complete and submit their time-tracking sheets. In order to 'encourage' voluntary participation, a 'lucky draw' was announced in which the (voluntary) submitters of those time-tracking sheets could possibly win a reasonably sized 'prize' at the end of the semester. However, in spite of the 'enticing' with a 'prize', our students' participation in the voluntary time-tracking exercise was insignificantly low and could, for this reason, not be evaluated. The remainder of this paper deals thus only with our students' self-perception in contrast to their actual academic 'performance'.

During the course our students had to write some semester tests. Each of the test papers consisted of two parts: one compulsory, one voluntary. The first (compulsory) part consisted of the usual questions and calculation tasks in the field of Operating Systems, whereas the second (voluntary) part asked for the students' contribution to our case study research. In that part, as shown in **Figure 2**, the students were asked to *predict* (or guess) their own results to be achieved in the various questions of the first (compulsory) part.



II. Answering this question is voluntary. Choosing to answer or not will not have any bearing on your marks. By answering this question you consent to your data anonymously be used in research about self-evaluation and time-tracking in learning. Students who answer the question qualify for a **lucky draw for R500** at the end of the semester. All time-sheets and mark estimations we receive will be included in the lucky draw. You are encouraged to keep your time-tracking sheets up to date and be prepared to submit them with the written exam at the end of the semester. (0)

Use the following table to estimate your marks for the previous Part I of this test paper:

Question	A	B	C	D	E	F	G	H	**Total**
Marks (max.)	6	5	6	4	7	9	4	9	**50**
My estimation									

Figure 2: Assessment of Students' Self-Estimation.

The numbers of students, who participated in our case study, are listed in **Table 1**.

[3] Elderly complaints about a perceived "decline of virtues" amongst "the youth" have been recorded in writing since the days of classical antiquity.

[4] http://www.goodluckexams.com/8-steps-to-effective-time-management-for-students/

Table 1: Response Rates.

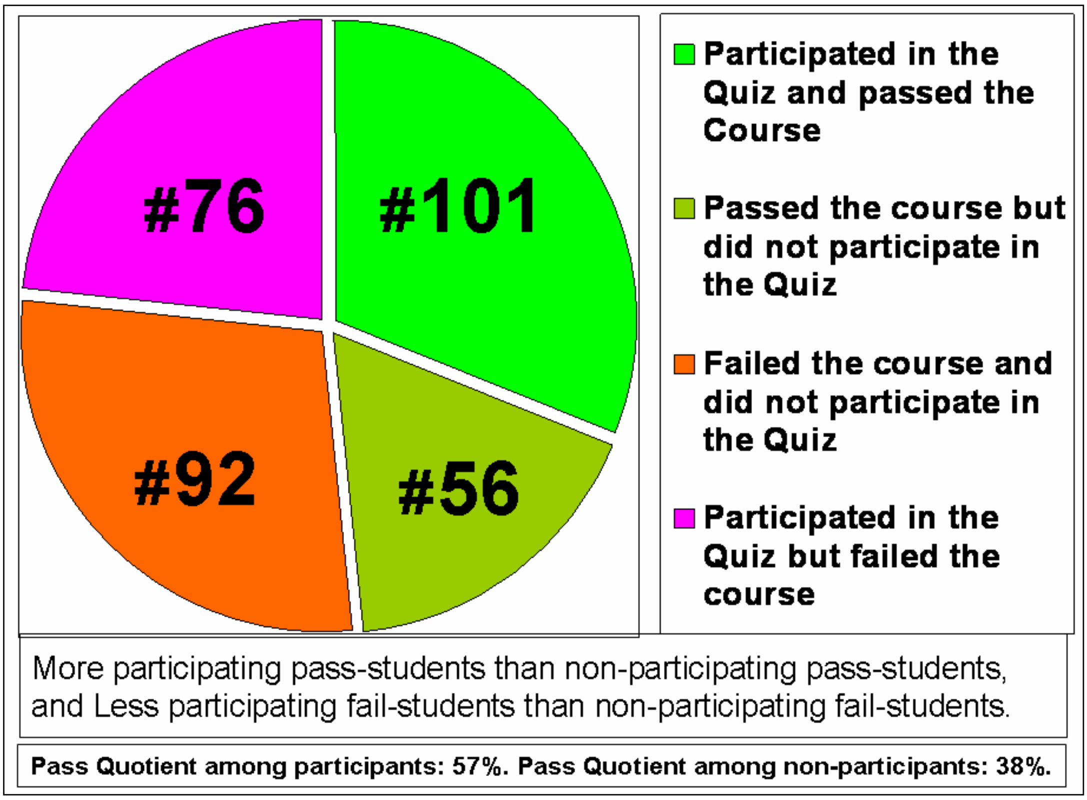
Students	Test #1	Test #2
# Students registered for the Operating Systems course	370	356
# Students who wrote the semester test	350	336
# Students who voluntarily answered our research question (Part II)	146	129

4 Observations: Data and Evaluation

Many of our students turned out to be fairly indifferent towards our intentions and purposes. Some students, however, might have been influenced by our initiative. The following is a comment written by a (weak) student who did *not* participate in the estimation exercise (Part II) of the first semester test (Test #1): *"I have no confidence in any of the answers provided above. Therefore, it would be hard for me to provide an estimation without making myself feel bad. Please do include this question in the following tests as I will be more confident"*. The student realized his shortage of understanding after writing the test, but was not yet prepared to 'quantify' it. This insight seemed to motivate him initially to be better prepared for the second semester test. However, this same student did then also not participate in the voluntary estimation exercise in the second semester test.

Another student, who did not attempt a detailed estimation per question, provided a rough 'overall' estimation for the test as a whole and voiced a similar self-reflection after the first semester test: *"After writing this test I am disappointed in myself because I should have spent more time studying, but my estimation is that the mark I will get will be around 45%–50%"*. Also he was one of those students who grossly over-estimated their own actual results.

All in all, among the 325 students who wrote the final exam after the semester (whereby 55 of the students who were registered at the time of the first semester test had dropped out before the final exam) there were 177 participants, yielding a participation rate of ≈54%: see **Figure 3**. The same figure also shows that the final pass rate (after the exam) was notably higher among the interested participants (101 : (101+76)) than among the non-participants (56 : (56+92)). In other words: the on-average 'stronger' students were the ones who were on-average also more 'open-mined' towards voluntarily participating in the (potentially 'painful') self-estimation exercise. *Not* all participants participated in *both* self-estimation exercises: some participated only in the first one, some participated only in the second one.



More participating pass-students than non-participating pass-students, and Less participating fail-students than non-participating fail-students.

Pass Quotient among participants: 57%. Pass Quotient among non-participants: 38%.

Figure 3: Proportions of Students who were reached by our Investigation.

The self-estimation results from the first part of our investigation are displayed in **Figure 4**. The diagonal line in the figure separates the two sub-spaces of self-over-estimation from self-under-estimation. As it is shown by the black dots in the diagram, the large majority of participants strongly over-predicted their own performance (regardless of whether those participants actually failed or passed the Operating Systems test the outcome of which they had to predict). Only few participants predicted the result of the test more or less accurately, or even under-estimated their actual achievement. As shown by the pale-blue triangles in **Figure 4**, about *half* of those (few) students, who *under*-estimated their own ability, actually passed the written Operating Systems test. With a pass-score of 25 points for this specific test, the reader can easily 'extract' from **Figure 4** the following four categories of participants in four 'quadrants' of the dot-space:

1. self-over-estimated and failed the test,
2. self-over-estimated but passed the test,
3. self-under-estimated but failed the test nevertheless,
4. self-under-estimated and passed the test.

All in all, **Figure 4** shows clearly that most participants could be found in the first of the four categories.

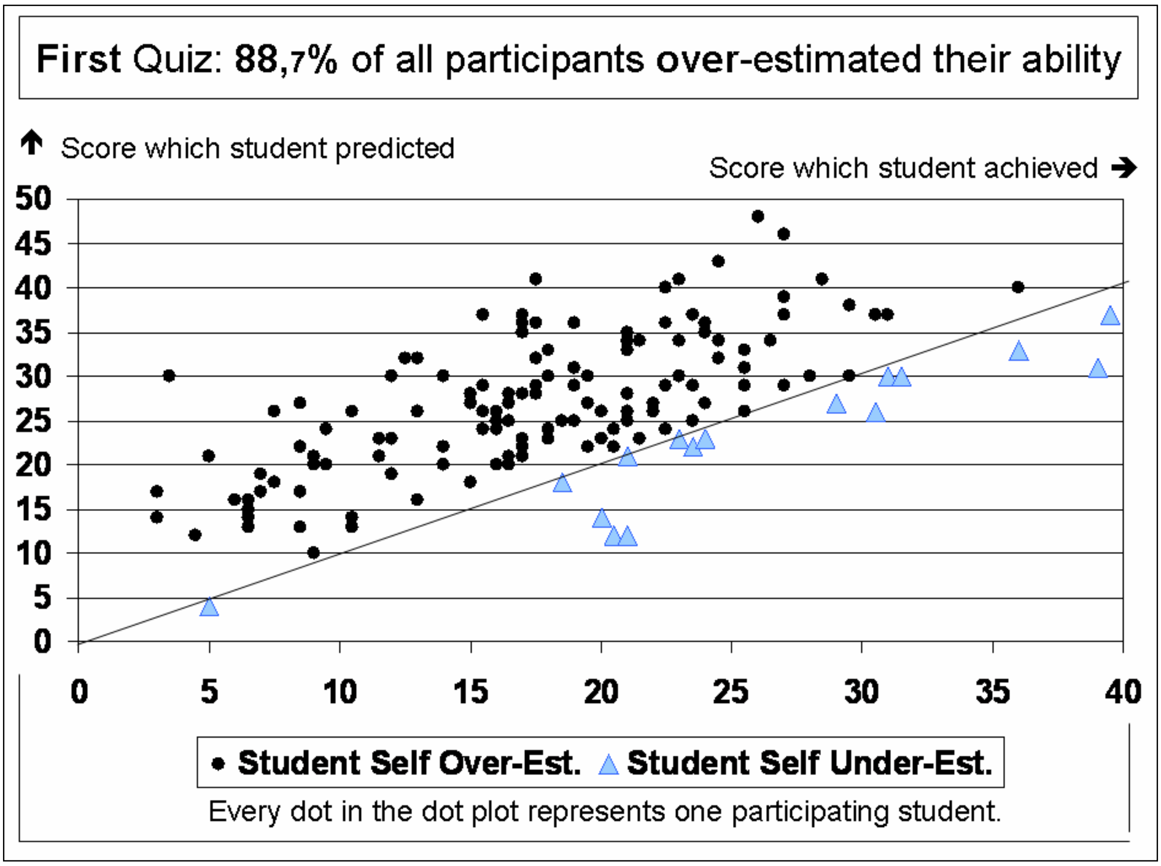


Figure 4: With this test's pass score of 25 points, most participants fell into the category 'over-estimate and fail'.

In the second self-estimation exercise we had fewer participants than in the first one, as it can be seen from the `sparser' distribution of dots in **Figure 5**. In spite of this being the second test (i.e.: at least the twice-participating students should have been able to remember their own prediction accuracy, or shortage thereof, from the previous self-estimation exercise), the image displayed by figure **Figure 5** is qualitatively similar to the previous one: also in this second test, most participants came (again) into the 'over-estimate and fail' category.[5]

[5] During the semester it was thanks to the programming practicals, rather than those written tests, that many students of our course eventually obtained the entrance permit for the final exam after the semester.

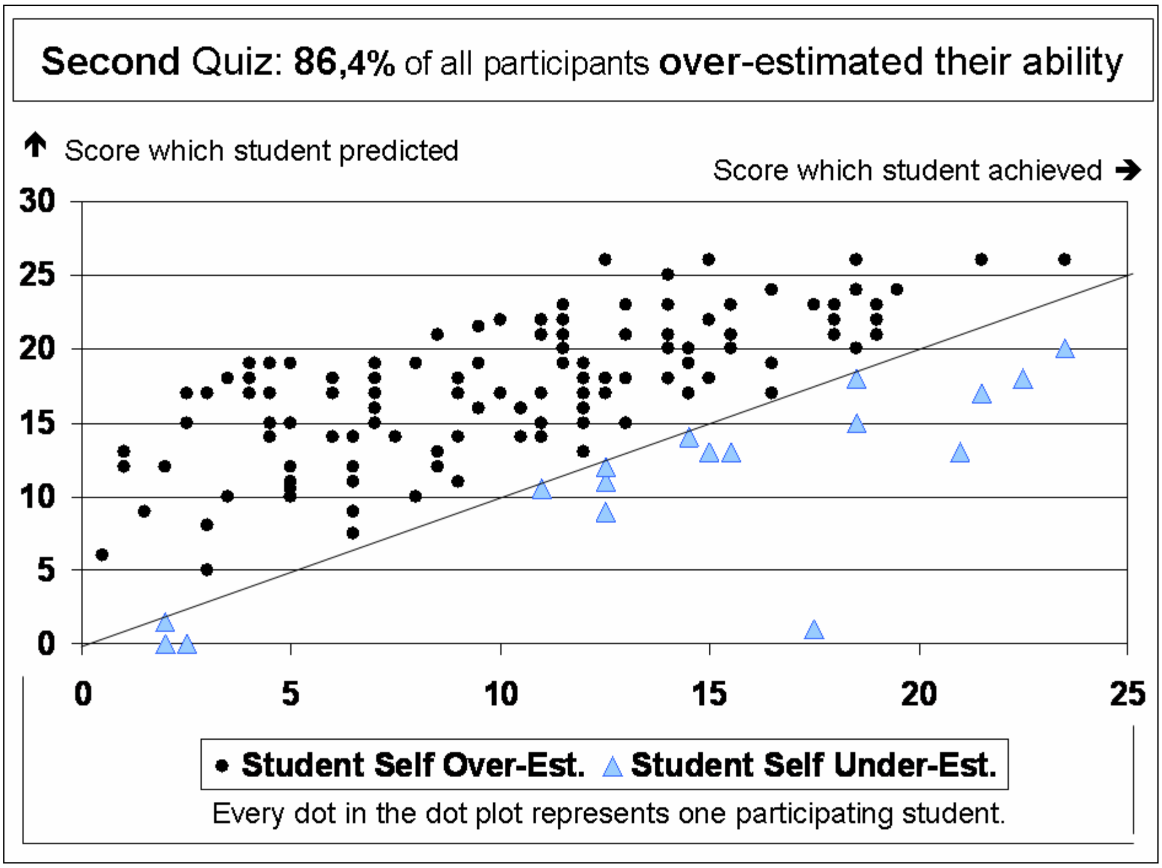


Figure 5: With this second test's pass score of 15 points, most participants fell again into the category 'over-estimate and fail'.

The *combined findings* of the afore-mentioned two self-estimation exercises are depicted in **Figure 6**. It captures the results from only those (fewer) students who had participated in *both* of those self-assessment exercises. The figure shows clearly that the proportion of those participants, who under-estimated themselves twice, is only small. Some participants sometimes over- and sometimes under-estimated themselves (as it could also have happened in a random number game), but the overwhelming majority of participants over-estimated their capability in *both* subsequent exercises (possibly *hoping* that "*next time* everything will be much better", as weak students often say to console themselves).

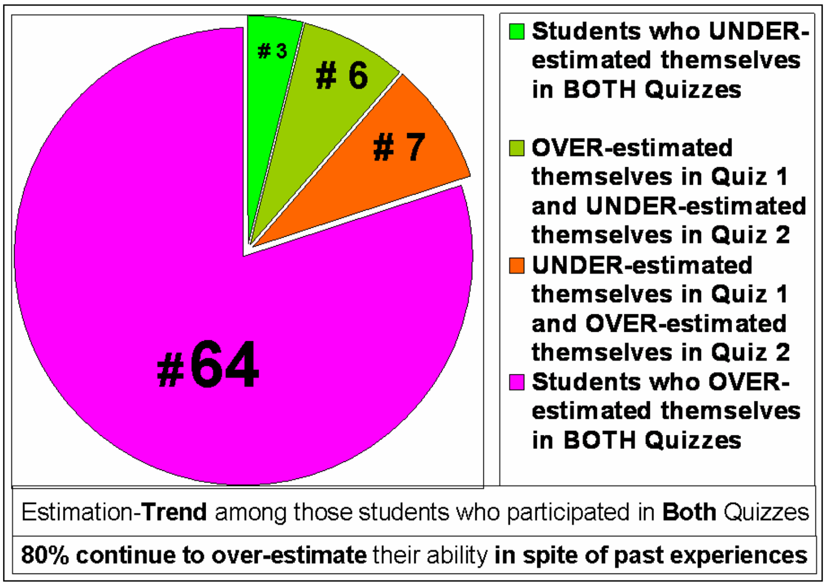


Figure 6: Proportion of twice-participants who over-estimated their ability in both self-estimation exercises.

5 Discussion and Recommendations

Our (admittedly limited) case study highlights how most of our rather weak Computer Science students *continuously over-estimated their own academic capabilities* (even in spite of previous failure experiences) in an attitude that we may call 'stubborn optimism'. Such attitude seems to be a wonderful and life-serving privilege of the youth, though the phenomenon also occurs among older and higher-educated people; (for example: 94% of the academics surveyed by [8] believed to be 'above average' in their professional qualities).

Our findings support those of [13] whereby in 68% of cases the surveyed students gave themselves higher marks than they would have obtained from their educators. Our observations are also 'in line' with [18] whereby better students were able to make more accurate predictions as their semester went on, whilst poorly performing students remained over-confident all the time in spite of the feedback they had received on earlier exams. These and similar phenomena have become notorious under the label of the *Dunning-Kruger effect*,[6] in which mediocre individuals suffer from illusory superiority, mistakenly assessing their ability as much higher than it really is, whereas highly competent individuals are more inclined to underestimate their relative competence (and may erroneously assume that tasks, which are easy for them, would also be easy for others). In that context Dunning and Kruger postulated that the 'mis-calibration' of the incompetent would stem from an error about the self, whereas the 'mis-calibration' of the highly competent would stem from an error about others; recall for comparison Bertrand Russell's well-known aphorism.[7]

In spite of the 'alignment' of our findings with the above-mentioned related work, we must point out that we have made our observations in only one course, in only one specific university, in only one specific academic year, in only one specific sub-discipline of the ACM Computer Science curriculum. Hence we cannot 'universalise' our results. Nevertheless it seems reasonably justifiable to provide, at the end of this paper, at least the following few *tentative* 'recommendations' as 'lessons learned' for our future practice:

5.1 Recommendations to University Students

According to our findings, (too) many students are naively unaware of their inability to cope with the academic expectations in Computer Science. Self-awareness, however, can be improved by 'training' like any other 'skill' can be trained. Hence we suggest that students should consciously work on their self-knowledge by regular (perhaps weekly) 'reflections' about their own level of preparedness, with the goal of being able to identify feasible actions by means of which their preparedness could be enhanced. Self-awareness could be increased by any interested student, for example, along the lines of some of the following simple *self-consultation questions* which are motivated by the classical Delphic maxim [24] of "know yourself":

- Did I take a similar course already before university, in secondary school?
- Do I anticipate to fare similarly well for the university courses I need to take, and, if I believe so, then what is my basis of such belief? Is my belief based on real experiences?
- How does the curriculum at university differ from the similar course(s) I took during secondary school, and what might be the consequences of such differences?
- How do the methods of education at university differ from the methods of education at secondary school?
- Am I still 'learning' like an externally 'instructed' school pupil, or am I already 'studying' autonomously like a university student?
- How must my individual and social habits and status change due to my transition from secondary school to university?
- Whom could I possibly ask for advice if I am 'first-generation academic' and my family is not comfortably 'embedded' into the already well-educated social class of the *Bildungsbürgertum*?

5.2 Recommendations to University Lecturers

The above-mentioned preparedness problem, including the observed discrepancy between the students' self-image and their external reality, may be due to the challenges that students face when making their transition from secondary school to university. Such challenges include (but are not limited to) the following points:

[6] https://en.wikipedia.org/wiki/Dunning-Kruger_effect

[7] *"The fundamental cause of the trouble is that in the modern world the stupid are cocksure while the intelligent are full of doubt"* , http://en.wikiquote.org/wiki/Bertrand_Russell

- Students suddenly find themselves in a different, more mature and less restricted environment in which they have more freedom to do what they want (as opposed to being in secondary school). In this new environment, students often do not know how to 'handle' their new freedom and are thus easily distracted by the many other pastime possibilities which the modern life-style is offering to them. Particularly to some students who come to university out of remote, secluded, 'close-knit', isolated sectors of society, the sudden exposure to so many different opinions and so many different 'walks of life' at university may even amount to a veritable 'culture shock'.
- Students typically 'learn' new concepts from a starting position of narrow and restricted viewpoints. Thus believing to have already understand the newly learned concept fully and comprehensively, they find out only during the perusal of an already written test or exam paper that they did not actually understand the concept correctly because the wider 'context' of the newly learned concept was not properly taken into account. The lecturers, on the other hand, often forget that their own perception of the taught concepts is already standing on a much broader context basis; hence the lecturers are often surprised by their students' apparent 'narrow-mindedness' – see [14] for comparison.

Hence, lecturers ought to take note more 'proactively' of the magnitude of the problem that students have with the above-mentioned discrepancy between self-perception and external reality. Consequently, lecturers might help their students along the lines of the following recommendations:

- Motivate students by fostering onto them a positive attitude towards successfully completing their university courses and degrees. This can be accomplished with motivational lectures at the beginning of a course, or during the usual 'orientation days' at the beginning of a semester. These lectures should especially refer to the 'generational changes' and their implications for contemporary students to be able to live up to the expectation posed by the university.
- Intervene again and again during a course in order to take remedial actions proactively (rather than reactively), alert students frequently about their risks, and rectify the students' own misconceptions as soon as you become aware of them, even if such 'interventions' might be perceived as a tedious, time-consuming distractions from 'getting through' your course's mandatory syllabus.

5.3 Recommendations to Teachers at Secondary Schools

Secondary schools before university are currently not doing enough to prepare their pupils for the kind of freedom which is going to await them at university. School teachers should already before university begin to treat their older pupils in 'university style', by taking (amongst others) the following points into account:

- The transition from secondary school to university is often 'large', and secondary school pupils are often found un-prepared for it. School teachers should thus liaise with universities more systematically to organise information and preparation seminars for secondary school pupils well in advance. The pupils must receive early warnings from their teachers about the high likelihood of failure at their next higher level of education: 'excellent' school pupils are likely to be merely 'good' at university, and 'good' school pupils are likely to become merely 'copers' or 'strugglers' at university.
- One reason of the transition from secondary school to university often being so 'large' is the notable difference in 'standard' from subjects taught before and at university. In South Africa, in particular, this dilemma is exacerbated by low pass thresholds in secondary schools. For example, we are aware of cases in which mathematics exams at South African secondary schools were considered as 'passed' with merely 30% (per cent) of its tasks being solved. Such a 'standard' instills in pupils already at a young age the damaging misconception that a 30% achievement would be 'good' or, more generally speaking, that mediocrity would be acceptable. With such an attitude, however, failure at university will follow soon.
- School teachers should also train their pupils in 'information retrieval' and should encourage them to use school libraries or public municipal libraries as frequently as possible by their own initiative. This is because the specific problems (insufficient self-awareness) discussed above in our paper are often connected with the problem of too 'narrow' world-views in general, which can be mitigated by 'far-ranging' eclectic leisure-time reading on various topics. Recommendable topics for pupils' leisure-time reading (outside the school's mandatory curriculum) are high-quality youth-books which illustrate (in an entertaining manner) to their young readers the typical problems of youth (e.g.: troubles with school, generational conflicts: young versus old, conflict resolution, and the like), and which at the same time can also improve the pupils' often sub-acceptably low language proficiency in reading and oral articulation.

6 Conclusion and Future Work

Our paper described and discussed the problem of university students who are only insufficiently able to estimate how they would fare in certain university courses, in our case: a 2nd-year course from the 'core' of the Computer Science (and Computer Engineering) curriculum [30]. In good agreement with previously published work (see: 'Dunning-Kruger effect') on this problem we found that especially the weak students tend to overestimate very strongly their ability to do well in the university course we examined.

Although our case study had some 'quantitative' elements it, it yielded by-and-large a 'qualitative impression' result which is still in need of further methodological consolidation and statistical corroboration, also including a solid analysis of significance versus spurious-ness [7](sect. 14.4). To-date our findings might still be regarded as 'anecdotal evidence', because merely one course in one academic year was in the focus of our investigation. Nevertheless we can find a number of methodological hints and guidelines in papers like [11] and [16] along the lines of which it should be possible to strengthen the trustworthiness of our preliminary empirical findings as well as the plausibility of our preliminary theoretical conjectures in the not-too-far future.

Moreover, some kind of weighted scoring 'model' might also be developed by means of which a student's self-assessment ability could be 'measured' (quantitatively) with sufficient reliability. Such measurements could then perhaps also be helpful for the purpose of 'guiding' particularly the weak students through their curricula at university.

Last but not least: suitable `knowledge transfer' within the universities, as well as from the universities `back' into the secondary schools (from where our young students are coming), might be able to mitigate (at least to some extent) this problem of the 'Generation Me' which our paper has addressed.

Acknowledgments

Thanks to the anonymous reviewers for their helpful-critical remarks, and thanks to our students for their voluntary participation in our survey.

REFERENCES

1. Ames, C.: *Classrooms: Goals, Structures, and Student Motivation*. Journal of Educational Psychology 84(3), 261 (1992).
2. Ames, C.: *Motivation: What Teachers Need to Know*. Teachers College Record 91(3) (1990).
3. Atman, C.J., Sheppard, S.D., Turns, J., Adams, R.S., Fleming, L.N., Stevens, R., Streveler, R.A., Smith, K.A., Miller, R.L., Leifer, L.J., Yasuhara, K., Lund, D.: *Enabling Engineering Student Success*. Final Report for the Center for the Advancement of Engineering Education, Tech. Rep. CAEE-TR-10-02 (2010).
4. Bandura, A.: *Self-Efficacy: The Exercise of Control*. Macmillan (1997).
5. Boud, D.: *The Role of Self-Assessment in Student Grading*. Assessment & Evaluation in Higher Education 14(1), 20-30 (1989).
6. Bouffard, T., Markovits, H., Vezeau, C., Boisvert, M., Dumas, C.: *The Relation between Accuracy of Self-Perception and Cognitive Development*. British Journal of Educational Psychology 68(3), 321-330 (1998).
7. Bunge, M.: *Philosophy of Science, Volume Two: From Explanation to Justification*. Revised ed., Transaction Publ., (1998).
8. Cross, K.P.: *Not can, but will College Teaching be improved?* New Directions for Higher Education 17, 1-15 (1977).
9. Diebolt, C.: *L'Evolution de Longue Periode du Systeme Educatif Allemand (XIXe et XXe Sieecles).* Histoire Quantitative de l'Economie Francaise (Serie AF) 23 (1997).
10. Dunning, D., Heath, C., Suls, J.M.: *Flawed Self-Assessment.* Psychological Science in the Public Interest 5(3), 69-106 (2004).
11. Dunning, D., Meyerowitz, J.A., Holzberg A.D.: *Ambiguity and Self-Evaluation: The Role of Idiosyncratic Trait Definitions in Self-Serving Assessments of Ability*. Journal of Personality and Social Psychology 57(6), 1082-1090 (1989).
12. Ehrlinger, J., Johnson, K., Banner, M., Dunning, D., Kruger, J.: *Why the Unskilled are Unaware: Further Explorations of (absent) Self-Insight among the Incompe-tent.* Organizational Behavior and Human Decision Processes 105(1), 98-121 (2008).
13. Falchikov, N., Boud, D.: *Student Self-Assessment in Higher Education: A Meta-Analysis.* Review of Educational Research 59(4), 395-430 (1989).
14. Fleck, L.: *Entstehung und Entwicklung einer wissenschaftlichen Tatsache*. Benno Schwabe & Co. Publ., (1935). English translation available as: *Genesis and Development of a Scientific Fact*. University of Chicago Press, (1979).
15. Ghazvini, S.D.: *Relationships between Academic Self-Concept and Academic Performance in High School Students.* Procedia Social and Behavioral Sciences 15, 1034-1039 (2011).
16. Gramzow, R.H., Elliot, A.J., Asher, E., McGregor, H.A.: *Self-Evaluation Bias and Academic Performance: Some Ways and Some Reasons Why.* Journal of Research in Personality 37(2), 41-61 (2002).

17. Gruner, S.: *On the Future of Computer Science in South Africa: A Survey amongst Students at University.* In: Proceedings SACLA`15, 44th Annual Conference of the Southern African Computer Lecturers' Association, Johannesburg (2015).
18. Hacker, D.J., Bol, L., Horgan, D.D., Rakow, E.A.: *Test Prediction and Performance in a Classroom Context.* Journal of Educational Psychology 92(1), 160-170 (2000).
19. Herrnstein, R., Murray, C.: *The Bell Curve: Intelligence and Class Structure in American Life.* Free Press (1994).
20. Hodges, B., Regehr, G., Martin, D.: *Difficulties in Recognizing One's Own Incompetence: Novice Physicians who are Unskilled and Unaware of it.* Academic Medicine {\bf 76}(10), s78-s89 (2001).
21. Krüger, J.: *Der Begriff des Massenmenschen in der Philosophie Nietzsches, Ortegas, und Heideggers.* Doctoral Dissertation, Faculteit Lettere en Wysbegeerte, Universiteit van Pretoria (1974).
22. Kruger, J., Dunning, D.: *Unskilled and Unaware of it: How Difficulties in Recognizing One's Own Incompetence lead to Inflated Self-Assessments.* Journal of Personality and Social Psychology 77(6), 1121-1134 (1999).
23. Kruger, J., Dunning, D.: *Unskilled and Unaware of it: How Difficulties in Recognizing One's Own Incompetence lead to Inflated Self-Assessments.* Psychology 1, 30-46 (2009).
24. Lamb, W.R.M. (ed.): *Lysis, Symposium, Gorgias.* Plato in Twelve Volumes 3, Harvard University Press (1967).
25. Machanick, P.: *Teaching Operating Systems: Just Enough Abstraction.* In: S. Gruner (ed.), ICT Education, Proceedings SACLA`16, 45th Annual Conference of the Southern African Computer Lecturers' Association, Cullinan. CCIS {\bf 642}, 104-111 (2016).
26. Ortega y Gasset, J.: *The Revolt of the Masses* (1930), 4th impr., Unwin Books (1970).
27. Russell, B.: *New Hopes for a Changing World.* Simon and Schuster (1951).
28. Stefani, L.A.: *Peer, Self and Tutor Assessment: Relative Reliabilities.* Studies in Higher Education 19(1), 69-75 (1994).
29. Suleman, H.: *Flipping a Course on Computer Architecture.* In: S. Gruner (ed.), ICT Education, Proceedings SACLA`16, 45th Annual Conference of the Southern African Computer Lecturers' Association, Cullinan. CCIS 642, 83-94 (2016).
30. Stallings, W.: *Operating Systems: Internals and Design Principles.* 8th ed., Pearson Publ., (2015).
31. Thompson, G., Pilgrim, A., Oliver, K.: *Self-Assessment and Reflective Learning for First-Year University Geography Students: A Simple Guide or Simply Misguided?* Journal of Geography in Higher Education 29(3), 403-420 (2005).
32. Titze, H., Herrlitz, H.-G., Müller-Benedict, V., Nath, A.: *Das Hochschulstudium in Preußen und Deutschland 1820-1944.* Vandenhoeck & Ruprecht (1987).
33. Trzesniewski, K.H., Donnellan, M.B., Robins, R.W.: *Do Today's Young People really Think they are so Extraordinary?* Psychological Science 19(2), 181-188 (2008).
34. Twenge, J.M, Campbell, W.K.: *The Narcissism Epidemic: Living in the Age of Entitlement*. Atria Books (2009).
35. Twenge, J.M., Foster, J.D.: *Mapping the Scale of the Narcissism Epidemic: Increases in Narcissism 2002-2007 within Ethnic Groups*. Journal of Research in Personality 42(6), 1619-1622 (2008).
36. Twenge, J.M., Konrath, S., Foster, J.D., Campbell, W.K., Bushman, B.J.: *Further Evidence of an Increase in Narcissism among College Students*. Journal of Personality {\bf 76}(4), 919-928 (2008).
37. Young, S.M., Du, F., Dworkis, K.K., Olsen, K.J.: *It's all about All of Us: The Rise of Narcissism and its Implications for Management Control System Research*. Journal of Management Accounting Research 28(1), 39-55 (2016).
38. Zell, E., Krizan, Z.: *Do People have Insight into their Abilities? A Metasynthesis*. Perspectives on Psychological Science 9(2), 111-125 (2014).

Authorized Cheat-sheets as an Educational Tool in Computer Science Examinations

Katherine M. Malan[1] and Vreda Pieterse[2]

[1]University of South Africa, malankm@unisa.ac.za
[2]University of Pretoria, vpieterse@cs.up.ac.za

Abstract. Authorized cheat-sheets in formal assessment provide some of the advantages of open-book examinations, such as reduced student anxiety, without the disadvantages, such as the time used by students to look up information during examinations. The use of authorized cheat-sheets has the potential added benefit of students spending more time consolidating the course concepts in order to summarize the most important aspects while preparing their cheat-sheets. This paper describes an investigation into the use of authorized cheat-sheets in the final examination of a second year computer science module on computer architecture and assembly programming. The volume and type of content as well as the quality of student-constructed cheat-sheets were analysed. The impact of these different aspects on student performance was investigated. It was found that the quality of the cheat-sheets created by the students was positively correlated with improvement in individual performance, but that the general performance decreased when cheat sheets were introduced.

Keywords: authorized cheat-sheets, quality content analysis.

1 Introduction

On the spectrum of examination formats from closed-book to take-home examinations, the use of cheat-sheets falls between these two extremes. Authorized cheat-sheets refer to the practice of allowing students the opportunity to compile limited notes in preparation for the formal assessment and to use these notes during the assessment.

One disadvantage of closed-book examinations is that students may spend valuable study time memorizing facts, rather than trying to understand concepts. Complete open-book examinations, on the other hand, have the disadvantage that students may spend less time studying the material in preparation for the examination [2]. Students may prepare less, knowing that they will be able to look up information during the examination. This is usually a false sense of security, however, because if a student does not have a proper understanding of the concepts beforehand, they usually do not benefit from having the information on hand during the assessment.

An added advantage of cheat-sheets over closed-book and open-book formats is that it offers an opportunity for the students to consolidate their thinking in order to create the cheat-sheet. This can help to structure study time since it involves a concrete task that must be completed before the assessment. When using authorized cheat-sheets a limitation on the quantity and format of allowable notes during the examination is specified to level the playing field during the examination in the sense of allowing students equal opportunity to choose the content they want to access. Original handwritten format was chosen to not advantage students with wider access to resources than others and to discourage copying.

Other than access to course materials, aspects to be considered in the design of tests and examinations also include the time allowed, level of permitted assistance / collaboration, and scope. The form of assessment investigated in this study was time-restricted (3 hour examination), without any assistance or collaboration and of clearly specified scope. In a series of formative tests a closed-book format was used, while in the examination, authorized cheat-sheets were allowed. The aspect that is investigated in this study is therefore the level of access to course materials while writing a test or exam.

The remainder of the paper is organised as follows: Section 2 discusses some related work on the use and evaluation of authorized cheat-sheets, while Section 3 describes our investigation in the use of cheat-sheets in a

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

computer science module. We discuss our observations in Section 4, and in Section 5 conclude and describe possible applications of the work.

2 Related Work

The use of authorized cheat-sheets is not a new idea and a number of studies have argued for the benefits of this approach, although results on whether their use is beneficial to learning have been mixed. This section discusses some of the related work in more detail.

2.1 The Impact of the Use of Authorized Cheat-sheets on Student Performance

Many studies advocate for the use of cheat-sheets in examinations. Some of the arguments for the use of cheat-sheets include the following:

1. reduced student anxiety as a result of students having some information on hand to consult during the examination [5, 7, 13];
2. increased learning as a side-effect of the actual cheat-sheet construction (the coding hypothesis [6]);
3. promotion of higher-level thinking skills such as problem solving and reasoning [1, 11];
4. improved performance over closed-book [10] and over open-book [9] formats by students during examinations; and
5. less time wastage for students during examination (compared with open-book examinations) [13].

Some studies argue that the use of authorized cheat-sheets does not enhance learning [5]. This could be because students rely on their cheat-sheets and so are less inclined to learn the material [8] (the dependency hypothesis [6]). In contradiction to the dependency hypothesis, Mathew [10] found that there were no differences in the reported study time between students studying for open-book, closed-book and cheat-sheet exams. Song and Thuente [12] found that the use of cheat-sheets helped the weaker students to improve more than it helped the stronger students. The context of all of these studies [5, 6, 8, 10] was the teaching of psychology. It may be possible that the effects of authorized cheat-sheets could differ in the context of different fields of instruction, such as computer science.

In a study of the use of cheat-sheets in an introductory programming module, De Raadt [4] found that the use and compilation of cheat-sheets had a positive impact on student performance. He also found that students who used cheat-sheets (as opposed to those who chose not to use them) showed a larger improvement in marks relative to the mean mark from one assessment to the next. In our study, we wanted to see whether cheat-sheets, similarly, had a positive impact on student performance in a second-year computer science module.

2.2 Content Attributes of Authorized Cheat-sheets

As an extension to investigations related to the impact of using cheat-sheets on the performance of students, researchers have analyzed the contents in student-compiled cheat-sheets. De Raadt [4] rated cheat-sheets in an introductory programming course on two main aspects: layout and content. The analysis of the layout refers to quality aspects of the cheat-sheets. These are discussed in Section 2.3. De Raadt categorized the content of student created cheat-sheets using four binary features: (1) whether code examples were present or not, (2) whether abstract representations were present or not, (3) whether sample answers to examination questions were included or not, and (4) whether the student included unnecessary language references (provided in the examination paper). He found that the inclusion of sample exam answers was negatively related to student performance.

Song and Thuente [12] extended De Raadt's binary classification, by quantifying the number of sample answers, formulae and graph representations on the cheat-sheets. The context was a senior-level computer science networking course. They found that the quality of cheat-sheets was highly related to students' grades.

In this study, we similarly analysed the content of the student cheat-sheets in terms of the quantity of information and the type of information (facts, diagrams, examples or code). We used a density measurement to quantify the content in each of the categories. The aim was to investigate whether a link could be found between these characteristics and student performance.

2.3 Quality Attributes of Authorized Cheat-sheets

Other than the actual content of cheat-sheets, the quality could also relate to student performance.

When rating cheat-sheets in an introductory programming course, De Raadt [4] included quality aspects related to the layout of the cheat-sheets. He categorized the cheat-sheets using three binary features: (1) dense/not dense, (2) organised / not organised, and (3) whether the order on the cheat-sheet matched the course content order or not. De Raadt found that higher performance was positively related to cheat-sheets where the order on the sheet matched the order of the course content and positively related to cheat-sheets that included abstract representations. Song and Thuente [12] extended De Raadt's binary classification, by quantifying different levels of density and organisation on the cheat-sheets.

In the context of a psychology and statistics courses, Mathew [10] rated cheat-sheets on two aspects, namely, richness of detail and the level of organization, but no analysis was conducted based on these ratings. In a medical context, Burke et al. [3] developed a computerized system called QNOTE to evaluate the quality of clinical notes. Apart from identifying the elements that should be part of a clinical note, they identified seven quality aspects to evaluate these notes namely clarity, completeness, conciseness, currency, organization, prioritization, and sufficiency of information.

Inspired by these studies, we evaluated the quality of computer science cheat-sheets on five aspects, namely, abstraction, discretion, coverage, structure and legibility and analysed the impact on students' performance.

3 Gathering Data

3.1 The Context of our Investigation

The context of this study was a second year module called "Computer Organisation and Architecture" at the University of Pretoria. The module involved a theoretical aspect and a practical programming aspect. Theory topics included representation of data on the machine-level; organisation of the machine on the assembly level; the architecture and organisation of memory; inter- and intra-component interfacing and communication; data paths and control; and parallelism. The practical aspect covered assembly programming, in particular 64-bit Intel assembler for Linux.

The module was introduced in 2014 and student performance was not good. The pass rate after the examination (before the supplementary examination) was 58%. Students expressed high anxiety around the examination, possibly due to the wide range of different concepts and levels of thinking required by the students, from high-level understanding of instruction set design to intricate knowledge of assembly programming on a byte-level.

To assist the students, the use of authorized cheat-sheets in the examination was introduced in 2015. The students were instructed to compile a single A4 hand-written page of notes. The reason for specifying hand-written notes was to ensure that students spent individual effort compiling the notes, rather than simply photocopying from the textbook or from other students' notes.

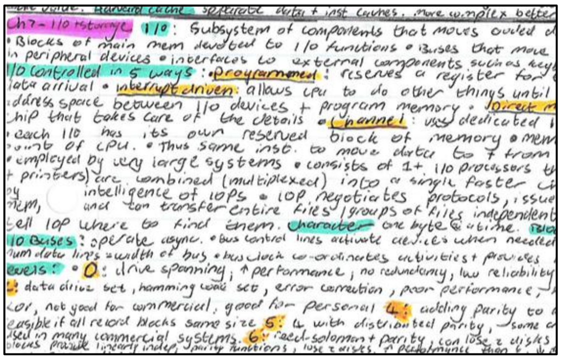
During the semester, students wrote closed-book tests. This provided us with an opportunity to compare the performance of the students when using authorized cheat-sheets in the examination with how they performed in closed-book tests. Of the 87 students who participated in the exam, 75 students submitted their notes. We analysed all the submitted notes.

3.2 Analysing Content

When scanning the pile of submitted cheat-sheets, one of the most obvious distinguishing features between sheets was the volume of content. Some sheets had all possible white space on both sides crammed with minutely written content, while others contained only a few lines of content sparsely filling only half of one side of the permitted A4 sheet. Another distinguishing characteristic was the type of content on the sheets. Some sheets comprised only code listings, some only written facts, while others also included diagrams and worked out examples. To analyse the content of the cheat-sheets, five categories of content were defined as follows:

1. **Empty**: Blank portion with no text or annotations.
2. **Facts**: Textual facts such as definitions or descriptions of advantages/disadvantages of approaches. Figure 1 and Figure 2 show two examples of factual content.
3. **Diagrams**: Graphical representations such as flow diagrams or circuit symbols. Figure 4 shows examples of diagrams.
4. **Examples**: Worked examples such as the calculation of a sample problem or a table of numbers with their binary and hexadecimal values. Figure 3 shows some specimens of content of type 'Examples'.

5. **Code**: The syntax of assembly commands or the code for the implementation of an algorithm. Figure 5 shows two examples of code content.



[illegible] value. Harvard cache separate data + inst caches. more complex better

Ch7 - I/O + storage I/O: Subsystem of components that moves coded d

• Blocks of main mem devoted to I/O functions • Buses that move

in peripheral devices • interfaces to external components such as key

I/O controlled in 5 ways: • Programmed: reserves a register for

data arrival • interrupt driven: allows cpu to do other things until

address space between I/O devices + program memory • Direct m

chip that takes care of the details • Channel: uses dedicated

• each I/O has its own reserved block of memory • mem

point of cpu. • Thus same inst. to move data to + from

• employed by very large systems • consists of 1+ I/O processors

+ printers) are combined (multiplexed) into a single faster c

by intelligence of IOPs • IOP negotiates protocols, issue

mem, and can transfer entire files / groups of files independent

tell IOP where to find them. Character one byte @ a time. Blo

I/O Buses: operate async. • bus control lines activate devices when needed

num data lines = width of bus • bus clock co-ordinates activities + provides

evels: • 0: drive spanning, ↑ performance, no redundancy, low reliability

1: data drive set, hamming code set, error correction, poor performance,

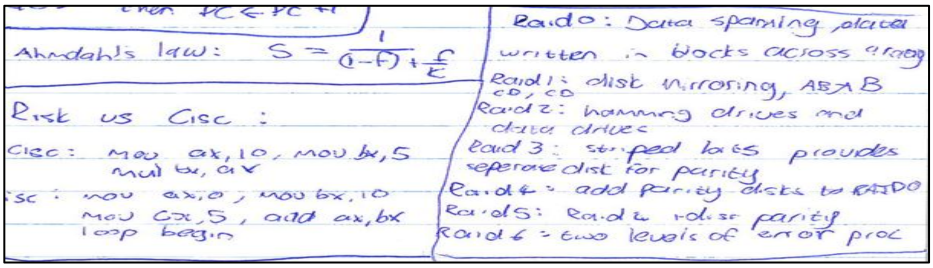
cor, not good for commercial, good for personal 4: adding parity to

easible if all record blocks same size 5: 4 with distributed parity, some

used in many commercial systems. 6: reed-solomon + parity, can lose 2 disks

blocks provide linearly indep. parity functions, lose 2 disks, ↑ performance than [illegible]

Fig. 1. Content showing facts – dense.



then PC ← PC +1

Ahmdahl's law: $S = \frac{1}{(1-f) + \frac{f}{k}}$

Risk vs Cisc:

Cisc: mov ax,10, mov bx,5
mul bx, ax

isc: mov ax,0, mov bx,10
mov cx,5, add ax,bx
loop begin

Raid0: Data spanning, data written in blocks across array

Raid1: disk mirroring, AB→AB
CD, CD

Raid2: hamming drives and data drives

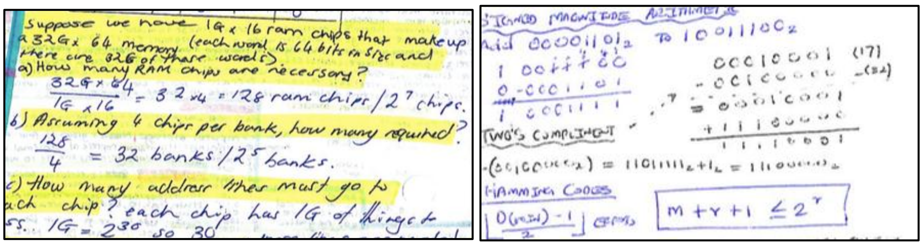
Raid 3: striped bits provides seperate disk for parity

Raid4: add parity disks to RAID0

Raid5: Raid 4 + disk parity

Raid 6: two levels of error proc

Fig. 2. Content showing facts – sparse.



Suppose we have 1G x 16 ram chips that make up a 32G x 64 memory (each word is 64 bits in size and there are 32G of these words)

a) How many RAM chips are necessary?

$\frac{32G \times 64}{1G \times 16} = 32 \times 4 = 128$ ram chips / 2^7 chips.

b) Assuming 4 chips per bank, how many required?

$\frac{128}{4} = 32$ banks / 2^5 banks.

c) How many address lines must go to each chip? each chip has 1G of things to

ss.. $1G = 2^{30}$ so 30

SIGNED MAGNITUDE ARITHMETIC

Add 00001101_2 to 10011100_2

1 0011100
0 - 0001101
1 0001111

00010001 (17)
- 00100000 -(32)
= 00010001
+ 1110000
11110001

TWO'S COMPLIMENT

$-(00100000_2) = 11011111_2 + 1_2 = 11100000_2$

HAMMING CODES

$\left\lfloor \frac{D(min) - 1}{2} \right\rfloor$ errors

$\boxed{m + r + 1 \leq 2^r}$

Fig. 3. Content showing examples.

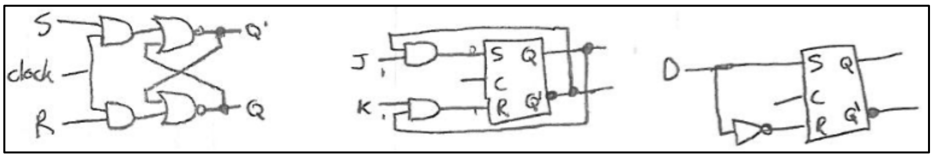
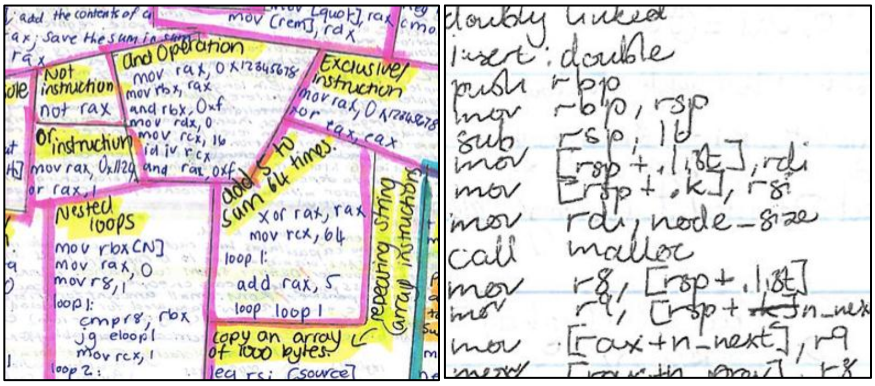


Fig. 4. Content showing diagrams.



```
; add the contents of a
ax; Save the sum in sum
rax

Not instruction
not rax

And Operation
mov rax, 0x12845678
mov rbx, rax
and rbx, 0xf
mov rdx, 0
mov rcx, 16
idiv rcx
and rax, 0xf

Or instruction
mov rax, 0x1129
or rax, 1

Exclusive/ instruction
mov rax, 0x12845678
xor rax, rax

mov [quot], rax
mov [rem], rdx

Nested loops
mov rbx [N]
mov rax, 0
mov r8, 1
loop1:
  cmp r8, rbx
  jg eloop1
  mov rcx, 1
loop2:

add 5 to sum 64 times.
xor rax, rax
mov rcx, 64
loop 1:
  add rax, 5
  loop loop 1

repeating string (array instruction)
copy an array of 1000 bytes.
lea rsi, [source]
```

```
doubly linked
insert: double
push rbp
mov  rbp, rsp
sub  rsp, 16
mov  [rsp+.l,8t], rdi
mov  [rsp+.k], rsi
mov  rdi, node_size
call malloc
mov  r8, [rsp+.l,8t]
mov  r9, [rsp+~~.k~~n_next]
mov  [rax+n_next], r9
```

Fig. 5. Content showing code.

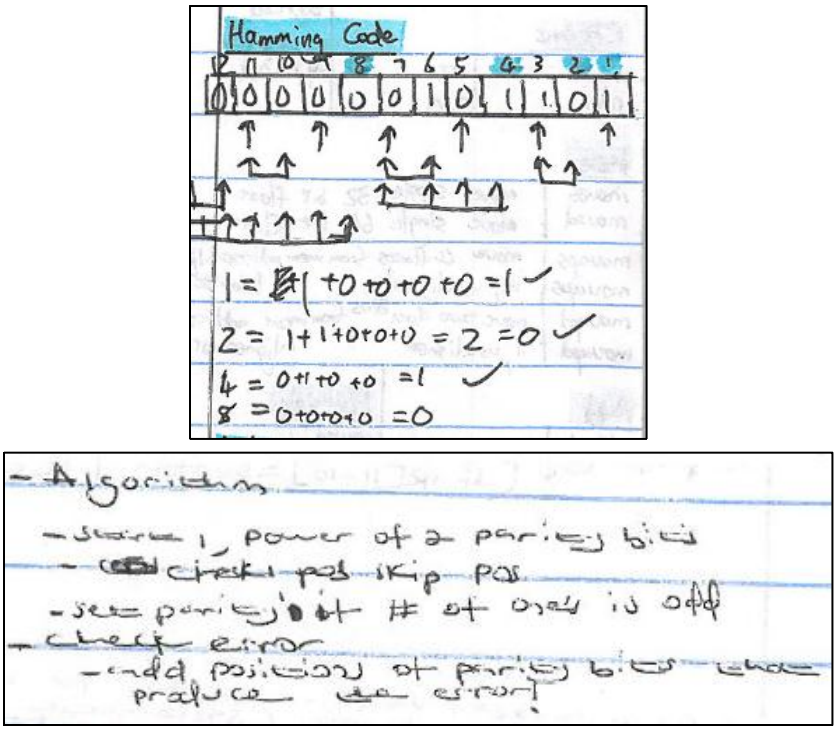
To estimate the quantity of each type of content on each sheet, the number of words in a 2.5cm × 2.5cm unit area containing representative written content was counted (using a cutout cardboard template). This number of words was used as a measure of density for the sheet. The number of 2.5cm × 2.5cm blocks containing facts, diagrams, examples and code on both sides of the sheet were then multiplied by the density measure to quantify the volume for the whole sheet.

No evidence of copying among students was observed during the analysis of cheat sheets. It seemed as if the restriction of having original handwritten cheat sheets did have the desired effect of students compiling their own notes.

3.3 Analysing Quality

The quality of the student notes was assessed using five aspects of quality: (i) Abstraction, (ii) Discretion, (iii) Coverage, (iv) Structure and (v) Legibility. The meaning these aspects is defined by the rubric used to rate each aspect, given in Table 1.

Figure 6 shows an example of content with low and high abstraction of the hamming code algorithm. The top figure shows a worked out example with detail down to the bit level (low abstraction), while the lower figure gives a high-level description of the same algorithm.


Hamming Code
12 11 10 9 8 7 6 5 4 3 2 1
0 0 0 0 0 0 1 0 1 1 0 1

1 = 1 +0 +0 +0 +0 = 1 ✓

2 = 1+1+0+0+0 = 2 = 0 ✓

4 = 0+1+0 +0 = 1 ✓

8 = 0+0+0+0 = 0

- Algorithm
  - start 1, power of 2 parity bits
    - check pos skip pos
  - set parity bit if # of ones is odd
- check error
  - add position of parity bits that produce the error

Fig. 6. Sample extracts showing low (top figure: worked example) and high abstraction (lower figure: description of algorithm) content of the hamming code algorithm.

Each cheat-sheet was assigned a value from 0 to 2 for each aspect of quality as defined in Table 1. A total measure of quality out of 10 was then calculated as the sum of the five aspects of quality.

Table 1. Rubric used for assessing the quality of the student notes.

Aspect of Quality	**0**	**1**	**2**
Abstraction	Too much detail, either full sentences or long code listings	Some summaries or categorisation	Good abstraction of concepts, high level algorithms in place of detailed code
Discretion	Includes many unimportant facts/aspects of the content	Some important facts/aspects, but some not important	Notes reflect the essential concepts
Coverage	Only a few aspects of the course are covered in the notes	Notes partially cover the breadth of the course	Notes cover the full breadth of course topics
Structure	Almost no structure in notes	Some structure in notes	Notes are well structured allowing concepts to be found easily
Legibility	Notes barely legible, messy	Notes are fairly legible	Notes are highly legible

4 Results

4.1 The Impact of Introducing Cheat-sheets on Overall Performance

Our first investigation involved determining if the introduction of authorized cheat-sheets had an impact on exam performance for the students who were required to use cheat-sheets compared with the performance of students who wrote a closed book exam. We compared the marks of the students in the exam of 2014 and the exam of 2015. In 2014, the exam was closed book while in 2015 the students were authorized to use their notes. The summary statistics of the two exams are shown in Table 2. Although the average did not change from 2014 to 2015, the distribution of marks changed slightly with a lower maximum, a lower minimum and a larger standard deviation.

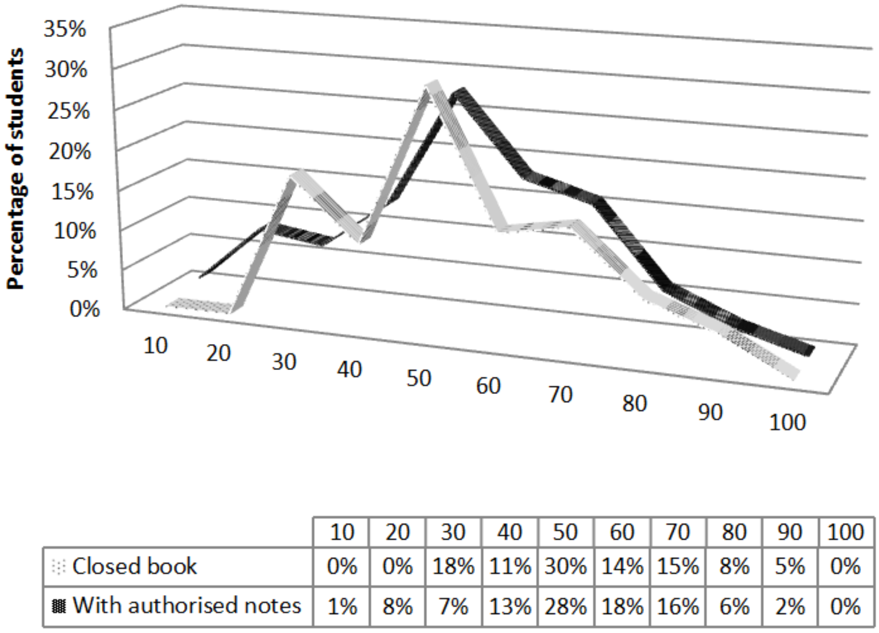
Table 2. Statistics of the exam marks of COS284 student in 2014 and 2015.

	2014	**2015**
Number of students	66	83
Average exam mark	48	48
Median exam mark	48	48
Maximum exam mark	86	83
Minimum exam mark	21	10
Standard deviation	16.3	17.0

Figure 7 shows the distribution of the examination marks in the two years. Since the total number of participants in these events are different, the values are normalised to percentages for comparison. The graph and the table in Figure 7 show the percentage of students in ten performance classes: 0 – 10%, 11 – 20%… 91 – 100%. The distribution of marks for the exam with authorized notes seems to be closer to a normal distribution compared with the distribution of marks for the closed book exam.

More students who participated in the exam with cheat-sheets were in the lowest two classes (0 – 20%) than students who participated in the closed book exam. This supports the notion that poor students may fare worse with the use of cheat-sheets than in closed book exams. This could be attributed to students' inability to compile useful notes or to reduced preparation time due to false confidence based on the availability of notes.

Although the poorer students fared worse in 2015, the total percentage of students failing the examination (with marks under 50%) was slightly lower in 2015 (57%) than in 2014 (59%). This supports the notion that borderline students may fare better when they have access to notes. It is possible that borderline students are likely to fail closed book exams due to failed short-term memory, while the availability of notes contributes to reduced anxiety leading to better performance. The overall pass rate in the COS284 module improved from 58% in 2014 to 62% in 2015.



	10	20	30	40	50	60	70	80	90	100
⁘ Closed book	0%	0%	18%	11%	30%	14%	15%	8%	5%	0%
▩ With authorised notes	1%	8%	7%	13%	28%	18%	16%	6%	2%	0%

Fig. 7. Distribution of the exam marks of COS284 in 2014 and 2015

4.2 The Impact of Using Cheat-sheets on Individual Student Performance

Our second analysis involved comparing how the performance of the same cohort of students differed when using cheat-sheets or not. In our sample of 75 students, 57.3% of the students performed worse in the examination using cheat-sheets than in the closed book summative assessment tests, which is contrary to Mathew [10] who observed overall improved performance when using cheat-sheets. This decrease in performance could be due to factors other than the cheat-sheets, such as increased content in the summative assessment. It is, however, interesting to analyse which students performed worse and which performed better.

Figure 8 plots the performance of students in closed-book tests compared with the written examination, showing the averages of groups of five students sorted by final mark. For example, for the group of five students who had the lowest final marks (with an average of 31%), the average mark obtained by these five students for the closed book tests was 34%, whereas the average mark obtained for the examination was 16%. Similarly, the other students with low final marks (below the 50% pass mark) performed worse when using cheat-sheets. A number of the middle achieving groups, however, achieved higher marks when using cheat-sheets. These findings are the opposite of Song and Thuente [12] who found that the use of cheat-sheets helped their weaker students to improve.

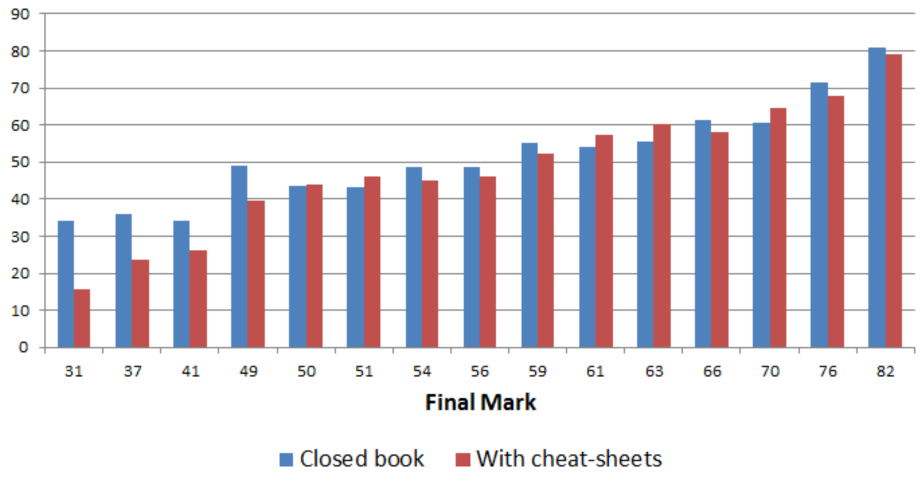


Fig. 8. Final marks plotted against marks for each way of examination.

The impact of the type of content and the quality of the cheat-sheets on the improvement in student performance are investigated in the following sections.

4.3 Cheat-sheet Content and Student Performance

The horizontal axis of Figure 9 plots the average change in mark from closed book tests to examination (using cheat-sheets) of five consecutive students after sorting all the students by the value of this change. For each of these groups the bar shows the average number content-units (where one content unit is equivalent to 1000 words of written text) filled with each of the types of content. For example, the five students with the largest decrease in marks (an average of -22%) from tests to examination are plotted as the first column in the figure.

There is no apparent trend in Figure 9. The two groups of five students who provided the most content on average on their cheat-sheets achieved a decrease of 13% and an increase of 12%. In addition, there is no pattern in the type of content across the performance classes. We conclude that there is no relation between the volume or type of content on the cheat-sheets and the change in student performance.

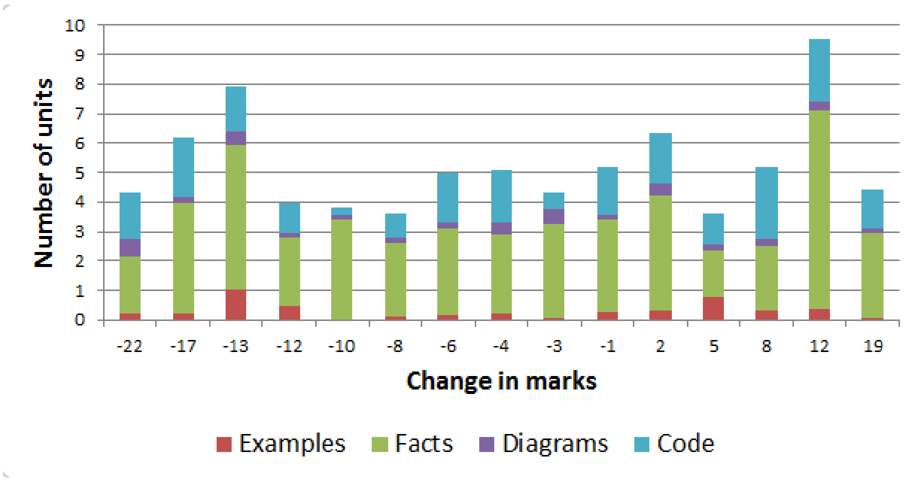


Fig. 9. Change in mark plotted against cheat sheet content.

4.4 Cheat-sheet Quality and Student Performance

Figure 10 shows the average quality per aspect for the same student groups as in Figure 9. Here it seems that there is a positive correlation between higher quality of the cheat-sheets and the improvement in performance of the students. Higher values associated with better discretion and abstraction is evident on the high side of the graph, i.e. for the students who showed greater improvement. This points to a possible future intervention strategy whereby students should be guided to create cheat-sheets of higher quality. Song and Thuente [12] found that in a class of engineering students, when students were able to improve the quality of their cheat-sheets between examinations, their grades improved. A possible intervention strategy could therefore be to assist low-performing students in the compilation of high quality cheat-sheets, with the hope that examination preparation and performance can be improved.

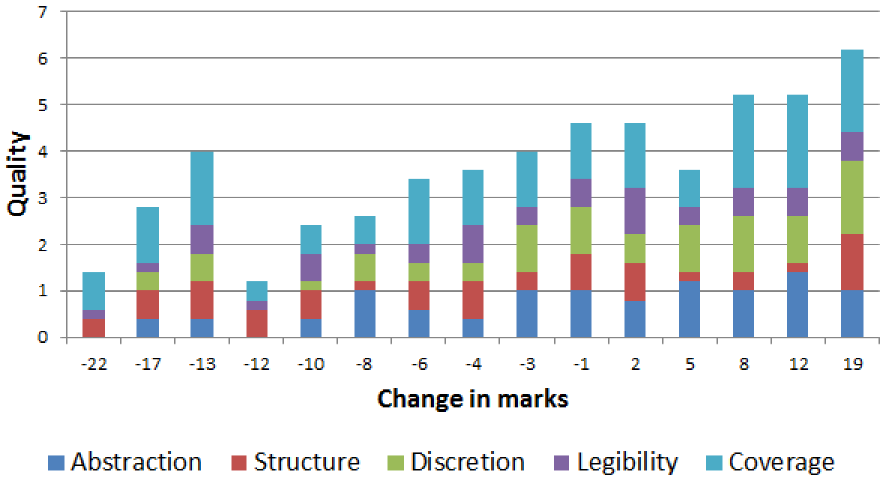


Fig. 10. Change in mark plotted against cheat sheet quality.

5 Conclusion

In this paper, we investigated the use of authorized cheat-sheets in the final examination of a second year computer science module on computer architecture and assembly programming. We described how we evaluated and classified these sheets. Our classification of content types as well as the rubric we used for quality assessment could be useful in other contexts.

Results show that the use of authorized cheat-sheets did not have a significant overall impact on student performance. For most students, especially the weakest ones, it had a negative impact. We have, however, seen that some students improved their marks significantly when using cheat-sheets and that the sheets created by these students were generally of a higher quality than the other students, especially when considering discretion and abstraction.

We conclude that the use of authorized cheat-sheets is beneficial to students who are willing and able to produce quality cheat-sheets. The introduction of cheat-sheets does not improve student performance by default, but it has the potential to be used as an educational tool to assist students to prepare better for examinations.

We believe that students can benefit more when required to compile such cheat-sheets if they are made aware of the learning opportunity imposed by the action and guided to put the correct kind of effort into creating these cheat-sheets.

References

1. P. K. Agarwal, J. D. Karpicke, S. H. K. Kang, H. L. Roediger III, and K. B. McDermott. Examining the testing effect with open- and closed-book tests. Applied Cognitive Psychology, 22(7):861 – 876, Nov 2008.
2. D. Boniface. Candidates' use of notes and textbooks during an open-book examination. Educational Research, 27(3):201–209, 2008.
3. H. B. Burke, A. Hoang, D. Becher, P. Fontelo, F. Liu, M. Stephens, L. N. Pangaro, L. L. Sessums, P. O'Malley, N. S. Baxi, C. W. Bunt, V. F. Capaldi, J. M. Chen, B. A. Cooper, D. A. Djuric, J. A. Hodge, S. Kane, C. Magee, Z. R. Makary, R. M. Mallory, T. Miller, A. Saperstein, J. Servey, and R. W. Gimbel. QNOTE: an instrument for measuring the quality of EHR clinical notes. Journal of the American Medical Informatics Association : JAMIA, 21:910 – 916, Sep-Oct 2014.
4. M. de Raadt. Student created cheat-sheets in examinations: Impact on student outcomes. In Proceedings of the Fourteenth Australasian Computing Education Conference, pages 71–76, 2012.

5. K. L. Dickson and J. J. Bauer. Do students learn course material during crib sheet construction? Teaching of Psychology, 35(2):117–120, 1985.
6. T. N. Dorsel and G. W. Cundiff. The cheat-sheet: Efficient coding device or indispensable crutch? The Journal of Experimental Education, 48(1):39–42, 1979.
7. B. Erbe. Reducing test anxiety while increasing learning: The cheat sheet. College Teaching, 55(3):96–98, 2007.
8. S. C. Funk and K. L. Dickson. Crib card use during tests: Helpful or a crutch? Teaching of Psychology, 38(2):114–117, 2011.
9. K. H. Larwin, J. Gorman, and D. A. Larwin. Assessing the impact of testing aids on post-secondary student performance: A meta-analytic investigation. Educational Psychology Review, 25(3):429–443, 2013.
10. N. Mathew. Student preferences and performance: A comparison of open-book, closed book and cheat sheet exam types. In Proceedings of the National Conference on Undergraduate Research, 2012.
11. S. R. Smith. Is it time to close the book on closed-book examinations? Medicine and health, Rhode Island, 82(8):285 – 288, August 1999.
12. Y. Song and D. Thuente. A quantitative case study in engineering of the efficacy of quality cheat-sheets. In Frontiers in Education Conference (FIE), 2015, pages 1–7, Oct 2015.
13. K. Trigwell. The crib card examination system. Assessment & Evaluation in Higher Education, 12(1):56–65, 1987.

The concept of mobile applications as educational tool to enhance information security awareness

Lynette Drevin, Albie le Grange, Martin Park

Computer Science & Information Systems, North-West University,
Potchefstroom, South Africa
Lynette.Drevin@nwu.ac.za, albie.legrange.rsa@gmail.com, martinpark02@gmail.com

Abstract: There is a growth in the use of mobile devices for a variety of applications, such as financial, healthcare and location-based applications. More and more sensitive information is used and stored by these software systems. Mobile devices are often not as securely protected as personal computers. Often the users of these devices are not aware of probable security threats or they are ignorant of their own shortcomings or their potentially unsafe behaviour. Cyber users may behave more securely when their security awareness is raised and when they know themselves better in order to be more mindful in their online behaviour. The aim of this paper is to describe two mobile applications that were developed in two recent studies to indicate how the concept of mobile apps may be used as education tools. The purpose is to promote secure behaviour and enhance security awareness of cyber users. The approach followed was to do empirical studies where after the results were used in certain aspects of the apps. The apps were evaluated by the users to get their feedback. The findings indicate that mobile apps may be used to raise information security awareness. Users gained insight into their own personality traits that may have an influence on their security behaviour. The value of this research is that these applications or their concepts can be taken further and additional apps can be developed after research into specific contexts in order to raise awareness. This may promote more secure behaviour and make users more mindful about their own actions. Especially young people who are increasingly using mobile devices may find it appealing to use mobile apps as an educational platform for the improvement of their security awareness and behaviour.

Keywords: Information security awareness, mobile application, apps, mobile development, secure behaviour, personality traits, education.

1 Introduction

During the last decade a plethora of articles were written on the use and development of mobile applications (mobile apps). There is an improvement in hardware and network capabilities relating to mobile computing and thereby enhancing the capabilities of mobile devices [1]. Literature also indicates the growth in information security incidents in companies or experienced by individual home users. These incidents include for example virus attacks, ransomware attacks and phishing attacks. [2, 3]. These are examples of threats that cyber users should be knowledgeable about in order not to fall prey to such malicious attempts. Apart from directed attacks there are also the unmindful or ignorant computer users who may be their own worst enemy by acting unsafely without even recognizing it due to a lack of knowledge. The use of mobile devices has increased tremendously over the past few years. However, applications, programs and data stored on mobile devices are not as securely protected from unauthorized access or other threats than those on most desktop computers [4]. Additionally, users are not conscious about protecting their smartphones at the same level of security as they do their computers [5]. Therefore, in this domain, the vulnerability of cyber users is on the increase. Information security education may be an important way of enhancing secure cyber and the online behaviour of computer and mobile device users.

The aim of this paper is to report on two separate studies that were conducted regarding (1) information security awareness of users and (2) information security behaviour and users' personalities. The scope of the first study [6] was information security awareness of students and the use of mobile devices (MSA). The second study [7] focused on personality traits of computer users and their online security behaviour (PSB). The paper reports not on the complete content of the two studies, but rather focuses on specific secondary outcomes of the studies, namely the development of mobile applications as a result of the investigations carried out in these studies. It is

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

shown in this paper how the concept of mobile apps can be used as an educational tool by a computer user to improve security awareness by:

a) Raising information security awareness;

b) Getting insight into one's own personality traits and understanding how this may have an effect on one's information security behaviour.

The remainder of the paper is structured as follows. In Section 2, a brief theoretical background of information security, awareness, education, behaviour, and personality traits is presented. In Section 3, the methodology used in the studies and the development of the mobile apps are briefly discussed. In Section 4, the layout and description of the apps are presented. In Section 5, interpretations and feedback on the apps are offered. The paper is then concluded in Section 6 with final remarks.

2 Theoretical background

2.1 Information security

Overall, information security is viewed or described as the protection of confidentiality, integrity and availability of information and/or computer resources [8]. Often other goals are also set such as usability, non-repudiation, possession and accuracy [9]. In order for computer users to adhere to these goals, they need to be knowledgeable and mindful about possible threats, vulnerabilities and methods of exploitation and their behaviour in cyber space should be secure with regard to potential security hazards. Many technical precautions can be put into place to protect online and computer resources but people are often seen as the weakest link [8,9,10] regarding information security. Therefore they should be informed and trained to act in a secure way in the cyber environment.

2.2 Information security awareness and education

Much research has been done and more is ongoing on the topic of information security awareness and education. An example of a bigger initiative is that of Kortjan and von Solms [11] where a conceptual framework for cyber security awareness and education in South Africa is proposed. Five layers are proposed in this framework and are shortly as follows:

- The Strategic Layer. This layer is on governmental level.
- The Tactical Layer. This layer suggests schemes that a country should have for cyber security awareness and education goals.
- The Preparation Layer. This layer looks at the content of the previous level.
- The Delivery Layer. This level defines the receiver or beneficiary of the previous level (participants).
- The Monitoring Layer. This layer focuses on the progress with the schemes.
- Resources. This component defines the resources that have to be used in previous layers.

The work reported in this paper is mostly on the delivery layer. Young people are identified as the audience/receiver of information security efforts. These include mostly university students as they are seen as the future information technology professionals and users. In the first phase the students are the participants in two separate surveys regarding information security awareness (1) and correlation of personality and information security behaviour (2). In the second phase two mobile applications were developed and are presented to the participants of the two studies to use the apps and evaluate them.

According to a study of Walaza et al. [12] mobile device use is on the increase among young people. Mobile devices are used for a variety of applications such as social media and instant messaging but also increasingly for academic purposes. It is essential for all users using the Internet to act in a secure way. Several academic institutions include information security courses in their academic programs. Future information security professionals and security aware residents are shaped by education and awareness programs when these topics are offered.

Previous research indicated how non-conventional methods can be used for cyber security awareness training in an engaging format such as video games. Cone et al. [13] demonstrated how the game CyberCIEGE was used to adhere to a specific set of Navy Information Security and Assurance training requirements. However the game was flexible enough to be used in a variety of environments and also included a range of security topics. In this paper the use of mobile apps is discussed as a way to enhance information security awareness among young people. This is demonstrated by using new technology that may be appealing to youngsters seeing they are so accustomed of having mobile devices with them most of their time.

Security awareness programmes are often put into practice to raise the awareness levels of users. Kruger and Kearney [14] and Kruger et al. [15] list three factors of security awareness, namely knowledge: what the users know, attitude: what the users think, and behaviour: what the users do. The two studies referred to in this paper had security awareness and behaviour as overall topics and specifically security awareness of mobile users (MSA) and the correlation between computer users' personality and their security behaviour (PSB).

2.3 Secure behaviour

Information security behaviour can be regarded as the security-related behaviour that users demonstrate when they use or interact with personal or corporate information and interrelated online resources of a sensitive kind, whether using a computer or mobile device [7]. Information security is not only described in terms of physical or software security, but also includes people and other computer users who are in control of or use these security systems and functions. It is often stated that security is like a chain and a chain is only as strong as its weakest link. The human is still said to be the weakest link concerning information security [10]. Each individual has certain personality traits and this has shown to have an influence on his/her information security behaviour, be it intentional or unintentional risky behaviour [16].

2.4 Personality traits

The five-factor model of personality was introduced by McCrae and John [17]. This has become the most important description of personality traits in current psychology [18]. The personality traits of individuals can be assessed by using a measuring instrument consisting of personality-related questions. The five factors are:

- Neuroticism
- Extraversion
- Openness to experience
- Conscientiousness
- Agreeableness

McBride et al. [19] used the five-factor model of personality in a study and found that people with different personalities responded in different ways to several information security settings. Certain conclusions are drawn when analyzing personality and information security behaviour [19]. For example: Open, Agreeable and Conscientious individuals are more likely to respect information security guidelines, whereas more Neurotic and Extroverted individuals are more likely to violate security guidelines. The PSB study used this five-factor model to assess computer users' personality traits and their information security behaviour and suggest ways to improve their online behaviour.

The above discussions give insight into the problem-domains of the security awareness and education, behaviour and personality of computer/mobile device users. The focus of this paper is to indicate how mobile applications were used in two studies to assess and involve users in the information security issues. Mobile application development has grown immensely with the increased use of mobile devices. Mobile application development can be described as a practice whereby applications are developed for hand-held devices including smartphones and tablets [20]. In a study of Ophoff and Robinson [21] end user smartphone security awareness was also assessed in a South African context indicating that the public are complacent in their security behaviour showing high levels of trust towards the app stores. They do not often consider privacy and security as important when installing and downloading new applications. These results are in line with what the study of Park [6] found amongst young people when assessing their mobile device security awareness and perceptions.

The next section briefly presents the methodology of the two studies and then specifically the methods used for developing the mobile apps.

3 Methodological approach

3.1 Research methodology

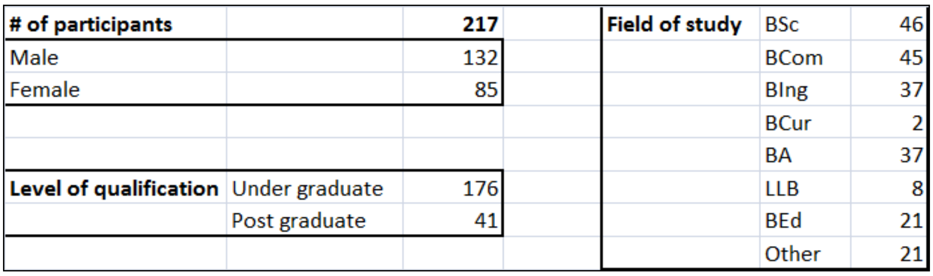
Both of the studies consisted of two phases. An empirical study was done in each setup with a description and write up of the results. After this a mobile app was developed for each study using some of the survey results in the apps as information to app users. The apps were presented to the participants of the first survey to use and evaluate. The general methodology followed in these two studies was to use online questionnaires to collect data from the participants. The questionnaires consisted of Lickert scale and yes/no questions as well as a few open questions to get comments or opinions. Informed consent was asked from respondents and the ethical risk level was assessed as low to no risk. The first phase of the two studies is summarized in table 1.

Table 1. Information about the two studies – phase 1: Survey

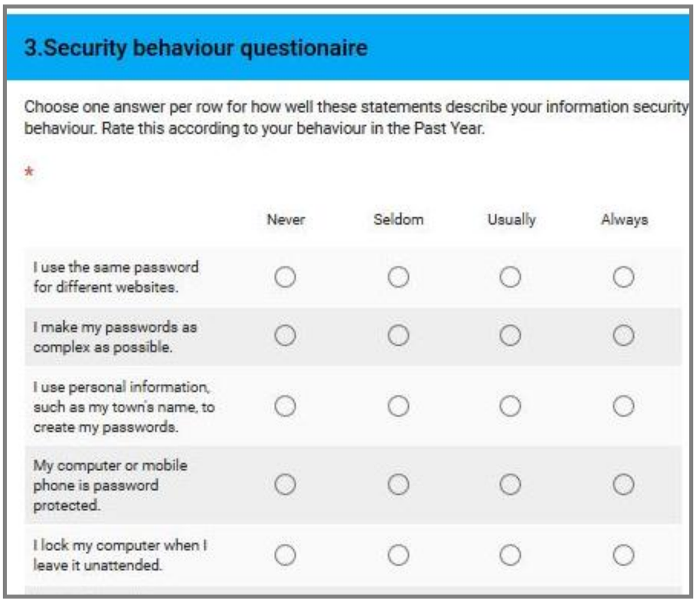
Aspect of study	**Mobile device security awareness of young people (MSA) Study - 1**	**Personality traits of computer users and their online security behaviour (PSB) - Study 2**
Research question	What are the perceptions and attitudes of young users regarding mobile device security?	What is the correlation between the personality and the information security behaviour of a computer user?
Aspects assessed	Demographics, knowledge, attitude, behaviour, open issues	Demographics, personality questions, information security behaviour, open issues
Data collection	Online questionnaire: Examples • Demographic data: Age, gender, etc. • Knowledge: *What is encryption?* • Attitude: *Security software is essential for mobile devices* • Behaviour: *I always read the licence agreement when installing a new application*	Online questionnaire: Examples • Demographic data: Age, gender, etc. • Personality questions to determine highest personality trait: *I feel comfortable around people* • Information security behaviour questions: *My computer or mobile phone is password protected* (An example is shown in Figure 1)
Data analysis	Descriptive and interpretive statistical methods, Factor analysis, T-tests, effect sizes	Descriptive and interpretive statistical methods, Factor analysis, T-tests, effect sizes
Examples of results	• Knowledge: The participants generally know the most common security terms; Users must be made more aware about all the threats and vulnerabilities regarding mobile device security - use educational programmes in schools and universities and community initiatives. • Attitude: There are mixed opinions from the participants regarding the secureness of apps on mobile devices. There is definitely a need for an increase in mobile device security • Behaviour: Act with caution when downloading online content on the mobile device, thereby taking responsibility for own actions. ,	A statistical significant relationship exists between Conscientiousness as highest personality trait and total security threat score of a user.

For the MSA study there were 217 respondents which completed the questionnaire. They were students from a South African university from different study areas. Some demographic information is shown in table 2.

Table 2. Demographic data of the MSA study



# of participants		**217**		**Field of study**	BSc	46
Male		132			BCom	45
Female		85			BIng	37
					BCur	2
					BA	37
Level of qualification	Under graduate	176			LLB	8
	Post graduate	41			BEd	21
					Other	21



3.Security behaviour questionaire

Choose one answer per row for how well these statements describe your information security behaviour. Rate this according to your behaviour in the Past Year.

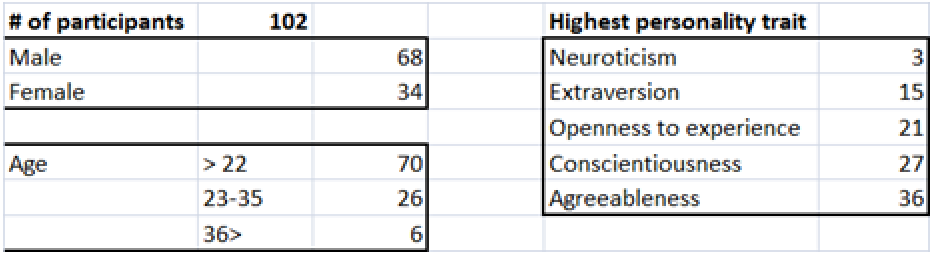
*

	Never	Seldom	Usually	Always
I use the same password for different websites.	○	○	○	○
I make my passwords as complex as possible.	○	○	○	○
I use personal information, such as my town's name, to create my passwords.	○	○	○	○
My computer or mobile phone is password protected.	○	○	○	○
I lock my computer when I leave it unattended.	○	○	○	○

Fig. 1. Example of security questions [7]

For the PSB study there were 102 participants that completed the survey. Certain demographic and personality data is show in table 3.

Table 3. Demographic and personality data of the PSB study



# of participants	**102**			**Highest personality trait**	
Male		68		Neuroticism	3
Female		34		Extraversion	15
				Openness to experience	21
Age	> 22	70		Conscientiousness	27
	23-35	26		Agreeableness	36
	36>	6			

From the survey done in this PSB study and from studies found in the literature [22,23] on the topic of information security behaviour and personality, the results in table 4 show an example of what was found in this study and in others when interpreting the security behaviour of users and their associated highest personality traits.

Table 4. Example of Agreeableness and threats and strength of respondents [7]

Highest personality trait	Found in this survey	Literature [22,23]
Agreeableness "People that are agreeable are trusting, sympathetic, modest, cooperative and straightforward. These people are good with conflict resolution and therefore less argumentative. Being empathetic and the tendency to help people are other characteristics of agreeableness. The opposites of agreeableness are aggressiveness, antagonism and suspiciousness."	Threats: -Program security -Unsafe password behaviour Strengths: -Physical security	Threats: -Social engineering Strengths: -Tend to follow security guidelines

After phase 1 of each study – as a secondary objective, a mobile application was developed – each with its own aim. Table 5 indicates certain aspects of each application.

Table 5. Information about the two applications – phase 2: App development

Aspect of app	**Mobile device security awareness of young people (MSA) – app 1**	**Personality traits of computer users and their online security behaviour (PSB) – app 2**
Software development methodology used for the application	MASAM (Mobile Application Software Agile Methodology)	An adapted version of James Martin's Rapid Application Development Methodology (JMRAD)
Aim of a mobile app	The aim was to help with the information security awareness of mobile device users and to increase their knowledge and to help users act more securely when using their mobile devices. A quiz on information security aspects was presented to the users, information on the survey was displayed and feedback from the users was required.	The main aim of the app was to host a personality test - the Five-Factor IPIP (International Personality Item Pool) test, in the form of a questionnaire. The results are displayed on a bar graph showing the personality factor scores. A quiz was presented to the users to assess their knowledge on information security aspects. Information on the survey was also presented to the app users. Tips on how to act securely depending of the type of personality was offered to the users.
Issues for evaluation of application	Questionnaire: • User friendly • Use the app without written instructions • Easy navigation • Keep me engaged • Easy to learn • App response is adequate • Designed for all levels of users • Learned more about mobile device security • Became more aware of mobile device security • Quiz questions were useful for the purpose of the app • Any other issues	Oral feedback: • User friendly • Easy navigation • Learned more about information security behaviour • Became more aware of own personality traits • Any other issues

The details of the specific results of the studies are beyond the scope of this paper as the main focus is on the development, applicability and usability of the apps as a concept of an educational tool to enhance security awareness. Details of the main studies can be obtained from [6,7].

3.2 Development of the applications

In order to decide how the mobile apps had to be developed, sources on software development methodologies were used to search for solutions. There are different routes that can be followed to develop mobile applications. Flora and Chande [20] propose the agile approach, stating it is a likely fit for the development of mobile applications. Mobile D, Hybrid Methodology Design, MASAM (Mobile Application Software Agile Methodology) and SLeSS (Scrum and Lean Six Sigma) were considered as development methodologies.

For the first app (MSA study), the MASAM development methodology in which an agile approach to quickly develop mobile applications was mainly applied [6]. For the second app (PSB study), an adapted version of James Martin's Rapid Application Development Methodology (JMRAD) was used [7]. This was a one-person-as-developer adaption thereby removing certain aspects of the original JMRAD methodology. For both apps, the Android platform was used as being the most popular and widely used operating system on mobile devices [24]. Both apps also had an evaluation phase in order to get feedback from users.

In the next section layouts and discussions regarding the two apps are presented.

4 Layout and description of the two mobile apps

4.1. Mobile security awareness – app 1

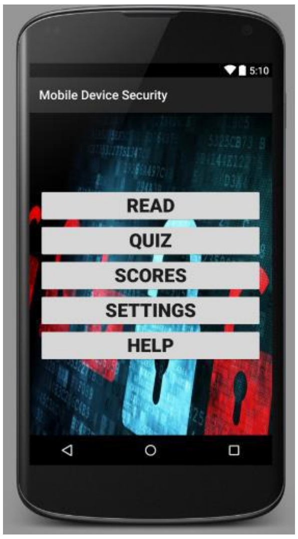
For study 1 the aim of the app was to aid in the information security awareness of mobile device users and to increase their knowledge and to help them to act more securely when using their mobile devices. This was done by presenting information on these topics and offering a quiz to assess knowledge of the app users. Findings of phase 1 – the survey were also presented to the app users.

The Main Menu consists of five buttons:

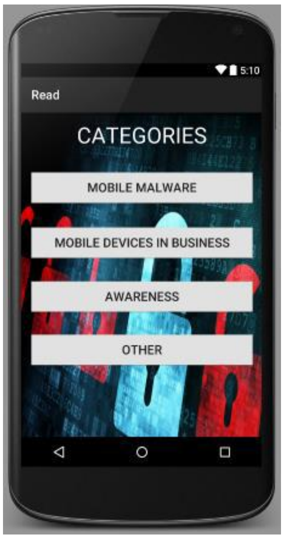
- Read: When the read button is clicked the user can choose between the following categories: Mobile malware, Mobile devices in business, Awareness, and Other. The user can read about these subjects when he/she clicks on them. The read data is stored in a database.
- Quiz: The quiz activity starts with a question and three possible answers. The user must select the correct answer. If the user is correct one mark will be added to the score. The quiz continues until a wrong answer is selected. The quiz questions and answers data are stored in a database.
- Scores: In the scores activity the top three scores for the quiz appear on the screen of the device.
- Settings: The settings activity has the option to reset all the scores. There is also a link to evaluate the application and to get feedback from the users.
- Help: The help activity consists of text that gives more information about the application and shows the version number of the application.

The following images of the screens indicate some of the main screens of the app [6]: - They include the main menu (screen 1) to choose an option from;

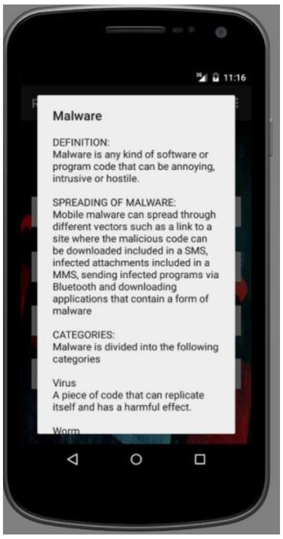
- To read information on different categories (screens 2,3) on information security topics (e.g. mobile malware, mobile devices in business, findings of this particular study and awareness);
- To take part in a quiz (screen 4) (e.g. Trojan horse, worm, firewall, encryption, strong password, etc.), and to show the users' results (e.g. 10/20);
- To show top scores; and to do some housekeeping work where scores can be reset; and feedback can be given.
- Help information could be displayed as well.



Screen 1: Main menu screen



Screen 2: Read screen



Malware

DEFINITION:
Malware is any kind of software or program code that can be annoying, intrusive or hostile.

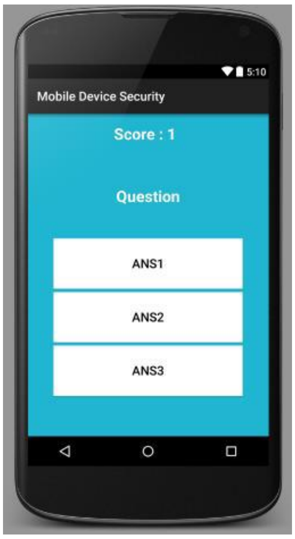
SPREADING OF MALWARE:
Mobile malware can spread through different vectors such as a link to a site where the malicious code can be downloaded included in a SMS, infected attachments included in a MMS, sending infected programs via Bluetooth and downloading applications that contain a form of malware

CATEGORIES:
Malware is divided into the following categories

Virus
A piece of code that can replicate itself and has a harmful effect.

Worm

Screen 3: Read category screen



Screen 4: Quiz screen

An example of the quiz question is:

"What is a computer virus?"
1. "A program that is used to communicate with other people"
2. "A piece of code that is imbedded in programs that can replicate itself with a harmful effect"
3. "A program that protects the hardware." [6]

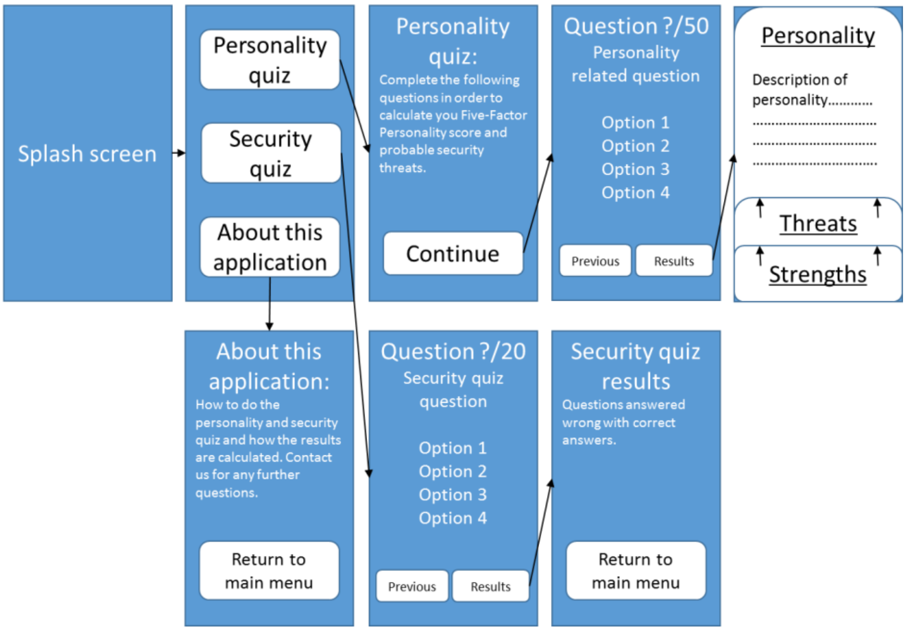
The app was loaded on Android Play Store from where users could download it and use it and give comments. The feedback on the use and evaluation of the app is presented in Section 5.

4.2. Personality and security behaviour – app 2

The aim of this app was to host the personality test and display the results to the app user. A quiz was accessible to users on information security related terms. Tips were stated as how to behave more securely depending on one's personality traits.

The survey results from phase 1 of the study were also presented to the app users. The information was used in the app to make users aware of their personality and possible security threats to which they may be vulnerable. This information may be useful in a company, for example: An employee having a lower Conscientiousness score than other employees may have more unsafe information security behaviour than other employees. This can also be formulated to state that an employee with a high Conscientiousness score may have more secure behaviour concerning information security. Other factors may, however, also influence security behaviour, for instance training, experience and awareness in this area.

Figure 2 presents the planned layout of this app indicating the different screens and navigation between screens.



Complete the following questions in order to calculate you Five-Factor Personality score and probable security threats.

Description of personality...........
....
....
....

How to do the personality and security quiz and how the results are calculated. Contact us for any further questions.

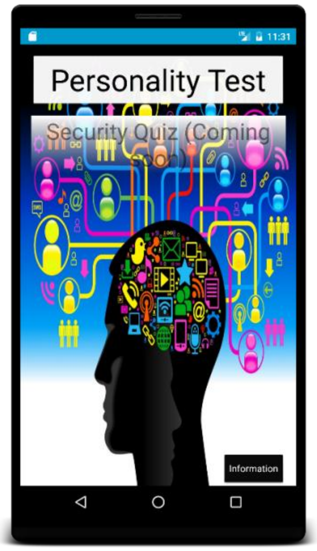
Questions answered wrong with correct answers.

Fig. 2. Planned layout and functionality of mobile application [7]

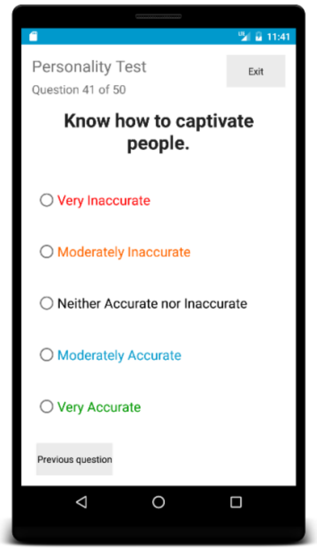
The mobile application home screen contains the personality test button that contains the main functionality of this application screen. The home screen also contains a Security Quiz button and an About the application button as seen in Figure 2.

After this, instructions are given on how to complete the questionnaire. The personality questionnaire follows which consists of 50 personality related questions. These questions can be answered by tapping one of the five descriptions that agrees most with the view of the user. There is also the functionality to return to a previous question if the user has tapped on the wrong answer.

The next screenshots of the app that was developed for the Android platform present certain aspects of the app [7].



Screen A: Splash screen


11:41

Personality Test
Question 41 of 50

Exit

Know how to captivate people.

○ Very Inaccurate

○ Moderately Inaccurate

○ Neither Accurate nor Inaccurate

○ Moderately Accurate

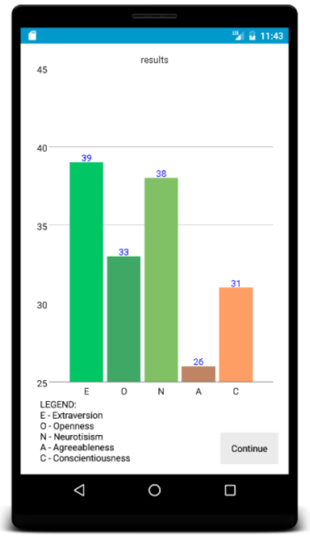
○ Very Accurate

Previous question

Screen B: Personality test question

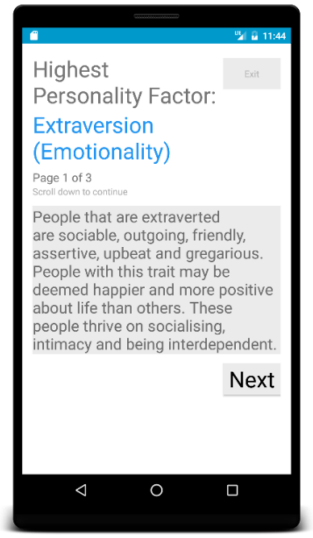
The app assessed the personality traits (screens A, B) of the user following the model of Five-Factor Model of Personality.

A quiz was also presented to the user asking questions on information and computer security topics. The personality results of the user were displayed to the user showing the scores of each trait and a description of the highest factor (screens C, D).



LEGEND:
E - Extraversion
O - Openness
N - Neurotisism
A - Agreeableness
C - Conscientiousness

Screen C: Personality score

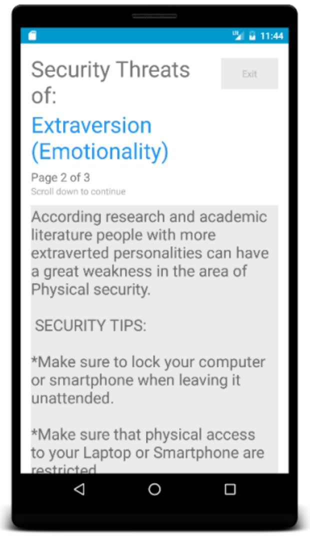


Highest Personality Factor:
Extraversion (Emotionality)

Page 1 of 3
Scroll down to continue

People that are extraverted are sociable, outgoing, friendly, assertive, upbeat and gregarious. People with this trait may be deemed happier and more positive about life than others. These people thrive on socialising, intimacy and being interdependent.

Screen D: Description-personality trait



Security Threats of:
Extraversion (Emotionality)

Page 2 of 3
Scroll down to continue

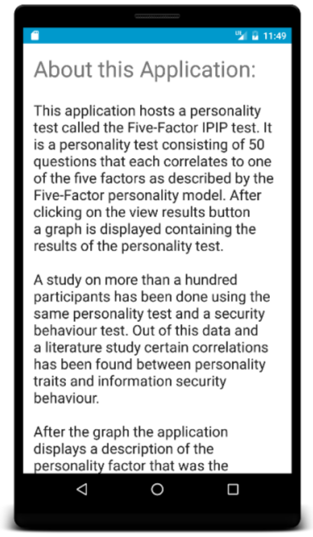
According research and academic literature people with more extraverted personalities can have a great weakness in the area of Physical security.

SECURITY TIPS:

*Make sure to lock your computer or smartphone when leaving it unattended.

*Make sure that physical access to your Laptop or Smartphone are restricted

Screen E: Possible security threats per highest personality trait



About this Application:

This application hosts a personality test called the Five-Factor IPIP test. It is a personality test consisting of 50 questions that each correlates to one of the five factors as described by the Five-Factor personality model. After clicking on the view results button a graph is displayed containing the results of the personality test.

A study on more than a hundred participants has been done using the same personality test and a security behaviour test. Out of this data and a literature study certain correlations has been found between personality traits and information security behaviour.

After the graph the application displays a description of the personality factor that was the

Screen F: About information

The user was presented with an explanation of his/her highest personality factor and related possible security threats (screen E) correlating with the personality trait. This was determined by the survey in this particular study, as well as from results from other surveys found in relevant literature. Possible guidelines or tips to counteract the weaknesses were then offered to the user. The About screen (F) explained the aim of the app.

Feedback obtained from users on using the app is presented in the next section.

5 Interpretation and feedback

5.1 MSA - Mobile and security awareness app evaluation

An evaluation of the usefulness of the mobile application that was developed was conducted in the form of an electronic survey in which questions were posed to app users as set out in table 5. An analysis and interpretation were carried out for each question. An example of a Likert-scale question with the results is shown in Figure 3: "The application is user friendly".

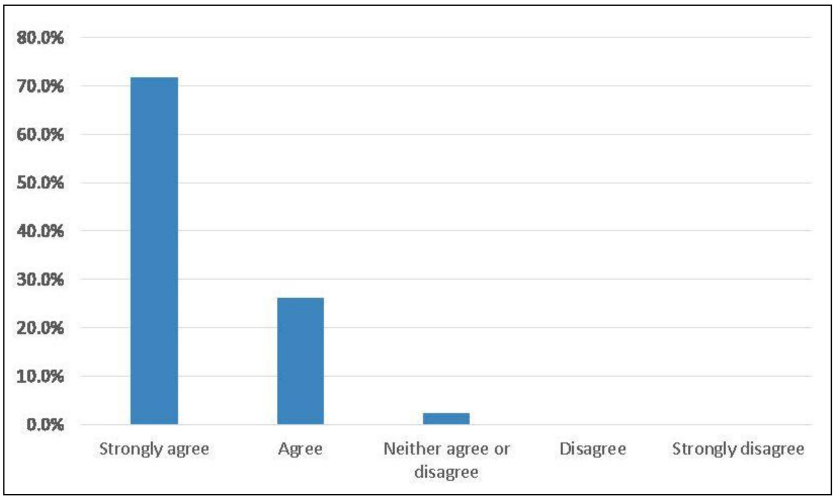


Fig. 3. The application is user friendly [6]

Most respondents agreed that the app was user friendly. The intention of the app was to improve users' awareness and raise knowledge of mobile device security. The users were asked questions to gain feedback on the app concerning whether this was indeed the case. Figure 4 indicates the responses for two questions as example: (1) I learned more about mobile security whilst using this app and (2) I became more aware of mobile device security whilst using this app.

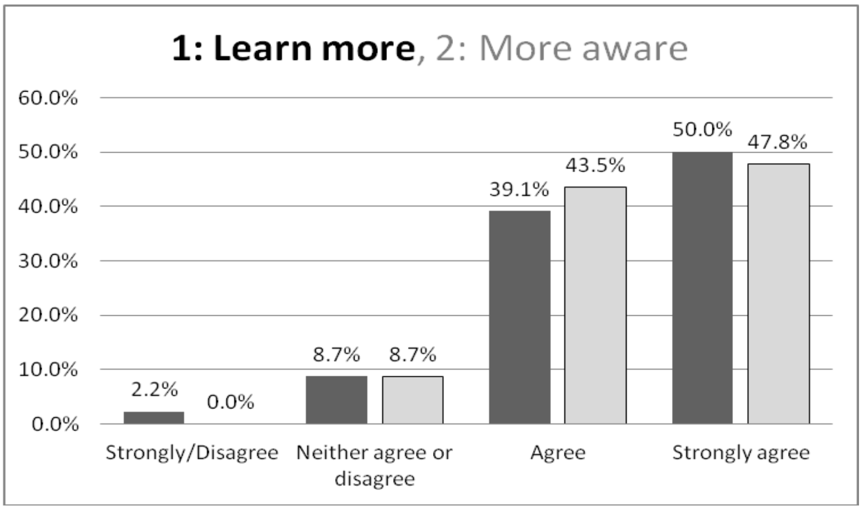


Fig. 4. I learned more (1) and became more aware (2) of mobile device security while using this app

This app succeeded in this in that 89.1% of users agreed or strongly agreed with the statement that they have learned more about mobile device security and in addition 91.3% agreed or strongly agreed that they have become more aware of mobile device security when using the app.

Other issues in the evaluation of the app include for example: Ease of use and navigation, motivation to keep on using the app, learn more about mobile security awareness and meeting expectations. There were also a few open questions where comments could be given. Positive aspects of the application were [6]:

- "*The application is easy to use and user friendly,*
- *Informative information is given;*
- *There are fun facts included;*
- *It improves common sense and knowledge of security;*
- *It states the basic but most important information*".

Feedback also highlighted that mobile users including those with some background in IT are too trusting concerning security issues.

Negative aspects of the application that were given were:

- With additional work the app could be expanded in future to include more topics;
- Some of the answers to questions in the quiz were direct definitions, which made the alternative answers obvious.

5.2 PSB: personality and security behaviour app evaluation

The feedback from users of this app was minimal due to time restrictions of the project. During an artefact day where the app was demonstrated verbal feedback from respondents was recorded as follows:

- *"The personality test was quite long*"– 50 questions. However each user was eager to see his/her personality factor score;
- *"The app was easy to use and navigate*";
- Considering that psychology issues are not usually part of IT users' context this was an interesting app for users to experience and to notice subsequently where the shortcomings may reside regarding their safe or unsafe behaviour concerning information security and use of information resources;
- *"Information security tips provided by the app were helpful".*

These comments and feedback can be taken into consideration in future applications when information security issues are addressed in apps as educational tools.

6 Conclusions

Two mobile applications were developed as second phases in two separate studies concerning (1) information security awareness and mobile device users and (2) personality and information security behaviour of computer users. This paper focused mainly on the development of these apps. For the first study (MSA), a mobile application was developed to address the shortcomings regarding the awareness of users concerning mobile device security. The aim of the application was to make users more aware and to improve their knowledge regarding mobile device security. The application presented results to the app users from the first phase (survey) indicating the knowledge levels, attitude and behaviour of students regarding information security awareness. There was also a quiz were app users could test their knowledge. A question stated to the participants whether this survey has made them more aware of information security issues were answered mostly in a positive way. The application thus achieved its goal as most of the feedback through an evaluation phase from users was positive regarding the application's ease of use, functionality, relevance to the topic and advancing awareness and knowledge of users regarding this topic.

The second app (PSB) had as aim to assess the users' highest personality traits in terms of the five-factor model and to inform them of possible threats in their computer or online actions. It also presented tips or guidelines to improve their security behaviour focusing on the specific highest personality score of a user.

It is observed from these two examples that the concept to use mobile applications can be valuable when educating users regarding information security and their awareness and safe cyber behaviour. Results from the surveys were presented through the apps to the users. By including the quiz or test, knowledge could be assessed and feedback on their answers was provided to the users. Mobile apps can also be used to assess the users' personality traits and their level of secure behaviour and subsequently provide techniques to app users to enhance their information security awareness.

For future work the apps can be adapted to focus on users of different age groups (these two apps focused mainly on young adults) and the apps can be released on other mobile OS platforms. Content of these apps can be modified as the information security threats change over time. Gaming aspects can also be included in the design of the mobile applications.

References

1. Holzer, A., & Ondrus, J. Trends in mobile application development. International Conference on Mobile Wireless Middleware, Operating Systems, and Applications (pp. 55-64). Springer Berlin Heidelberg (2009).
2. Chickowski, E. Biggest Attacks Of 2016 (So Far) http://www.darkreading.com/cloud/biggest-attacks-of-2016-(so-far)/d/d-id/1325865. .
3. Peng, S., Yu, S. & Yang, A. Smartphone malware and its propagation modeling: A survey. IEEE Communications Surveys and Tutorials, 16(2):925-941 (2014).
4. Lawton, G. Is It Finally Time to Worry about Mobile Malware? Computer, 41(5):12-14 (2008).
5. Ben-Asher, N., Kirschnick, N., Sieger, H., Meyer, J. Asaf Ben-Oved, A., & Möller, S. On the need for different security methods on mobile phones. In Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11). ACM, New York, NY, USA, 465-473. DOI=http://dx.doi.org/10.1145/2037373.2037442 (2011).
6. Park, M. Mobile device security: Young people’s awareness and perceptions. Masters dissertation. NWU. South Africa (2016).
7. Le Grange, A. The correlation between the personality and information security behavior of a computer user. BSC Honors project report. NWU, South Africa (2016).
8. Pfleeger, C.P., Pfleeger, S.L., & Margulies, J. Security in Computing (5th Edition). 5th ed. Prentice Hall Press: Upper Saddle River, NJ, USA (2015).
9. Whitman, M.E. & Mattord, H.J. Principles of Information Security 5th Edition. Cengage Learning. Boston. USA (2016).
10. Schneier, B. Secrets and Lies: Digital Security in a Networked World. 1st edition ed. Wiley (2004).
11. Kortjan, N., & Von Solms, R. A conceptual framework for cyber-security awareness and education in SA. South African Computer Journal, 52(1), pp.29-41 (2014).
12. Walaza, M., Loock, M. & Kritzinger, E. A pragmatic approach towards the integration of ICT security awareness into the South African education system. In Information Security and Cyber Forensics (InfoSec), 2015 Second International Conference on (pp. 35-40). IEEE (2015).
13. Cone, B.D., Irvine, C.E., Thompson, M.F. & Nguyen, T.D. A video game for cyber security training and awareness. Computers & Security, 26(1), pp.63-72 (2007).
14. Kruger, H.A., & Kearney, W.D. A prototype for assessing information security awareness. Computers & Security, 25:289-296 (2006).
15. Kruger H.A., Drevin L., & Steyn T. A Framework for Evaluating ICT Security Awareness. In: Proceedings of the ISSA 2006 Conference, ISBN 1-86854-636-5(2006).
16. Shropshire, J., Warkentin, M., Johnston, A., & Schmidt, M. Personality and IT security: An application of the five-factor model. AMCIS 2006 Proceedings. pp. 3443-3449 (2006).
17. McCrae, R.R. & John, O.P. An introduction to the five-factor model and its applications. Journal of personality. 60(2):175–215 (1992).
18. John, O.P., Robins, R.W., & Pervin, L.A. Handbook of Personality, Third Edition: Theory and Research. Guilford Press: London (2008).
19. McBride, M., Carter, L., & Warkentin, M. Exploring the Role of Individual Employee Characteristics and Personality on Employee Compliance with Cybersecurity Policies. Technical Report, RTI International (2012).
20. Flora, H.K., & Chande, V. A review and analysis on mobile application development process using agile methodologies. International Journal of Research in Computer Science, 3(2):9-18 (2013).
21. Ophoff, J., & Robinson, M. Exploring end-user smartphone security awareness within a South African context. In Information Security for South Africa (ISSA), 2014 (pp. 1-7). IEEE (2014).
22. Mann, M.I. Hacking the Human: Social Engineering Techniques and Security Countermeasures. Gower Publishing, Ltd.: Burlington. p267 (2012).
23. McBride, M., Carter, L., & Warkentin, M. Exploring the Role of Individual Employee Characteristics and Personality on Employee Compliance with Cybersecurity Policies. Technical Report, RTI International (2012).
24. Argawal, H. Top 10 Mobile Phones Operating Systems, https://www.shoutmeloud.com/top-mobile-os-overview.html, last accessed 2017/01/22.

Graduate attributes for Computer Science in Libya

Hamzh Alaiat and Isabella M. Venter

University of the Western Cape, Cape Town, South Africa
Alkanoni2010@gmail.com

Abstract. We stand on the brink of the fourth industrial revolution, the third being the digital revolution that started in the 20th century. This technological revolution will fundamentally change the way people live and work. To be prepared for this fast-changing technological reality, computer science curricula should be continuously adapted and tailored to prepare students for the reality of life after completing their degrees. In this study, the graduate attributes as suggested by the Association for Computing Machinery were considered in terms of the experience of graduates of higher technical and vocational institutions in Libya. This was achieved by evaluating the perceived graduate attributes that computer science graduates from Libyan institutions gained against those characteristics recommended by the Association for Computing Machinery; computer science curriculum 2013. Design science research was used as the research methodology and both qualitative and quantitative methods were used to explore the attributes that computer science graduates believe they gained from studying at Libyan institutions. This study found that in many cases graduates felt that they were not sufficiently prepared for the workplace as they did not possess the necessary graduate attributes. The results of this study will form the basis of a subsequent study to create a framework which could be used to keep computer science curricula relevant. The framework could also be used to advise how the Libyan curricula should be adapted to ensure that students that graduate from Libyan universities are globally competitive.

Keywords: Information technology; Libya; higher technical and vocational institutions; computer skills; computer science; curriculum; developing country; design science research, graduate attributes.

1 Introduction

Advancement in technology in the last 50 years has made life and work without technological devices or systems almost impossible to imagine. Technology has not only transformed the way in which business is conducted but also the manner in which pedagogy is practiced. Libya, even though it is a rich country with its oil and gas industry accounting for approximately 70% of its Gross Domestic Product (GDP), is regrettably not at the forefront of technological development [1]. Why would this be so? It can probably be ascribed to a lack of infrastructure and the recent unrest in the country.

It is important to address this skills deficiency in Libya by revising or improving the Computer Science (CS) curriculum to keep abreast of the new global technological landscape [2]. A revised curriculum's initial focus should be on the graduate attributes required to equip students with skills that will enable them to participate fully in society and compete globally —the topic of this paper. CS graduates with skills as defined by the Association for Computing Machinery (ACM) will undoubtedly contribute to the advancement of the Libyan nation's social and economic growth [3].

Many institutions in Libya offer CS as a subject, however, although the curricula and study plans are standard and supplied by the government, they are adapted by the lecturers and often lack sufficient flexibility in implementation and application to meet the needs of the labour market and keep pace with technological development [4]. Currently, Libya has nearly 382 post school institutions which offer technical education at various levels providing education to more than 160 000 students across several disciplines [4]. Bachelor degrees can be taken by students who have already obtained their secondary school certificate [5]. Advanced studies are offered after the completion of a university or higher vocational education, and these include postgraduate diplomas, Master's Degrees and PhD Degrees in various specializations (see Table 1) [6].

Table 1. Education system in Libya

	Level		Years	Academic Qualifications	
School	Primary school		6	Medium Diploma	
	Primary secondary		3		
Secondary School Qualifications	Specialized secondary schools (4 years)	Vocational qualification	3		
			1	School leaving certificate	
Higher Education Undergraduate	University	High technical and vocational institution	3	Bachelor Degree (4 years)	Vocational Diploma (3 years)
		Specific higher institution	1		Convert Diploma to degree
Advanced Studies Postgraduate	Masters degrees		2	MSc	
	PhDs degrees		3	PhD	

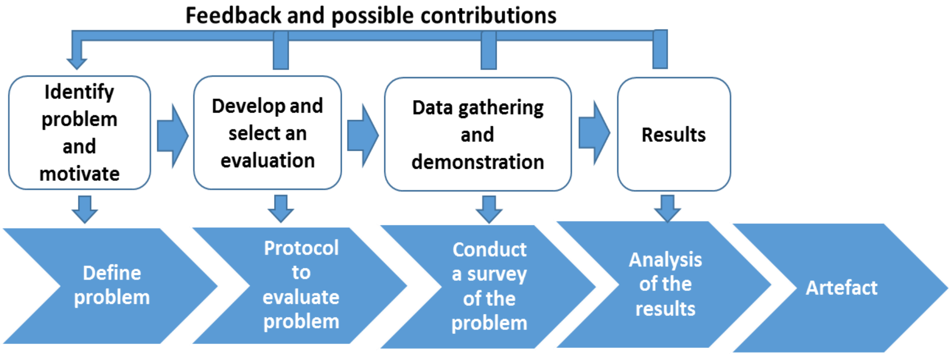
According to Sztendur the use of information technology (IT) among the general public in Libya is minimal, and this applies to the education sector as well. Many students and even some educators in Libyan schools are either computer illiterate or have very little experience in computer usage. The level of basic computer related knowledge is even low in Higher Technical and Vocational Institutions (HTVI) other than in the Computer Science Departments [7]. The question thus is: How should the Libyan Higher Technical and Vocational Computer Science curriculum transform to be aligned with international trends?

According to Impes et al., curricula and study plans of many Libyan institutions are lacking the necessary content to be relevant in this fast changing paradigm [4]. Employers require that graduates have skills such as: good communication skills; problem-solving skills; critical and analytical thinking; ability to work in a team; willingness to learn; good interpersonal skills; ability to work in a modern organization and; graduates who are knowledgeable. These so-called graduate attributes can be developed in CS graduates if the CS curriculum makes provision for them [8]. In this research the aim was to explore the graduate attributes acquired by CS graduates at Libyan institutions and check whether they align with those associated with CS curricula based on the ACM CS curriculum.

2 Materials and Methods

Using a predesigned questionnaire, data was collected from ten employees employed at different companies and industries in Libya as an exploratory study. Semi-structured interviews were subsequently conducted with five of these HTVI graduates.

Fig. 1. Design Science Research framework based on the methodology as proposed by Halinen [9]


Feedback and possible contributions
Identify problem and motivate
Develop and select an evaluation
Data gathering and demonstration
Results
Define problem
Protocol to evaluate problem
Conduct a survey of the problem
Analysis of the results
Artefact

Based on the information gained from the exploratory study a questionnaire was created using Google Forms and was electronically distributed to fifty-two graduates from Libyan institutions currently employed at various Libyan companies. The questionnaire was designed based on the eleven characteristics of a Computer Science graduate as defined by the ACM Curricula 2013 [8].

These ACM defined characteristics include:

1. Technical understanding of CS: Graduates ought to have a mastery of CS as stated by the fundamental body of knowledge.
2. Appreciation of interaction between theory and practice i.e. how theory and practice impact each other.
3. Problem-solving skills i.e. ability for a graduate to apply knowledge gained to solved real world problems.
4. System-level perspective i.e. graduates ought to think at multiple levels of detail and abstraction.
5. Familiarity with common themes and principles. Graduates should understand common themes in CS such as concurrency, complexity, abstraction, evolutionary change, sharing of common resources and security.
6. Project experience i.e. ability for graduates to apply knowledge gained is manifested if the graduate has been involved in at least one substantial project.
7. Commitment to life-long learning. Graduates should have a solid background in CS that allows them to maintain and acquire relevant skills as the filed evolves.
8. Commitment to professional responsibility. Graduates should know the interplay of technical problems, ethical issues and legal standards when developing computing systems.
9. Organization and communication skills. Graduates should be able to effectively communicate and present technical problems and solutions to an audience.
10. Awareness of the wide application of computing.
11. Awareness and appreciation of domain-specific knowledge. Hence graduates must be able to interact with experts from different fields and domains in the course of their careers.

Permission to conduct the initial survey and interviews as well as the subsequent survey was granted by the University of the Western Cape Ethical Committee (UWC registration no. 15/7/238). The aim of the second survey (the on-line questionnaire) was to gauge what the respondents' perceptions are of the graduate attributes they acquired during their studies. Each of the 52 respondents was contacted via e-mail to enquire if they would agree to be participants in the survey.

An adapted from of Design Science Research (DSR) was used as a methodology for the research since it enables a combination of methods, underpinned by theory, to be used during data collection and analysis [10]. *However, in this paper, only the first four steps of the DSR methodology are dealt with and these are the focus of this paper.*

Design science is a paradigm that deals with the construction of an artefact as opposed to behaviour science, which deals with the behaviour of humans as they use artefacts [11].

The methodology can be described as five iterative steps (see Figure 1):

1. Identify problem: During this phase a problem is discovered or identified. The output of this phase is a research proposal. The problem identified in this instance is: How does the Libyan Higher Technical and Vocational CS curriculum differ from that recommended by the ACM CS curriculum?

2. Develop and select an evaluation method for the problem: Here a protocol is identified that will aid in evaluating the identified problem. In this instance, the similarity of the Libyan Higher Technical and Vocational CS curriculum to the ACM CS curriculum will be evaluated by analysing feedback from surveys on professions in the CS and related industries.
3. Data gathering: In this phase, professionals in the CS and related fields were interviewed and tasked with filling in questionnaires that were developed to aid in assessing the CS curriculum in Higher Technical and Vocational institutions in relation to market demands.
4. Results: The results of the analysed interviews and questionnaires were produced to act as a guide in assessing the problem stated in step 1 above.
5. Artefact development: The results of the four steps were used to develop an artefact that will aid in addressing the differences that exist between the CS curriculum in Libyan Higher Technical and Vocational institutions and those of the ACM international standards.

The artefact development has not been covered in this paper and will form the basis of future related research to be conducted by the corresponding author.

3 Results

3.1 Survey Results

The first section of the on-line survey collected descriptive statistics about the backgrounds of the 52 respondents. Eighteen of the respondents qualified at universities in Tripoli, eight studied at institutions located in the city of Zawiya, seven in Tarhona, six in Bani Waled, four in Misrata, and three in Benghazi. Five were from institutions located in other towns. Thirty-four respondents received tuition in both Arabic and English, fifteen did it in Arabic alone and the rest received it exclusively in English. Twenty-seven respondents worked as technical computer engineers, eighteen worked as associate computer engineers, one was a senior computer engineer and the rest did not specify their profession. Seventeen of the respondents hold higher diploma qualification; fifteen have bachelor’s degrees; fourteen have both a higher diploma and bachelor degree; two have a bachelor and master’s degree, as is the case for those who hold a higher diploma, bachelor and master’s degree. Two specified that they have a master’s degree but did not specify what other qualification they have (Fig. 2).

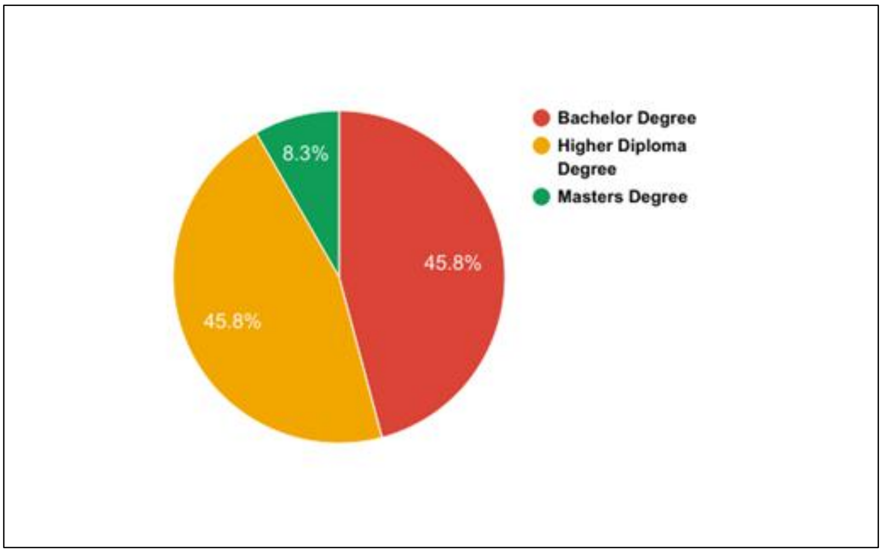


Fig. 2. Qualifications earned whilst at the higher institution.

Most of the respondents are still new in their profession with 22 being in their third year of work since their qualification. 20 have worked for two years, seven have five or more years working whereas three are in their first year of work.

The rest of the questionnaire focused on the graduate attributes as is suggested by the ACM. In a question that deals with how they understand certain concepts, only nine understood the concept of Abstraction; sixteen understood the concepts of Complexity and Security; seventeen understood the concept of Evolutionary Change; twenty-seven understood the concept of Resource Sharing and; only eleven understood Concurrency.

In response to the question "Do you feel that the theory that you have learnt during your studies prepared you for the practical requirements of your work-place?" twenty five rated it as 2 and sixteen rated it as 1 on the Likert scale [12] where 1 means "Strongly disagree" and 5 "Strongly agree", indicating that their studies did not prepare them well enough for the workplace (see **Fig. 3**).

When asked whether their qualification prepared them to think at multiple levels of abstraction and detail, 76% felt it did not. Similarly, 65% of the respondents felt that their communication skills in explaining computer solutions to others were poor. Only 21% of the respondents indicated that they were involved in at least one substantial programming project during their studies. Even though only 40% of the respondents indicated that they read books or articles about computing, 62% felt that they are up to date with the latest developments in computing. Most (82%) of the respondents stated that they had neither attended nor participated in any recent computing conference, competition or course with only 11% having given a presentation on technical problems and their solutions to a larger audience. Additionally, 46% of the respondents stated that they have personal web-pages or blogs. Just fewer than 10% of the respondents have used high performance computer cluster at their workplace or have attended a course that dealt with the social, legal or ethical aspect of the discipline of computing; with only 17% having ever consulted an expert in a domain outside their personal area of expertise.

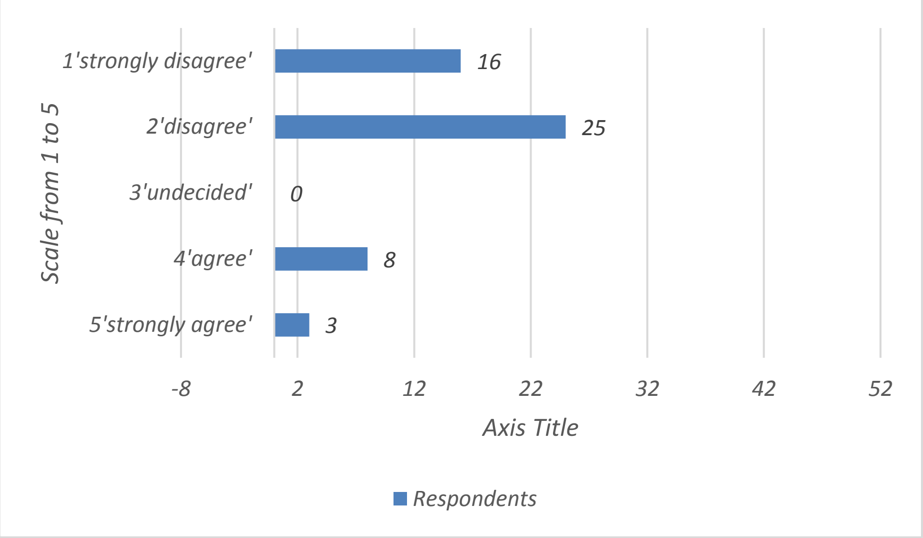


Fig. 3. School-work theory effect on practical requirement at work

3.2 Interview Results

All participants (five) interviewed during the exploratory study were male. The interviews were conducted via Skype and in both English and Arabic. It seems as if some of the questions were misunderstood – probably due to the language and its nuances.

Three of the participants indicated that they chose their careers because it is related to their field of interest while the rest said that they find their careers interesting. Most of the participants indicated that they are responsible for the installation of software and maintaining the company's networks. When asked what type of degree an employee in their company should have to be successful, two indicated that a Bachelor's degree is enough whereas another two indicated that a Master's degree would be better. Only one of them answered that a higher diploma in software engineering would be optimal for his company.

When asked what skills they acquired at university which they felt stood them in good stead in their positions, most of them answered that it was basic software programming skills such as the ability to use web applications, java script and multi-media. In addition one added that English is very important, whereas another added that solving mathematical problems is. All of them indicated that they acquired communication skills (when working with a team) at their place of employment. Four added that they acquired the skill to solve networking problems and five that they are now able to deal with different operating systems such as iOS, Linux and UNIX. However when asked which courses should be added to the curricula of their institutions most felt that they would have liked to learn more programming languages and to have been introduced to a variety of operating systems.

Most of the participants, when asked how they think their college/university education could address these skills more effectively, said that the universities should focus on network problem solving skills and Graphical User Interface courses that would allow them to design websites. Two felt that they need more training in the maintenance of hardware and one that the university should focus on web programming skills. In addition, most of them felt that they needed technical skills more than theory. All of the participants agreed that the university CS curricula should incorporate technological changes to keep up to date.

When asked what graduate attributes had contributed most to their accomplishments in their current company, one participant said that his ability to solve software problems and to work with a team effectively. Most of the participants were members of a team for some aspect of their work and felt that it was an important graduate skill to be able to contribute to the team effort. Some felt that in order to deal with conflict (in the team) it is important to be patient and be prepared to listen and learn. Another felt that if they could not deal with the conflict they would ask their employer to intervene. When asked how they handled situations where their communication failed them, some indicated that they revised their opinion and tried to be more assertive by calling a meeting to solve the misunderstanding.

Most of the respondents felt that the government and the ministry of higher education are responsible for curriculum development but should keep international standards in mind. Staff at tertiary institutions should preferably be involved with curriculum development in order to keep it relevant. This they observed would be easier to accomplish if the universities fostered partnerships with other institutions of higher learning in other countries across the continent and more so in those located in developed countries. Teaching staff members should furthermore belong to research units that are affiliated to international bodies. It would make it easier for them to keep up to date with revised curricula. Industry could contribute to the development of student skills by funding internships and forming partnerships with the institutions.

4 Discussion and Conclusion

As inhabitants of the "*global village*", IT graduates of Libya should acquire skills that make them competitive globally. Rai et al. are of the opinion that African graduates should be equipped with the skills that will allow them to "*tackle challenges faced by their communities*" [13]. These skills are, according to Ogbor, skills which in "*Sub-Saharan Africa requires of them (much more than their counterparts in the developed world) to be able to build their own IT businesses—small and medium enterprises which have proven to contribute significantly to economic growth in this region*" [14]

To corroborate this, the graduates of higher institutions in Libya who completed the questionnaires and were interviewed, felt that they had not acquired the necessary graduate skills to function optimally in their places of employment. Most observed that more focus was put on the theoretical aspect of CS in school than practical work even though the opposite seemed to be what was needed at the workplace. Similar observations have been made in studies by Rai *et al*. [13] and Kroeze *et al.* [15]. Rai *et al*. in their study have found that [13] industrialists need graduates who: exhibited the ability to work independently; have problem-solving skills and; possess good communication skills. Similarly Kroeze *et al*. reports that [15], employers require graduates who can learn adaptively; are effective communicators; can work in a team; are critical thinkers and; ones who understand the fundamental concepts of the CS discipline.

Most of the interviewees felt that they ought to have been exposed to, inter alia, the ethics required for professionals in their field. It is evident that skills such as abstract reasoning, team work and communication skills seemed to have been neglected in the CS curriculum at tertiary institutions in Libya and in other countries in sub-Sahara Africa [13, 15]. Most admitted to struggling to properly articulate their ideas and contributions to fellow employees. These important skills were only nurtured and developed when they were already working.

According to Triki [16] the mission of universities and higher technical institutions should be to prepare students and train mature workers according to what the industry demands. However this may be a very short sighted view, surely universities should rather focus on preparing students who can function in a rapidly changing world, thus they should focus on graduate attributes as well as technical expertise.

5 Future work

Future work will entail creating a framework (artefact) that will allow lecturers to keep track of how their curriculum aligns with the curriculum as advised by the ACM. It will also act as a guide when trying to adjust the Libyan Higher Technical and Vocational Computer Science curriculum to be specific to both the Libyan and sub-Saharan contexts while still conforming to the set standards of CS curriculum as determined by the ACM.

References

1. N. Gelvanovska, M. Rogy and C. M. Rossotto, "Broadband Networks in the Middle East and North Africa Carlo Maria Rossotto," Washington, 2014.
2. D. S. Abuhadra and T. T. Ajaali, "Labour market and employment policy in Libya," 2014.
3. J. Cackler, E. Gu and M. Rodgers, "Technology in Developing Economies," 2008. [Online]. Available: http://cs.stanford.edu/people/eroberts/cs181/projects/developing-economies/. [Accessed 16 May 2015].
4. A. Impes, A.Wilson, D. Othman and C. Pislaru, "Analysis of Communication Barriers in Technical Vocational Education System in Libya," in International Conference on Communication, Media, Technology and Design, Istanbul, 2014.
5. S. Othman, O. Kashadah, A. A.-G. Gosbi and S. Khoja, "The higher education system in Libya," 2012. [Online]. Available: http://eacea.ec.europa.eu/tempus/participating_countries/overview/libya_overview_of_hes_final.pdf. [Accessed 19 July 2015].
6. Tamtam, F. Gallagher, A. G. Olabi and S. Naher, "Higher education in Libya, system under stress," Procedia - Social and Behavioral Sciences, vol. 29, p. 742–751, 17 December 2011.
7. R. I. M. Ewa M. Sztendur, "Attitudes towards e-Learning and Satisfaction with Technology among Engineering Students and Instructors in Libya," Australia, 2013.
8. S. Roach and M. Sahami, "CS2013: Computer science curricula 2013," 1 4 2013. [Online]. Available: https://www.acm.org/education/CS2013-final-report.pdf.
9. R. Halinen, "ICT- Design Science Research," Lahti,Finland, 2012.
10. V. Vaishanavi and W. Kuechler, "Design Science Reserch in Information Systems," 20 January 2004.
11. S. T. March and G. F. Smith, "Design and natural science research on information technology," Decision Support Systems, vol. 15, no. 4, pp. 251-266, 1995.
12. J. Gliem and G. Rosemary, "Calculating, interpreting, and reporting cronbach's alpha reliability coefficient for likert-type scales," in Midwest Research-to-Practice conference in adult, continuing and community education, 2003.
13. A. Rai, A. Rodrigues, I. M. Venter, G. Mills, H. Suleiman and J. Edumadze, "Relevant Computing Curricula in Sub-Saharan Africa," in 3rd International ICST Conference on e-Infrastructure and e-Services for Developing Countries, Stone Town, Zanzibar, 2011.
14. J. O. Ogbor, Entrepreneurship in Sub-Saharan Africa: A strategic management perspective, Bloomington: Authorhouse, 2009.
15. J.H. Kroeze, S.R. Ponelis, I.M. Venter, P.D. Pretorius and P. Prinsloo, "Aligning African computing disciplines' graduate attributes with international standards," in International multi-conference on society, cybernetics and informatics, 2012.
16. N. Triki, "Higher technical and vocational education and training programmes and its impact on the Libyan manufacturing industry," Literacy information and computer education, vol. 2, no. 2, 2013.

The new generation of students' ICT needs and expectations: Case of a developing country

Sibusisiwe Dube

University of Cape Town
Private Bag X3, Rondebosch 7701, South Africa
sibusisiwenkonkoni@gmail.com

Abstract. In this study, the aim was to identify, describe and explain the information and communication technologies (ICT) that students from developing countries find useful in their education. The cohort of forty students was drawn from one of the Zimbabwean state universities. Zimbabwe is a developing nation that boasts high literacy rates, despite economic challenges being faced. However, the country still has minimal ICT-enabled education exposure, hence the limited research on this phenomenon. This descriptive and explanatory research was based on the qualitative data collected from the final-year Computer Science undergraduate students with regard to their seniority and their ICT skills. The cohort of students was organised into ten focus groups of six students each with the aim of identifying the ICT learning preferences that they deemed useful for satisfying their educational needs regarding research, assessment and examinations. From the findings it emerged that the ICTs most popular with this cohort of students include Google Classrooms, Google Documents and YouTube video tutorials. The students indicated that they prefer to use these technologies as they enhanced their problem-solving skills, which translated to their learning effectiveness. These findings fill the existing gap in the body of knowledge relating to the current generation of students from the developing nations' learning ICT preferences and expectations.

Keywords: Current generation, Teaching and learning ICT preferences, Developing countries

1 Introduction

There has been an increased demand for the restructuring of education to take advantage of information and communication technologies (ICT). These demands are evident in the development of a millennium development goal (MDG) with the focus on ICT-based education. The call has also been necessitated by the current generation of students who are technologically savvy and adept in ICTs. Their lives revolve around ICTs; technologies with which they have been brought up. ICTs have a causal power to positively influence students' learning and thereby affording them many teaching and learning benefits. The advantages include enhancing and improving the learning process [1, 14]. For example, Learning Management Systems, such as Sakai or Moodle are a cheaper ICT-based learning method. They provide a learning environment that is independent of space and time, facilitating learning anywhere and at any time. ICTs also increase access to learning content that is consistent dependable, has a potential for re-use, as well as supporting convenient student-centered learning [17]. It is therefore clear that ICTs play an important role in the teaching and learning of this current generation of students, who depend heavily on ICTs for their social interactions [15]. This heavy dependence on ICTs has thus earned the current generation of students such names as "Net Generation,", "Millennials and "digital natives".

It is not surprising then to note that the generation of students currently attending higher education prefers ICT-based teaching and learning strategies and therefore take advantage of the proliferation of the small, high-powered and ubiquitous ICT devices for learning purposes. Hope [7] asserts that these are techniques that work best for this cohort of students [10]. Contrary to their predecessors, the current generation of students can use and absorb vast quantities of information quickly. Teaching and learning methods without ICTs tend to frustrate them. They get bored and lose interest when subjected to traditional teaching methods that are without ICTs. They learn better by engaging ICTs that enable them to work as a team, and to collaborate on assignments. ICTs provide these students with exciting, and experiential learning activities, hence an observation by Black [3] that the current generation of students is accustomed to using ICT-based video games and text messages, activities that tend to

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

reduce their patience for traditional, step-by-step instruction, lectures or assessment. They are, therefore, dissatisfied with these teaching and learning strategies, pressuring the learning institutions to divert from the traditional chalk and talk to on-line education based on virtual classrooms. However, few studies have documented this generation's specific ICT needs, a problem discussed in the next section.

2 Problem Statement

Despite the evidence in literature that the current generation of students expects ICT tools to be integrated in the education process, the higher education institutions (HEI) currently seem to fail them. The HEIs in the developing country environments are currently characterized by a slow ICT-adoption process. In a study by Faqih [4], ICT resources are still used sparingly for the teaching and learning practice in these contexts. Existing studies indicate that ICTs are not yet fully utilized in the education of the current generation of students. Where ICTs have been adopted, studies show that their use does not continue as expected, which could be attributed to various structural and cultural mechanisms that exist within the HEIs [18]. A disconnection exists between ICTs and the teaching methods currently used where ICTs are only used as information capture and storage tools for uploading notes and assignments. This scenario tends to convert the technologically savvy students into passive consumers of the learning content rather than to encourage them to become active co-producers of learning content and new knowledge. The current generation of students is therefore deprived of the opportunity to become problem solvers, hence the persisting need to transform education accordingly. There is, however, limited literature discussing ICTs that have been identified as appropriate for this cohort of students' learning process, since the existing studies have been criticized for focusing more on technology per se, and having little regard for learners' readiness to learn through specific ICT systems [6].

Lack of knowledge relating to students' ICT choices and preferences hinders education progress, hence the argument by Black [3] that the people who are tasked with educating the Millennials fail them by not meeting their technological expectations. In agreement, Jones and Ramanau [9] posit that the changes in technology have left the learning institutions with no option, but to comply with this generation of students' demand to move from teacher-centered to learner-centered teaching and learning approaches. Being the sole customers of HEIs, it is of paramount importance for HEIs in the developing world to rethink how they operate and teach to the satisfaction of this current generation of students and thereby prevent the demise of higher education. Education institutions are further advised to engage the current generation of students in a dialogue regarding their expectations about ICTs for teaching and learning if returns from technological investments are to be realized [16]. Based on the problem discussed above, the questions answered by this research are discussed in greater detail in the next section.

2.1 Research questions

The questions this study sought to answer are as follows:

a. Which choices of emerging ICTs satisfy the educational needs of the generation of students currently enrolled in the HEIs of developing countries?

b. Which students' recommendations need to be considered for the successful and innovative use of ICTs in the context of education in the developing world?

The answers to these questions will make a descriptive and explanatory contribution to the body of knowledge regarding ICT-enabled education in the HEIs of developing countries. The results are important to the ICT-enabled education researchers who are concerned about the persisting digital-divide problem in the developing world. The findings are empirically necessary for the organization of higher education environments in accordance with the requirements and expectations of the currently discontented students. The aim and objectives of this study are elaborated upon in the following section.

2.2 Research aim and objectives

The study aimed at investigating the current generation of students' ICT requirements and expectations in compliance with one of the eight MDGs. The objective was to identify appropriate ICTs suitable for the teaching and learning of the current generation of students in developing countries. The purpose of the study was also to inform and guide future choices and implementation of education ICTs in a way that promotes effective learning by addressing the students' ICT assumptions and preconceived ideas. The relevance of the findings is in that an understanding of the ICT requirements will result in an effective use of ICTs that stresses an adherence to learning strategies favoring the current generation of students' ICT-enabled learning needs.

Nevertheless, as shown in the next section on related work, there is limited research specific to the technological needs of the current generation of students in the context of the developing world. There is also little documentation of how the teaching and learning can adapt to this cohort of students' ICT needs. The existing research mostly focused on the factors affecting the integration of ICTs, as well as on the ICT users' competencies and ignored the students' actual requirements and expectations. In the subsequent section, the related work, the inadequacy of which has motivated and underpins this study, is discussed.

3 Related work

Several studies exist that discuss the ICT-based education phenomenon. However, few studies explicitly investigate the developing countries' current generation of students' ICT choices and preferences. On the contrary, most existing studies relate to the developed countries' students' choices and expectations. The few studies that have attempted to discuss this phenomenon from the developing countries' view have been criticised for the bias towards issues other than the current generation of students' ICT desires. For example, Matti and Minna [13] investigated ICT challenges faced by Tanzanian students, as well as the issues they regard to be the most important for their learning. Their results indicate that in addition to the pedagogical approach and course-content concerns, the students related that their study environment where the ICTs are being implemented also requires consideration for the contextualization of ICT in education. This proves that the current generation of students favors the use of ICTs in learning, which they consider to be helpful resources that enhance their understanding of the course content.

These students are different from their predecessors who have been affectionately termed the "digital immigrants" in that while the former find comfort in ICTs, the latter have had limited ICT exposure. Another study on the current generation of students revealed that they have a positive attitude towards ICTs and can comfortably use ICTs for learning since they have grown up in a world surrounded by computing devices, which support multitasking [2]. The current generation of students continuously communicates learning material through such ICTs as instant messages, social media, phone calls and emails, and acquires readily available information on the Internet via the emerging ICT technologies.

A related study was also conducted to identify strategies of enabling learners in developing countries to fully exploit the potential of learning management systems (LMSs) [5]. The findings indicate that the LMS services mostly desired and mostly accessed by the students include assignments, announcements, resources, course outlines and the chat room. Mobile devices were not so popular with these students and were thus rated the least used devices for accessing the LMS services. The above discussion is indicative of how few studies discuss the developing context of students' ICT preferences. It is on this premise that this research employed the methodology to collect data relating to the current generation of students' ICT choices and preferences appropriate for their learning, which is discussed in the next section.

4 Research Method

Qualitative data were collected from a single university in Zimbabwe, a developing country in Southern Africa. Zimbabwe is a country with ICT-enabled education strategies different from those of her counterparts in both the developing and the developed countries that are fully industrialized [8, 13]. Unlike her counterparts, Zimbabwe still depends on the basic ICTs in the form of PowerPoint slides and overhead projectors due to economic challenges. The university has a science and technology background whose inception was informed by the need to promote the science, technology, engineering and mathematics (STEM) initiative. The chosen institution is contributing to the highest adult literacy rates Zimbabwe is currently boasting of regionally and is one of the few HEIs that is making recognizable strides towards integrating ICTs in teaching and learning. Nevertheless, a top down approach in both the choices and implementation of ICTs is followed. There is little or no engagement of the students, the intended ICT beneficiaries, in the choosing and implementation process of ICTs for their leaning, hence this study.

Following the pilot study survey on the level of ICT integration in their studies, forty students volunteered to participate in this study. These students were organised into seven focus groups, each with six students, mostly males, due to the gender imbalance in the program. The focus group's audio-recorded discussions lasted between fifteen and twenty minutes. The theme of the discussion included the evaluation of the institutional ICT-based teaching and learning methods, as well as the existing institutional ICTs. The fact that these students were in the final stage of their undergraduate studies enabled them to relate their lived ICT-enabled learning experiences in comparison with their requirements and expectations. This focus group's discussion approach enabled the researcher to elicit information from the real-life education setting based on the personal and lived ICT experiences of the students.

Although based on a cross-sectional and not a longitudinal study, the cohort of students had had a four-year study experience and could freely express their learning experiences with or without ICTs. This group was an appropriate choice owing to their age range which fell within that of the Net generation, since they were born after 1980. They were better qualified to participate in the study and thus to contribute to the findings relevant for filling the persisting knowledge gap on ICT-enabled pedagogy. The audio-recorded focus group's discussions included the identification of their ICT priorities, the meanings the cohort of students attached to the institutional ICTs, as well as their expected use in education. Data from the audio recordings were analyzed on ATLAS ti computer software, resulting in the emergence of the major themes, which are discussed in the subsequent section.

5 Findings / Results

The findings from the groups show that numerous ICTs are a priority to the current generation of students. The results from the participating students resulted in the emergence of the following themes relating to the types of ICT for teaching, learning and research.

5.1 Students' preferred ICTs for teaching and learning

The students who participated in the study identified various ICTs, particularly the cloud-based services and tools that they found useful in their learning. They indicated that these services and tools have a capacity to enhance and improve their learning activities, as well as their academic performance. Although the e-learning systems tools are widely used in higher learning institutions, the new generation of students prefers cloud-based services and tools, a choice that can be attributed to the services' affordability, availability and usability. Most of these services are free of charge and accessible to any user with an internet connection. The services are also compatible and device independent, making them user friendly and easy to use. In the next section, is a discussion of the popular cloud-based and other tools and services that the new generation of students is fond of and would like to see integrated in its learning process.

5.1.1 Google Classroom

The students indicated that they place a high value on this platform because of the functionalities it provides. These include sharing resources in a more convenient manner. It saves them both time and money, since the learning material can be shared and used on-line without having to download and implement on the local device, hence, it is light on storage and resource requirements. From the participating groups it was learnt that attending lectures in a Google Classroom environment is the most convenient and easiest way of learning. All they needed was to enroll in a class and then participate in the learning activities that took place. Google Classroom is a platform that is time and space independent. It allows the students to attend lessons anytime without any geographical constraints. It exposes them to collaboration with other students the world over and defeats the race, ethnic, tribal and regional boundaries. The students tend to benefit from the expertise provided by experts that may be in short supply in developing countries due to high staff turnover influenced by the economic challenges that force the local experts to migrate to greener pastures. According to the participating students, the most attractive aspect of the Google Classroom is the collaboration feature, enabling them to work on a project or assignments with their global counterparts. There is no need for the students have physical or face-to-face contact to complete their projects and assignments. They can take on-line tests and are saved from the pressures of relocating to accredited institutions to benefit academically. They are not stressed by the need to make hard copies of their assignments as they can submit them on-line and are also guaranteed instant feedback that enhances their learning. With the Google Classroom there is no fear of losing an assignment.

From the findings, Google Classroom is not a mandatory teaching and learning method within the institution in question, since the institutional ICT policy is silent in respect of this ICT-based education method. Despite the university's technological background and mission of being a world class user of emerging technologies to promote teaching and learning, there has been an insignificant effort towards the adoption of the Google Classroom innovation. Of the seven faculties at the institution, only one faculty has lecturers who have created and implemented a Google Classroom for their courses. The results show that the Departments of Mathematics and Computer Science are the ones that have initiated this innovation for the benefit of their students. The findings, however, show that 100% of the study participants have enrolled in the Google Classrooms created by their course instructors as individuals and not as a result of an institutional initiative, hence the responses below to the question: Have you ever been enrolled in a Google Classroom?

All the participants indicated having been part of Google Classroom for sharing resources, doing challenging assignments, taking tests in preparation for their examinations, as well as interacting with a course-related tutor.

In response to the question regarding the benefits the students derived from being part of the Google class, the participants unanimously agreed that the service offers the following benefits:

> *Being connected to other students and lecturers*
> *Real time notifications, tests, etc.*
> *Ease of use*
> *Hassle free*
> *Location independent*
> *Access to course material and tutorials*

Regarding the low uptake of the Google Classroom service in spite of its affordability, the participating students had the following recommendations:

> *It would be better if our lecturers could stream their lecturers*
> *Timed assignments/tests should be available*
> *Time limits need to be added for tests*
> *Integration with SMS services for notifications*

It is the students' perception that considering and addressing the aforementioned recommendations could increase the value and benefits derived from the Google Classroom teaching and learning strategy.

Another learning enhancement service of interest to the current generation of students is the YouTube tutorials.

5.1.2 YouTube video tutorials

In this section, the meanings that the students attach to the YouTube ICT resource are discussed. In response to the question as to whether the students have listened to an audio recording or watched a video on YouTube for learning purposes, the results show that all the participants had done so. The following are the identified reasons for using YouTube:

> *So many things pertaining to education*
> *Skills for Photoshop, etc.*
> *Programming, theory and DIY*
> *Troubleshooting a faulty laptop*
> *For most of the Courses during my 4 years at university/*
> *Learning different concepts*
> *Video tutorials both for academic and non-academic purposes*

When asked to describe the benefits of YouTube tutorials, the students had this to say:

> *There are many tutorials, different perspectives and views and it's free*
> *Very informative*
> *Better explanations.*
> *Different approaches to solving problems*
> *I learned knew ways of tackling science problems*
> *Different views on the same subject*

However, due to the limited use of this service by the educators from the particular university, the students recommended the following:

More subject areas can be expanded on
Educators should take advantage of this service and create live chats for real time interactions with us
Educators need to create high quality videos based on specific education levels.
Videos should be subject specific and grouped in accordance with the purpose.

5.1.3 Google Documents

The students are also interested in Google Documents, a feature provided by Google. The students' perception about Google Documents is that they are useful for:

Writing academic Documents
Document Editing
Collaboration on an assignment
Instant feedback

The advantage of Google Documents in learning is that:

They are always available where there is internet connection.
Two or more users can edit the same document simultaneously.
Access of documents is platform independent; that is from any computer or mobile device.

5.1.4 Mobile phones

As observed by the author, mobile phones were also popular among the current generation of students. All the participants possessed a mobile device and used it to take screenshots of the lecture notes, to take videos of the lecture, as well as to record proceedings of the group discussions. Mobile phones were also used to share learning content, and to consult and collaborate via Facebook and WhatsApp, as well as to take on-line tests and quizzes.

5.1.5 Sakai LMS

However, the institutional Sakai LMS was less popular amongst the cohort of students. They criticized its reliability and stability. It was unanimously agreed by the participants that the institutional LMS lacked flexibility and did not compare to the Google Classroom tests and quizzes that could be taken by multiple students concurrently without freezing or crashing.

6 Discussion

ICT circumstances differ depending on the context, although existing studies attribute the limited use of ICTs to a lack of ICT infrastructure due to budgetary constraints and inadequate funding, a lack of educators' interest, evident in the resistance to change, as well as continued management commitment [11, 12, 17]. Interesting ideas have emerged from this study. Contrary to the norm in literature of shifting the blame on individual, and organizational and environmental factors, the results of the study have indicated the choice of ICTs as the major constraint to the successful integration of ICTs in education. The findings indicate that even in the economically challenged contexts characterized by tardy implementation of technology, the current generation of students in Zimbabwe share similar perceptions and ICT interests with counterparts in the developing and the developed nations. The study cohort of students values emerging technologies, such as Google Classroom, Google Documents, YouTube, LMS, and smart phones, among others.

Like their counterparts the world over, these students get frustrated when confronted with the learning methods that are without technology. The only difference is the level of usage that does not compare to that of their counterparts, particularly in the developing countries. While this group of students engages more with the Google Classroom and Google Docs and is frustrated by the outdated computing infrastructure, its Tanzanian counterparts are rather discontent with their brand-new computer facilities, a rare commodity in the Zimbabwean context [13]. Another difference is that while the YouTube videos were difficult to use due to poor internet connectivity in the

Zimbabwean context, Matti and Minna [13] reveal that the students in Tanzania are content with the very slow Internet.

This cohort of students also prefers assessments based on the emerging technologies, such as Google Classroom instead of the traditional and popular learning management systems, which they criticised because of constant unavailability, inaccessibility and instability. Nevertheless, Grace, Hussein and Gary [5] show that LMS services are still popular, mostly desired and mostly accessed for learning by the students in the developing world. Due to the proliferation of mobile phones in Zimbabwe, the current generation of students in Zimbabwe indicated their interest in using these ICT devices to compensate for the institutionally slow and outdated computers. By contrast, Grace, Hussein and Gary [5] indicate that mobile phones were not so popular with other students in the other developing countries, since they rated mobile phones the least used devices for accessing the LMS services.

Furthermore, the cohort of students also likes to engage on social media to collaborate and consult on school-related issues. This confirms the observationby Zanamwe, Rupere and Kufandirimbwa [19] that the current generation of students prefers to engage on social media during their learning process. This is proof that these students are technologically savvy and thus prefer learning that is supported by ICTs as a means of problem solving and enhancing thinking skills. It is, therefore, important for the learning institutions to employ strategies and models that promote ICT-enabled teaching and learning for the benefit of the digital native students who share similar interests and expectations, regardless of their economic background. Despite the economic constraints of the developing countries, leap frogging has facilitated the increase in the affordability, availability and accessibility of the ICTs for education. What remains is the usability of such technologies for the benefit of the current generation of students' learning. Therefore, usability is still a problem in the HEIs in the developing countries, a scenario creating a digital divide between the developed and the developing nations.

7 Conclusion

In this study, the popular ICTs among the current generation of students were identified. How these students prefer to integrate ICTs in learning was also described and explained. Although traditional strategies of teaching and learning are still very common in developing country contexts, the findings concur with the existing literature that the current generation of students is digital natives who favor ICT-enabled teaching and learning methods for problem solving and collaboration. They derive comfort and satisfaction from multitasking provided by ICT-based leaning such as Google Classroom, Google Documents and YouTube video tutorials, which have been found to enhance the effectiveness of their learning and research. These findings are relevant to the decision makers of higher education institutions as they inform the future choices of ICTs that are of value to the students, as well as informing them on the aspects to consider in the formulation of ICT policy that facilitates the adaption of teaching to suit the current generation of students. It is therefore important for educators to make the necessary changes in their teaching approach in such a way that this new generation of students is satisfied with learning in the 21st century digital society and is well prepared for survival in the current labor market. The study was however limited in scope and methodology. Future similar studies could extend the case study by focusing on multiple case studies and a mixture of qualitative and quantitative data for more clarity and generalization of the findings.

References

1. Al-Busaidi, K.A. 2013. An empirical investigation linking learners' adoption of blended learning to their intention of full e-learning. *Behaviour & Information Technology*, 32(11):1168-1176.
2. Barbara, N. & Linda, B. 2014. Changing academic teaching with Web 2.0 technologies. *Innovations in Education and Teaching International*, 51(3):315–25.
3. Black, A. 2010. Gen Y; Who they are and how they learn. *Educational Horizons*, 92-102.
4. Faqih, K.M.S. 2016. Which is more important in e-learning adoption, perceived value or perceived usefulness? Examining the moderating influence of perceived compatibility. In *4th Global Summit on Education GSE*. Kuala Lumpur, Malaysia, 2016. World Conferences.
5. Grace, S., Hussein, S. & Gary, M. 2012. Learning Management Systems: Understanding the Expectations of Learners in Developing Countries. Lisbon, Portugal, Proceedings IADIS e-learning 2012.
6. Hitendra, P., Kym, I. & Andrea, M. 2006. Developing a diagnostic tool for assessing tertiary students' readiness for online learning. *International Journal of Learning Technology*, 2(1):92-104.
7. Hope, K. 2015. Distance Education and The Evolution of Online Learning in The United States. *Curriculum & Teaching Dialogue*, 17(1/2):21-34.
8. Isaacs, S. 2007. ICT in Education in Zimbabwe. Zimbabwe Country Report. Survey of ICT and Education in Africa.

9. Jones, C. & Ramanau, R. 2009. The Net Generation enters university: What are the implications for Technology Enhanced Learning? In *23rd ICDE World Conference on Open Learning and Distance Education*. Maastricht, NL, 2009.
10. Junginger, C. 2008. Who is training whom? The effect of the millennial generation. *FBI Law Enforcement Bulletin*, (77) 19-23.
11. Kabweza, L. 2015. TechZim. [Online] Available at: http://www.techzim.co.zw/2015/02/numbers-6-interesting-facts-telecoms-internet-zimbabwe [Accessed 26 August 2016].
12. Khan, S., Hasan, M. & Clement, C. 2012. Barriers to the introduction of ICT into education in developing countries: The example of Bangladesh. *International Journal of Instruction*, 5(2): 61-80.
13. Matti, T. & Minna, K. 2009. Students' Perspectives on Challenges of IT Education in Rural Tanzania. s.l., IST-Africa 2009 Conference Proceedings.
14. Ramírez-Correa, P., J, A.-G. & FJ, R.-C. 2015. Gender and Acceptance of E-Learning: A Multi-Group Analysis Based on a Structural Equation Model among College Students in Chile and Spain. *Plus ONE*, 10(10): 1-17.
15. Reynol, J. & Jeanna, M. 2007. Connecting to the Net Generation: What Higher Education Professionals Need to Know About Today's College Students. National Association of Student Personnel Administrators Inc.
16. Roberts, G. 2005. Technology and Learning Expectations of the Net Generation. In D.G. Oblinger & J.L. Oblinger, eds. Educating the Net Generation. 1st ed. North Carolina: Educause. pp. 32-38. Sauter, V., 2003. Project success and failure. [Online] Available at: http://www.umls.edu/~sauterv/analysis/6840_f03_papers/frese [Accessed 26 August 2016].
17. Tarus, J.K., Gichoya, D. & Muumbo, A. 2015. Challenges of implementing E-Learning in Kenya: A Case of Kenya Public Universities. *International Review of Research in Open and Distributed Learning*, 16(1): 120-41.
18. Wu, B. & Zhang, C. 2014. Empirical Study on continuance intentions towards E-Learning 2.0 Systems. *Behaviour & Information Technology*, 33(10): 1027-1038.
19. Zanamwe, N., Rupere, T., & Kufandirimbwa, O. 2013. Use of Social Networking Technologies in Higher Education in Zimbabwe : A learners ' perspective . *International Journal of Computer and Information Technology*, *2*(1):8–18.

Undergraduate students who regularly complete online self-reflective assessments reap academic success

Arthur James Swart

Central University of Technology, Bloemfontein, Free State,
South Africa, 9301
drjamesswart@gmail.com

Abstract. Reflective practice is defined as the ability of individuals to reflect systematically on actions in order to learn continuously. This is considered by many academics to be an excellent pedagogical approach, as deep learning, critical thinking and problem-solving is promoted. The purpose of this paper is to highlight the impact that regular online self-reflective assessments have on the academic success of first-year undergraduate students at a university of technology. A descriptive quantitative research design is used. Students are required to complete weekly online self-reflective assessments, on a learning management system, that are based on the previous week's theoretical work presented in a classroom. This involves both the recall and application of factual knowledge within an engineering context. The results tend to suggest that the more students engage with regular online self-reflective assessments by reflecting on and correcting wrong actions with each attempt, the better their chances of reaping academic success at the end of the module. This is verified by the fact that 85% of the 240 students who passed the module in 2016 achieved more than 60% for all their online self-reflective assessments. It is recommended that awareness be created among academics of the benefits that online self-reflective assessments can play in the teaching and learning process.

Keywords: Blackboard^{TM}, Theory, Practice, Pass rates

1 Introduction

"By three methods we may learn wisdom: first, by reflection, which is noblest; second, by imitation, which is easiest; and third by experience, which is the bitterest." These words, by the Chinese teacher Confucius, suggest that reflection is one of the noblest, if not most critical, pedagogical methods for acquiring wisdom. Imitating others and learning from experience may well be ranked behind the importance of reflection, which was originally proposed by Dewey [1] as a particular form of thinking. However, one must reflect, or meditate, on knowledge that is factual, be it either scientific or spiritual, acting upon it to become truly wise. In fact, wisdom has been defined as expertise in coping with difficult problems in life for which there is no simple solution [2]. Problem-solving is at the heart of many, if not all, Engineering and Information Technology (IT) programmes in higher education. McMurtrey et al. [3] found, in their study of IT field personnel, that soft skills, including problem-solving, critical thinking and collaborative skills, were deemed the most important for entry-level IT professionals. Students in higher education must therefore be helped to recall and apply their factual (or theoretical) knowledge in different circumstances, so as to become truly wise within their particular field of study. A key pedagogy that academics may use in helping their students to fuse their theory with practice, so as to achieve wisdom, is that of reflective practice.

According to Robinson [4], reflective practice is defined as "a process in which individuals are confronted by change and must make decisions on their own". Another definition, according to Schön [5], states that it is "the ability of individuals to reflect systematically on actions in order to learn continuously". To change requires reflection on our own "belief, judgement and feeling", so as to transform our world view [6]. Students must be encouraged to reflect, or meditate, on their newly acquired theoretical knowledge, in order to correctly apply it in practice and become wise. One of the many ways in which students can engage in reflective practice includes reflecting on feedback [7], which is intrinsic to online self-assessments which are included as one of the many features in learning management systems (LMS) [8].

Álvarez et al. [9] states, that despite the many advantages of LMS for supporting the process of teaching and learning, the true potential of LMS have not yet been fully utilized by academics or students. This suggests that many of the integrated features of a LMS are often underutilized by academics and students, including the online self-assessment feature. This, in itself, may be depriving undergraduate students of the opportunity to engage even more with the course content through means of reflective practice.

Subsequently, the purpose of this paper is to highlight the impact that regular online self-reflective assessments have on the academic success of first-year undergraduate students at a university of technology. A descriptive quantitative research design is used. Different types of assessment and reflective practices are firstly discussed. The research context of the study is then provided, followed by the research methodology, results and conclusions.

2 Types of assessments and reflective practices

The seminal work of Schön [10] identified three types of reflective practices, namely reflection-in-action, reflection-on-action and reflection-for-action, as shown in Table 1. All three practices find application within higher education, and especially so within a laboratory environment where practical instruction is completed. A key pedagogy of promoting reflective practice among students is by using reflective assessments. Reflective assessments are a formative process through which students can experience assessments as part of the learning process, rather than as a separate evaluative process [11]. Daniels and Bizar [12] suggest that the habit of being actively metacognitive can be developed by students doing reflective assessments, which may include strategies for problem structuring, problem solving and creativity [13]. Reflective practice furthermore forms the basis for deep-learning, critical thinking, self-directed learning and lifelong learning skills [14], which students need to carry away from higher education.

Table 1. Reflective practices, definitions [10] and possible applications

Reflective practices	**Definitions**	**Practical applications**	**Theoretical applications**
Reflection-in-action	Thought process that occurs as an experience unfolds, guiding action with the experience	Students thinking about what step or procedure to follow next while engaging in a laboratory experiment	Students thinking about which immediate actions need to be changed in a current assessment
Reflection-on-action	Thinking back on an experience to gain new knowledge or gain a better understanding	Students thinking back on a completed laboratory experiment in order to draw appropriate conclusions	Students thinking back about which actions they should have changed in a past assessment
Reflection-for-action	Drawing inferences from one's past experiences to create possible action plans for the future	Students thinking about mistakes which they made in their laboratory experiments so as to avoid them in the future	Students thinking about how they will further improve their actions for future assessments

Table 2 presents a summary of the three main types of assessment that may be used in higher education, namely diagnostic, formative and summative. Diagnostic assessments may be considered as “assessment-in-learning”, as academics seek to determine what students have learned or achieved in previous modules. Formative assessments may be considered as “assessments-for-learning”, as students seek to determine what they still need to learn or achieve in the current module, so as to achieve academic success at the end of the module. Summative assessments may be considered as “assessments-of-learning”, as academics seek to determine what students have learned or mastered at the end of a module, so as to make a final judgement whether the student has successfully complete the module or not.

Regular online self-reflective assessments would be linked to both reflection-in-action and reflection-for-action, as students engage with current assessments in a repetitive fashion. They may receive multiple attempts to complete the assessment, being encouraged to learn from each attempt by reflecting on their wrong answers which are identified at the end of each assessment. This would further serve to re-inforce correct actions for future summative assessments and to remove any misconceptions or errors in their current thinking, thereby leading to reflection-for-action. This links to formative assessment, as students are provided with ongoing feedback to improve on their actions so as to align them with their factual knowledge. Fusing their theoretical and practical knowledge in this way, by means of reflection, may very well lead to wisdom that may be used to benefit students, their communities and the country at large.

Table 2. Main assessment types, purpose, and links to reflective practice

Assessment types	Purpose	Links to reflective practices
Diagnostic (assessment-in-learning)	Determine the specific weaknesses and strengths of students at the start of a semester [15]	Reflection-on-action as students think back on what knowledge they have gained and bring into the classroom
Formative (assessment-for-learning)	To provide a feedback loop to check for progress and detect learning gains, strengths, weaknesses, and narrow any gaps in teaching and learning through a semester [16]	Reflection-in-action and for-action as students think about changes to make in a current assessment to improve their chances for future assessments
Summative (assessment-of-learning)	To determine mastery of the course material and for accountability purposes, such as assigning a grade, at the end of a semester [17]	Reflection-in-action as students think about changes to make in a final assessment

3 Research context of this study

The context of this study is limited to the field of Engineering, where undergraduate students, enrolled for a module termed Electronics 1, are considered. This module is a compulsory offering or module in the first year of the National Diploma: Engineering: Electrical qualification in South Africa. This Diploma is a NQF (National Qualifications Framework) Level 6 qualification that requires students to obtain a minimum of 360 credits to successfully complete it. The majority of modules in this Diploma have a credit value of 12, indicating that students should dedicate at least 120 notional hours to each module. Approximately 300 students register for this module in Semester 1 of each year, with about 150 registering in Semester 2 (each semester is approximately 14 weeks in duration). The syllabus of Electronics 1 covers seven sections or units focusing on the oscilloscope, electrical basics, Thevenin's Theorem, resistors, capacitors, diodes, transistors and the design of power supplies and amplifiers.

The structure of the module is shown in Table 3, which covers 14 weeks. Each unit is discussed consecutively over a period of one week (2 x 45 minute sessions scheduled twice per week). Once a unit is complete, then an online self-reflective assessment is made available for 6 days to further engage students with the content. These assessments feature a balance between higher-order and lower-order questions [18], requiring a student both to recall factual knowledge and apply it in solving broadly defined engineering problems that requires engineering knowledge, problem analysis and the design of solutions (graduate attributes required by the International Engineering Alliance [19]). The online self-reflective assessments further feature many calculations, as many fields of engineering require a firm mathematical foundation for the design and evaluation of systems [18]. During the 14 weeks, students also engage with practical work in a laboratory, where they are required to complete six practical assignments and one practical assessment. This practical work is also designed to help students fuse theory and practice [20]. Three of the six practical assignments are selected along with the practical assessment to make up the practical grade of the student, as shown in Table 4.

Table 3. Structure of Electronics 1

Semester 1 and 2 of 2016	Time
Unit 1 – Unit 7 presented systematically over 7 weeks	Weeks 1 – 7
An online self-assessment is scheduled after each unit has been completed and is available for 6 days	Weeks 2 – 8
Main test written in a computer laboratory covering Units 1 – 5	Week 9
University recess of approximately 10 days	Week 10
Review of Units 1 – 7 by focusing mainly on problem solving	Weeks 11 – 13
Practical work completed and course grades assigned	Week 14

Table 4. Assessment weightings of Electronics 1

Assessment	Weightings
Test 1 (T1): Six online self-assessments + one peer assessment	25%
Practical grade: 3 practical assignments + 1 practical assessment	35%
Main test (T2): First 5 units covered in a laboratory with computer access	45%
Course grade total	100%
Weighting of course grade to final grade	40%
Examination grade	100%
Weighting of examination grade to final grade	60%

Grades are awarded to each of the online self-reflective assessments, depending on the number of questions contained in each one. The first assessment features approximately 20 marks, with a gradual increase in marks for each subsequent assessment. The final online self-reflective assessment would normally feature around 50 marks. This gradual increase in marks helps to prepare these first-year undergraduate students for larger assessments, with the 100-mark final summative examination in mind. The first six online self-assessments contribute to Test 1, and counts 25% towards the course mark. Students need to obtain a course mark of at least 40% to gain entry into the final summative examination.

The results of this paper focus only on Test 1, as it comprises six online self-reflective assessments where students engage in reflective practice. The assessments are compiled in Respondus (PC based software for setting and managing online tests and surveys), and then uploaded to the institutions LMS. Students are afforded three attempts to complete the assessment. This provides them with the opportunity to improve on their first attempt, as the individual scores per question are revealed at the end of each assessment. This feedback prompts students to engage in reflective practice (reflection-in-action), as they think back to their original answer (which may be wrong) and seek to understand where they went wrong in its formulation. The three attempts are often sufficient for students to identify and rectify any misconceptions or errors in their current thinking, as many students significantly improve on their first attempt grades. The system is set to record the highest grade of the three attempts made by the students.

4 Research methodology

A descriptive quantitative research design is used. Yin [21] describes three types of case studies, being exploratory (examines a situation where an intervention produces no single clear result), descriptive (describes a situation within a real life context) and multiple case studies (discover the differences between and within cases). A descriptive case study is used in this research as the phenomenon (online self-reflective assessments) is described within a real life context (first-year engineering students enrolled at a university of technology in South Africa). The independent variable of interest is the online self-reflective assessments and the dependant variable of interest is the final academic results of the undergraduate students registered for the module. In other words, it is hypothesized that the final academic results of these students will be dependent on their engagement with the online self-reflective assessments.

Descriptive statistics, rather than inferential statistics, are used as the results are interpreted with regard to specific engineering students enrolled at a university of technology. These descriptive statistics include the student profile and grade results of students enrolled in two different semesters of 2016 (quantitative data). Test 1 grades along with the pass rates and throughput rates of the module for each semester is considered. The quantitative data focuses on the performance of undergraduate students in the online self-reflective assessments, contrasting their highest grade awarded, from their three attempts, to their individual grades of their final summative examination at the end of the semester.

The target population is therefore restricted to all engineering students enrolled for the module Electronics 1 (ELE11 presented during Semester 1 and ELE12 presented during Semester 2) during 2016. No sampling technique was used as the results of all the students were included. The sample size for Semester 1 of 2016 was 294 while the sample size of Semester 2 was 148 (total n = 442). Student demography is also shown in order to contextualize the makeup of the student body.

5 Results and discussions

The profile of students registered for ELE11 and ELE12 are shown in Fig. 1 (home languages) and Table 5 (gender and age brackets). These results show very little difference between the two groups of students, where both groups major home language was Sesotho (indicative of the Free State province in South Africa [22]).

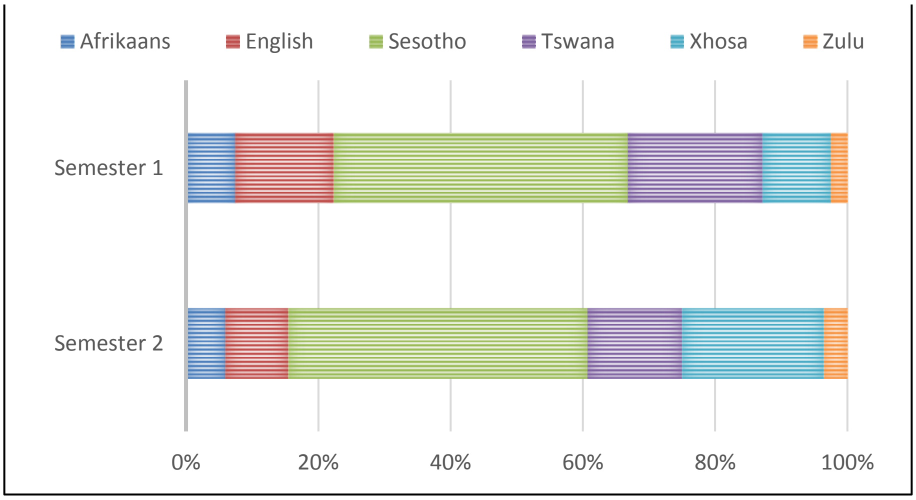


Fig. 1. Home languages of the undergraduate students

The minority of students are female, which is one of the reasons why a global drive exists to encourage more women in engineering [23]. The majority of students were younger than 25 years, which validates them as first-year undergraduate students who have completed their secondary or high school career (average age for Grade 12 learners in South Africa is 18 years [24]). The total number of students who registered for Semester 1 was 294, while those who registered for Semester 2 where 148. Clearly, more students register in Semester 1, which runs from February to May of each year, as many have just completed their school career in November of the previous year.

Table 5. Age brackets and gender for undergraduate students in 2016

	Semester 1	Semester 2		Semester 1	Semester 2
Age brackets			**Gender**		
Less than 25 yrs.	93%	90%	Male	86%	79%
25-29 yrs.	7%	8%	Female	14%	21%
30-34 yrs.	0%	2%			

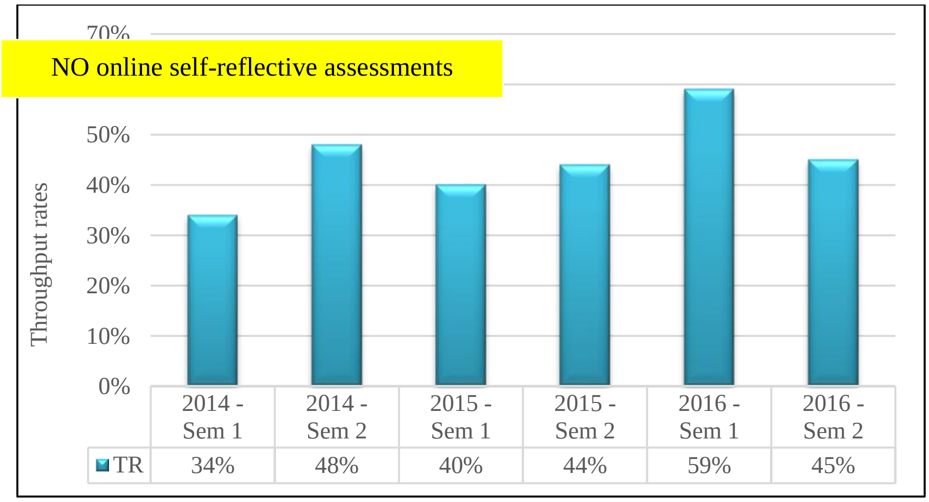
Table 6 illustrates the throughput rates of the undergraduate students in 2016. Throughput rates are defined as the number of students enrolling for a given module to the number of students passing the final summative examination for that module. It is evident that the overall module throughput rates are relatively low (59% for Semester 1 and 45% for Semester 2). It must be noted that many students register for modules in South Africa, but drop out before they even reach the final summative assessment. This picture is very similar at the Central University of Technology (CUT) where statistics indicate that of the 2014 cohort of students, only 26.6% have graduated to date, with another 22.7% still busy with the qualification for which they first registered in 2014. This means that 50.7% of the 2014 cohort have dropped out of their original study programme, but not necessarily out of the university [25].

Table 6. Throughput rates (TR) of the undergraduate students in 2016

Test 1 grade brackets	**Semester 1 TR**	**Semester 2 TR**	**Total Passing**	**% passing**
>80	89%	84%	88%	46%
70-80	82%	74%	78%	
60-70	64%	52%	58%	
50-60	47%	33%	40%	8%
40-50	50%	33%	42%	
30-40	31%	0%	15%	
<30	2%	3%	2%	
Module TR	**59%**	**45%**	**52%**	**54%**

However, it is also noteworthy to review the throughput rates for 2014 and 2015 for this module in this regard, as shown in Fig. 2. Clearly, the throughput rate of Semester 1 in 2016 is much higher than it has been in the preceding two years. The throughput rate of Semester 2 of 2016 (45%) is similar to that of 2015. However, it must be noted that many universities faced student unrests during the latter part of 2016, which was also the case at CUT [26]. This may have caused the final throughput rate of Semester 2 to drop below 50%.

Table 6 presents different grade brackets for Test 1 on the left hand side. The next column (Semester 1 TR) represents the percentage of students within each grade bracket of Test 1 who eventually completed the module. In other words, 88 students achieved more than 80% for Test 1 (six online self-reflective assessments) in Semester 1. Of these 88 students, eventually 78 passed the module with more than 50% giving rise to the 89% throughput rate listed on top of the second column. The third column represents this result for Semester 2. The Total Passing column represents this result for both Semesters, which indicates that 88% of all students in 2016 who achieved more than 80% for Test 1 (comprising the six online self-reflective assessments) successfully completed the module, thereby achieving academic success.



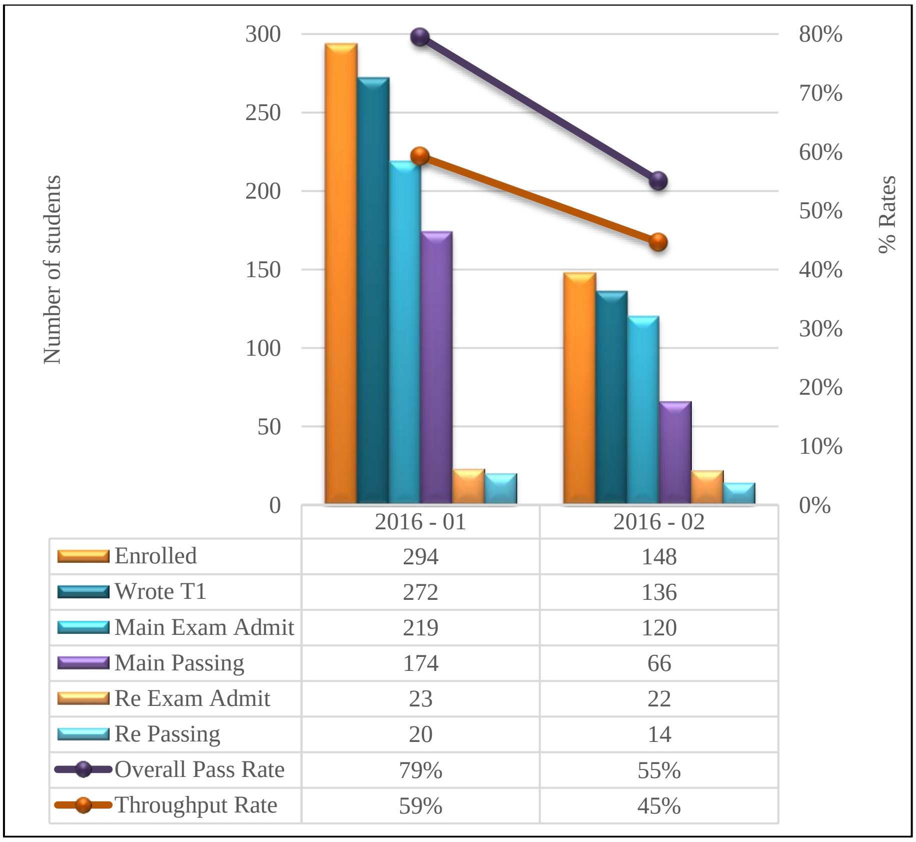
NO online self-reflective assessments

Fig. 2. Throughput results for 2014 – 2016

The highlighted block, in Table 6, shows that of the 54% (representing 240 students out of the original 442) of students who eventually passed this module on Electronics, 46% had achieved more than 60% for all their online self-reflective assessments. The total number of students who achieved more than 80% for Test 1 in 2016 equaled 113 (88 Semester 1 students + 25 Semester 2 students). To have achieved this high percentage indicates that they must have engaged regularly with and completed all the online self-reflective assessments. They must have engaged more with the course content, as they reflected each week on the previous week's work that was presented in the classroom. They must have been able to learn from their mistakes in their first attempt at the assessment, as the feedback provided by the system enabled them to identify and rectify any misconceptions or errors in their current thinking, thereby improving on their actions in their second and third attempt. Of these 113 students, 99 achieved academic success at the end of the module by obtaining more than 50% for their final grade, which equates to a success rate of 88%.

This tends to suggest that the more students engage with regular online self-reflective assessments by reflecting on and correcting wrong actions with each attempt, the better their chances are of academic success at the end of the module. Students who achieved higher grades for their online self-reflective assessments are therefore more likely to achieve academic success at the end of the module (the total passing percentage drops systematically from 88% to 2 % with each grade bracket of Test 1).

The overall results for 2016 is shown in Fig. 6. The total number of students who registered for this Electronics module in 2016 added up to 442 (294 + 148). Of these 442 students, only 339 (219 + 120) gained entry into the final summative examination at the end of the semester, as their final course mark was more than 40%. Of these 339 students, 240 (174 + 66) students eventually passed the module in 2016 equating to a pass rate of 70.8%. The number of students admitted to and passing the re-examination is also shown. This re-examination is made available to students who obtained a final grade of between 46 and 49% for their final summative examination.



	2016 - 01	2016 - 02
Enrolled	294	148
Wrote T1	272	136
Main Exam Admit	219	120
Main Passing	174	66
Re Exam Admit	23	22
Re Passing	20	14
Overall Pass Rate	79%	55%
Throughput Rate	59%	45%

Fig. 3. Overall results for Semester 1 and 2 in 2016

6 Conclusions

The purpose of this paper was to highlight the impact that regular online self-reflective assessments have on the academic success of first-year undergraduate students at a university of technology in South Africa. A descriptive case study was used in this research as the phenomenon (online self-reflective assessments) was described within a real life context (first-year engineering students enrolled for an Electronics module at a university of technology in South Africa). Students were granted three attempts to complete each of the six online self-reflective assessments which were systematically made available over a six-week period. Grades were awarded to each assessment that contained a balance of recall and application questions which were based on the factual knowledge that students received in the previous week during the classroom discussions.

Results indicate that, of the 442 students who first registered for this module, 339 gained entry into the final summative examination where 240 eventually attained academic success. Of these 240 students who passed the module, some 204 achieved grades of more than 60% for all their online self-reflective assessments. Narrowing this further down revealed that of these 204 students, some 99 achieved more than 80% for Test 1 (representing all the online self-reflective assessments). These results tend to suggest that the more students engage with regular online self-reflective assessments by reflecting on and correcting wrong actions with each attempt, the better their chances are of academic success at the end of the module. This is the key contribution of this paper.

Prior to introducing the online self-reflective assessments, the average throughput rate for the Electronics module was 42% for 2014 and 2015. Then in Semester 1 of 2016, it jumped to 59% with the introduction of the online self-reflective assessments. Two variables exist that have not yet been accounted for in this results. Firstly, the structure of the module was modified between 2015 and 2016, to accommodate these type of assessments. Secondly, the academic lecturer was changed between 2015 and 2016, to try and improve the low throughput rate in Electronics 1. The modification of the module structure would not influence the throughput rate that much, as the syllabus and course content remained the same between 2015 and 2016. The change in academic lecturer may have had a greater influence on the academic success of the undergraduate students, and subsequently on the throughput rate. Further research is required to eliminate this last variable and establish the key contribution of this paper relating to the benefits of using online self-reflective assessments to promote further student engagement and academic success.

Although this study is only limited to one calendar year, the similarity in results from students in two different semesters, who were exposed to this pedagogy, does tend to indicate reliability. A possible recommendation would be to conduct a longitudinal study over a few years to further validate the results of this study. Another recommendation would include creating awareness among academics of the benefits that online self-reflective assessments can play in the teaching and learning process.

Confucius once stated that reflection is one of the noblest methods for acquiring wisdom. When undergraduate students regularly engage in online self-reflective assessments, they can only benefit themselves as they acquire wisdom through systematically recalling and applying their factual knowledge.

References

1. J. Dewey, "How we think. Buffalo," *PrometheusBooks, New York*, (1933).
2. M. Linden, "Wisdom and Wisdom Psychotherapy in Coping with Stress," in *Somatization and Psychosomatic Symptoms*, ed: Springer, pp. 273-281, (2013).
3. M. E. McMurtrey, J. P. Downey, S. M. Zeltmann, and W. H. Friedman, "Critical skill sets of entry-level IT professionals: An empirical examination of perceptions from field personnel," *Journal of Information Technology Education,* vol. 7, pp. 101-120, (2008).
4. E. T. Robinson, "Applying the Theory of Reflective Practice to the Learner and the Teacher: Perspective of a Graduate Student," in *Annual Meeting of the American Educational Research Association* Chicago, IL., (1997).
5. D. A. Schon, *The reflective practitioner: How professionals think in action*. New York: Basic books, (1984).
6. J. Gladman, C. Ryder, K. Walters, and L. Walters, "Measuring organisational-level Aboriginal cultural climate to tailor cultural safety strategies," *Rural and remote health,* vol. 15, pp. 1-10, (2015).
7. S. Boyd, "Practical Strategies for Engaging Dissonance in Veterinary Medical Education," in *Engaging Dissonance: Developing Mindful Global Citizenship in Higher Education*, ed: Emerald Publishing Limited, pp. 173-190, (2017).
8. A. J. Swart, "Student usage of a learning management system at an open distance learning institute – A case study in electrical engineering," *IJEEE, International Journal of Electrical Engineering Education,* vol. 52, pp. 142-154, (2015).

9. A. Álvarez, M. Martín, I. Fernández-Castro, and M. Urretavizcaya, "Blending traditional teaching methods with learning environments: Experience, cyclical evaluation process and impact with MAgAdI," *Computers & Education*, vol. 68, pp. 129-140, (2013).
10. D. A. Schön, *The reflective practitioner: How professionals think in action*. London: Harpur Collins, (1983).
11. L. E. Bernold, "Journaling and sketching in support of formative learning in engineering," in *Teaching, Assessment and Learning for Engineering (TALE), 2013 IEEE International Conference on*, pp. 490-494, (2013).
12. H. Daniels and M. Bizar, *Teaching the best practice way: Methods that matter, K-12*. Portland, Maine: Stenhouse Publishers, (2005).
13. D. W. Barnett, S. H. Bell, and K. T. Carey, *Designing preschool interventions: A practitioner's guide*. New York: Guilford Press, (2002).
14. C. Tsingos, S. Bosnic-Anticevich, and L. Smith, "Reflective practice and its implications for pharmacy education," *American journal of pharmaceutical education*, vol. 78, pp. 18-28, (2014).
15. H. Walia, "Assessment and evaluation," in *Dimensions of Innovations in Education*, A. Mathur, S. Kaur, Y. Sharman, and J. Padmanabhan, Eds., ed New Delhi: New Delhi Publishers, p. 107, (2015).
16. L. Connors-Tadros and D. Schilder, "Resources to Inform Technical Assistance on Formative Assessment. CEELO FastFacts," *Center on Enhancing Early Learning Outcomes*, (2013).
17. D. C. Emanuel, C. G. Robinson, and P. Korczak, "Development of a formative and summative assessment system for AuD education," *American journal of audiology*, vol. 22, pp. 14-25, (2013).
18. A. J. Swart, "Evaluation of Final Examination Papers in Engineering: A Case Study Using Bloom's Taxonomy," *IEEE Transactions on Education*, vol. 53, pp. 257-264, (2010).
19. International Engineering Alliance, (20 August, 2013). *Graduate Attributes and Professional Competencies*. Available: http://www.ieagreements.org., (2013).
20. A. J. Swart, "Does it matter which comes first in a curriculum for engineering students – theory or practice?," *IJEEE, International Journal of Electrical Engineering Education*, vol. 47, pp. 189-199, (2010).
21. R. K. Yin, *Case study research - Design and Methods*. Thousand Oaks: SAGE publications Inc, (2009).
22. M. J. Mahloane and S. Trausan-Matu, "Metaphor Annotation in Sesotho Text Corpus: Towards the Representation of Resource-Scarce Languages in NLP," in *Control Systems and Computer Science (CSCS), 2015 20th International Conference on*, pp. 405-410, (2015).
23. J. M. Basart, M. Farrús, and M. Serra, "New ethical challenges for today engineering and technology," *Telematics and Informatics*, vol. 32, pp. 409-415, (2015).
24. S. Kruger and E. Sonono, "" Black gold" grade12-learners: relationship between leisure/sport and satisfaction with life," *South African Journal for Research in Sport, Physical Education and Recreation*, vol. 38, pp. 113-127, (2016).
25. Central University of Technology, "Cohort statistics - student graduation for multiple base years," 2017.
26. A. J. Swart and P. E. Hertzog, "The Impact of Student Unrest on Freshmen Engineering Students in South Africa," in *EDUCON 2017, IEEE Global Engineering Education Conference*, Athens, Greece (2017).

A Framework for University Partnerships Promoting Continued Support of e-Schools

Mattheus Johan Vorster[1] and Leila Goosen[2]

[1]Monash South Africa, Faculty of Business, Engineering and IT
144 Peter Road, Ruimsig 1732, South Africa
johan.vorster@monash.edu
[2]University of South Africa, College of Education
Preller Street, Muckleneuk, 0003 Pretoria, South Africa
GooseL@UNISA.ac.za

Abstract. This paper reports on the development of a draft framework describing partnership efforts, where universities continually support and enhance the success of e-schools along various dimensions. The framework reported here is the outcome of the first stage of a broader study, and was developed using a Design Science approach. The theoretical framework for this study was based on the Information Diffusion Theory, the Technology Acceptance Model, and the Levels of Teaching Innovation framework. Secondary data included success factors and barriers to educational technology integration identified from existing literature and policy documents. The next stage of this study follows a multi-case study approach to verify and refine the draft framework, using multiple schools as cases. The dimensions identified in the framework were derived from the major activity areas relevant to schools, such as management and administration, teaching and learning, and assessment, as well as enabling factors, such as policy and budget, infrastructure and connectivity, and training. The framework was developed based on the theory that diffusion happens at different rates. It was divided into three stages (School Implementation, Teacher Implementation, and Sustaining), which relate to the classification of barriers to educational technology integration. It is recommended that schools and universities use the framework to guide the inclusion and timing of activities as they jointly develop plans for collaboration from initial introduction towards the continued enhancement of the integration of educational technologies.

Keywords: University Partnerships, Supporting e-Schools

1 Introduction

The South African educational landscape has been facing a vast number of challenges for many years, culminating in a generally ineffective and expensive education system that does not adequately prepare students for higher education and the world of work. The effective integration of Information and Communication Technologies (ICTs) into education at all levels presents a significant opportunity to transform the poor efficiency and productivity of teaching and learning in the school and post-school system [1, 2]. Across the African continent, it has been recognized that significant investment in the development of e-education, e-learning and the integration of ICTs into education is a strategic imperative for facilitating the development of knowledge societies and the effective participation of African countries in the knowledge / digital economy. This is widely evident in initiatives such as the New Partnership for Africa’s Development (NEPAD) e-schools Initiative across Africa [3], and the establishment of the iKamva National e-Skills Institute in South Africa [4].

The policy goal of the 2004 White Paper on e-Education states that every “South African learner in the general and further education and training bands will be ICT capable (that is, use ICT confidently and creatively to help develop the skills and knowledge they need to achieve personal goals and to be full participants in the global community) by 2013.” All schools in the country should transform into e-schools, e.g. “use ICTs to enhance teaching and learning”, “for planning, management and administration”, and ICT resources to support curriculum delivery [1, p. 42]. The goal of the White Paper on e-Education, however, has not been reached and the South African education landscape continues to be plagued by marked inequalities, and poor achievement and success

SACLA’17, July 3–5, 2017, Magaliesburg, South Africa.

rates. This is compounded by the poor quality training of many teachers currently in the system, as well as the impact of socio-economic factors on students and communities [5]. The rate of adoption of ICT in different schools, and by different teachers also shows wide disparities that tend to reflect the socio-economic inequalities of the broader society.

The policy set out in the White Paper on e-Education focuses largely on the General Education and Training (GET) and the Further Education and Training (FET) bands, including what is now known as Technical Vocational Education and Training (TVET) colleges, but the same concepts and objectives can be argued to apply to Higher Education Institutions (HEIs) as well. In addition, the provisions of the White Paper for Post-School Education and Training [2] also expresses the need for closer cooperation between HEIs and the school system in developing the skills required by teachers and learners.

Even though the strategic objectives of the White Paper on e-Education have not been realised, there are several indications that the envisaged goals are still seen as relevant, and that the process should be accelerated [4, 6]. This study investigates the success factors and barriers of integrating ICT in education within schools. The purpose of this paper is to report on the development of a draft version of a framework for supporting partnership efforts with universities to continually support and enhance the success of e-schools along various dimensions.

2 Theoretical Frameworks

This study references various theories as part of its theoretical background. These theories are the Innovation Diffusion Theory (IDT); the Technology Acceptance Model (TAM) and its variants including the Unified Theory of User Acceptance of Technology (UTUAT) and the Levels of Teaching Innovation (LoTi) framework.

2.1 The Innovation Diffusion Theory (IDT)

Rogers' landmark model of diffusion of innovations has become a widely used theoretical framework applied to the study of innovation in various disciplines, and is seen as highly relevant to investigating adoption of technologies in education settings [7]. It is particularly useful in early stages of technology integration, as well as change processes [8]. The IDT will be used in this study, as it relates to the adoption and integration of ICT in e-schools. Within the field of Information Systems, it has been found that the characteristics of relative advantage, complexity, result demonstrability, and image are most predictive of users' intention to adopt technology [9].

For the purpose of this paper, only two concepts from IDT will be highlighted. The first relates to **adoption** being a process involving various categories of adopters [8]. Certain parts of the population, including innovators, early adopters and early majority are seen to readily adopt a proposed innovation. On the other hand, late majority and laggards will show resistance to the process of adopting a new innovation. Secondly, **diffusion** is the process of communicating an innovation between members of a social system through the use of certain communication channels over time [10]. The IDT suggests that innovations are first adopted by small numbers of adventurous pioneers, and taken up by the majority only when the value of the innovation becomes apparent. This notion is often popularly called the critical mass, but could be more accurately described as the tipping point. This framework may be applied to schools at two levels. At the macro level, individual schools may adopt ICT at different times. At the micro level within each school, teachers adopt ICT at different times.

Schools that are successful in integrating ICT can serve as a model to provide inspiration and motivation to other schools, particularly where these innovations are communicated and shared. Likewise, teachers that are champions of ICT integration within their schools can serve as role models and support to other teachers, particularly within their own schools. Where there are networks to communicate and share their experiences, they can also extend this influence to teachers at other schools.

2.2 The Technology Acceptance Model (TAM) and the Unified Theory of Acceptance and Use of Technology (UTAUT)

The Technology Acceptance Model (TAM) was first proposed by Davis in 1986 and has since been widely applied in Information Systems research related to acceptance of various technologies in a variety of settings, including business and education [11]. A review of fifty papers published from 1999 to 2010 and using TAM as their theoretical framework found that that these papers were mostly in the field of education, covering a wide range of areas such as online learning, internet, teacher attitudes, Web 2.0, interactive whiteboards, tablets, etc. [10].

TAM is intended specifically to explain user acceptance of information systems and technologies, by predicting the acceptability of a technological innovation and identifying the modifications, which must be made to a system so that it becomes acceptable to users. Two main factors serve as predictors. These are Perceived Usefulness (PU: the degree to which a person believes that the use of a system will improve their performance) and Perceived Ease of Use (PEU: the degree to which a person believes that the use of a system will be free of effort). Studies have demonstrated that Perceived Usefulness is found to be the most important predictor of user intention [10]. TAM incorporates the notion of External Variables to facilitate research on the influence of various external factors. In their review of existing research based on TAM, Lee, Kozar and Larsen [12] found that the external variables most frequently used include system quality, training, compatibility, computer anxiety, self-efficacy, enjoyment, computing support, and experience. Based on the original model, further extensions to TAM have been proposed by several researchers, in order to introduce additional variables as antecedents to PEU and PU [10, 13].

While we will refrain from discussing all the variants of TAM here, we will include a brief description of the UTAUT. Venkatesh, Morris, Davis and Davis [14] reviewed eight prior theories and models (including TAM and IDT) in order to select conceptual and empirical similarities from these models for their proposed unified approach. Research based on the use of the UTAUT has shown that prior models such as TAM were able to explain approximately 40% of user acceptance of technology, while the UTUAT succeeds in explaining approximately 70% of user acceptance [10, 14]. This would indicate that the UTAUT is a useful theory to apply as the primary theoretical framework for the current study, although it has to be supplemented by use of other theoretical models and theories in order to address the research questions that have been set. In fact, Venkatesh et al. [14] encourage future researchers to include antecedents identified in prior research on the various models in order to derive a richer understanding of technology adoption and usage, in particular with reference to moderating influences, different technologies, different user groups, and other organizational contexts. The UTAUT includes four key variables that serve as direct determinants of user acceptance and usage behavior [14] viz. performance expectancy, effort expectancy, social influence and facilitating conditions. In addition, four key moderators are identified, viz. gender, age, voluntariness, and experience.

2.3 Consolidating the Variables from IDT, TAM and UTAUT for this study

In addition to the four main UTAUT variables described earlier, other significant variables in the field of education have been identified in a review of TAM studies by Yucel and Gulbahar [10]. Their analysis identified four themes with related variables as shown in Table 1. Their analysis further identified the variables that were found to be respectively the most effective and the least significant as shown in Table 2.

Table 1. Variables/themes in TAM studies on education [10]

Course / Content	Software and Tools	Learner / Instructor	Organizational Issues
Satisfaction	Usage of system and tools	Technological competency	Organizational support
	Investigation of system and tools	Demographic characteristics	Administration support
	Satisfaction	Anxiety Perceived Enjoyment Satisfaction Attitude Intention Perceived ease of use Perceived usefulness Subjective norm	Organizational culture

Table 2. Most and least significant TAM variables [10]

Most significant TAM variables	Least significant TAM variables
Perceived usefulness	Anxiety
Perceived ease of use	Organizational effects
Intention	Satisfaction
Investigation of systems and tools	Perceived enjoyment
Technological competency	Demographic characteristics

The classifications shown above have an important bearing on this study, as the variables will be important in guiding the collection and analysis of data. It is, however, important that caution must be taken to not allow the indicated significance of variables shown above to limit the study. For instance, it is surmised that within the setting of an e-school, organizational effects will be found to be significant given the social context of the e-school.

2.4 The Levels of Teaching Innovation (LoTi) Framework

A major criticism that has been applied to studies based on the TAM is that system usage is frequently measured by self-reported use, which is a subjective measure. Several researchers have cautioned against this approach, as it introduces common method bias [12, 13]. For the purpose of this study, it is also useful to note a further criticism by Lee et al. [12], where they indicate that TAM studies are often applied to tasks that are too broad in nature, while research indicates that perception of a technology varies according to the task type. Based on these shortcomings, this study makes use of the levels of the LoTi framework as a basis for investigating the tools being used in e-schools, and for evaluating the level of integration of those tools. This will reduce the effect of self-reported usage, as the evaluation of technology integration will be assessed from the primary data gathered.

The LoTi framework was first introduced in 1994 as a tool to help with quantifying the use of technology in the classroom [15]. LoTi now includes three frameworks, which are supported by several different measurement tools. Given the focus of this study on technology adoption and integration, two of these frameworks are employed. The first of these is the original LoTi framework to measure the use of digital tools and resources in the classroom, where level 0 denotes non-use, and level 6 denotes seamless integration into educational activities. The second framework used measures the fluency levels of teachers in using digital tools and resources for student learning, with level 0 describing teachers that have little skill or confidence in using technology, and level 7 describing expert users.

3 Methods and Techniques

As described above, this paper reports on the draft partnership framework developed as the first stage of the broader study. Based on the purpose described earlier, the specific objectives of this study are to:

- establish a cohesive set of dimensions that can be used to describe an e-school holistically.
- identify the relevant set of success factors and barriers as they relate to those dimensions,
- identify the criteria along each dimension, where universities can partner with e-schools to enhance the adoption and success of the e-schools
- develop a conceptual framework of activities and interventions for partnership and collaboration between universities and e-schools.

To achieve these objectives, a qualitative study was undertaken, which includes the following stages:

- The first stage employed a Design Science approach to develop the draft framework reported in this paper.
- The next stage follows a multi-case study approach. This stage generates in-depth insight regarding how the case schools overcame barriers to the adoption of ICT, and these insights are applied to evaluate and refine the draft framework developed in the first stage.
- To ensure rigor in method and results, a last step was identified to include expert reviews of the final framework.

In the development of the draft framework in the fists stage of this study, secondary data in the form of existing literature and policy documents were used to identify the envisaged uses of technology in e-schools, as well as the success factors and barriers that have been identified by other researchers. Success factors and barriers to the successful integration of ICT in education has been studied by a large number of researchers, as can be seen from the review of literature undertaken by Bingimlas [16] and the British Educational Communications and Technology Agency (BECTA) review [17].

For the purposes of this study, it was necessary to examine how schools have achieved success and overcome the barriers that potentially exist. This understanding informs the inclusion of activities and interventions in the proposed draft framework. The theoretical framework described earlier also contributed important insights in the development of the draft framework.

The second part of the study takes the form of a multi-case study, where a number of schools are used as the individual cases. In each of the study cases, it is necessary to investigate the current state of technology integration in teaching and learning, and administration and management of the school. Of particular interest is investigating how e-schools achieve successful integration of ICT, and how they overcome barriers to this. This in turn will inform the further refinement of the proposed framework.

The design of the data collection instruments for this primary data are based on exploring the variables identified in the UTUAT, as well as other relevant antecedent variables, with a view to gaining an understanding of the factors that influence stakeholders in e-schools to adopt and make use of technology. An important part of this process will be to investigate the enablers and barriers that the stakeholders in e-schools have experienced, and how they overcome identified barriers.

While it is important to gather qualitative opinions relating to success and barriers from classroom-based teachers, the BECTA study also stresses that opinions from other stakeholders, such as principals, are important as they would have different perspectives [17].

Primary data being gathered from each of the schools in the case study are as follows:

- Semi-structured interviews with principals and/or technology specialists, including school-level information such as budgets, policy and training related to ICT, as well as the availability of ICT resources for various uses by stakeholders in the school community. Important here is how a conducive environment for technology integration was (or could be) established, and exploring the facilitating conditions for this.
- School-level data could also be found by observation and recording of available resources, and from analysis of school documents such as policy documents, budgets, floor plans, etc.
- Semi-structured interviews with teachers, used to gather information on their skills and attitudes towards technology; and use and availability of ICT resources for teaching and learning, as well as administration activities. Of particular importance here is exploring the barriers experienced by teachers, and the approaches that enabled (or could enable) those barriers to be overcome.

Both in-case and cross-case analysis will help to identify how schools have achieved success in overcoming barriers and integrating ICT into school activities. These insights must then be used to populate further detail in the proposed framework.

As mentioned earlier, it is important to evaluate the final framework, to ensure rigor in methods and results. Expert reviews will be employed to gather feedback on the final framework. While the previous stages of the study focused on respondents from the case schools, these experts will be drawn from the school system as well as from universities.

4 Findings

This project is work in progress, and the reported in this paper relates only to the draft framework developed in the first stage of the study. The draft framework was developed using a Design Science approach, drawing on analysis of literature, secondary data, and the theoretical framework. In the next stage insights gained from the multi-case study will inform how the case schools have overcome barriers / challenges, and how these schools have turned these around, which will lead to identification of enablers for ICT integration, as is indeed suggested in the BECTA report [17].

Bingimlas [16] and the BECTA review [17] discuss various approaches to classifying barriers to technology integration in education. The latter study, similar to the one reported on in this paper, is focused at the school level, and success factors and barriers can therefore be usefully classified as either school-level factors or teacher-level factors. These factors can then be summarized as shown in Table 3.

Table 3. Factors influencing technology integration in schools [17]

School-level factors	Teacher-level factors
Lack of time	Lack of time
Lack of access to resources (lack of hardware, inappropriate organization, poor quality software)	Lack of confidence
Lack of effective training	Resistance to change and negative attitudes
Technical problems	No perception of benefits
	Lack of access to resources (personal / home access)

It is not the intention to elaborate further on these factors in this paper. Evidently, schools that are successful in integrating ICT have put in place a number of measures to address these and other barriers. However, in the context of this study the importance of these factors are that the proposed framework should identify actions and interventions where universities can partner with schools to overcome these barriers, and turn them into enablers. Table 4 show exemplars of possible partnership and collaboration opportunities that stem from barriers given in Table 3.

Table 4. Partnership opportunities to address barriers to technology integration

Barrier	Partnership at School Level	Partnership at Teacher Level
Lack of time	Review and develop policies and capacity planning	• Training in time-management skills • Provide support in lesson planning that integrates ICT
Lack of access to resources (lack of hardware, inappropriate organization, poor quality software)	• Provide support in lesson planning that integrates ICT • Support in identifying and developing funding / budget proposals • Assist with developing infrastructure plans	• Assist with identifying appropriate and accessible tools and content relevant to subject pedagogy • Mentoring • Facilitate knowledge sharing and uncovering best practice within the school
Lack of effective training	Facilitate training on technologies, tools, devices, pedagogical approaches, and accessing content resources	Identify and promote the role of lead innovators to provide support to other staff
Lack of technical support	Facilitate training for support staff in educational technology and tools	Place students in schools as first-line support for teachers

Various stakeholders must be considered in completing the proposed framework, as activities / interventions would be designed for specific stakeholder groups:

- Schools as a social environment, and within a societal context (including the wider community where the school is situated).
- Higher levels of the education hierarchy, such as districts, provinces and national government, are relevant in relation to policy and establishment of large-scale initiatives (such as Gauteng Online) that schools would participate in.
- School-level staff, including principals, admin staff, librarians, technology officers, and teachers. It is also important that the role of early adopters and champions of technology integration is recognized as seen in the IDT.
- Learners.

The characteristics of e-schools as described in the White Paper on e-Education [1] are helpful in determining the set of dimensions that were derived for the proposed framework:

- The first set of dimensions relates to creating and sustaining an enabling environment for ICT integration to take place. These include the principal as champion; policy and budget; community participation; knowledge sharing; technical support; staff development; and infrastructure and connectivity.
- The second set of dimensions relate to the areas where ICT must be used in e-schools, viz. teaching and learning; assessment; and management and administration.

Based on the IDT it is also clear that the model must include a longitudinal perspective with regards to the different collaboration and intervention activities that can take place. The needs of schools and stakeholders are different in the early stages of technology adoption than they are in schools where the integration of technology is well established. The three phases that are included in the proposed framework are:

- The school implementation phase, where a conducive and enabling environment must be established,
- The teacher implementation phase, where staff start integrating a basic set of tools into education and administration activities, and
- The sustaining phase, where technology integration has become well established with a significant number of adopters, and experimentation with new tools becomes more viable.

5 Discussion

Based on the above results, this study proposed the draft framework for university partnerships to support technology integration in schools as shown in Figure 1. The framework will be refined based on the findings to be generated as further data analysis takes place.

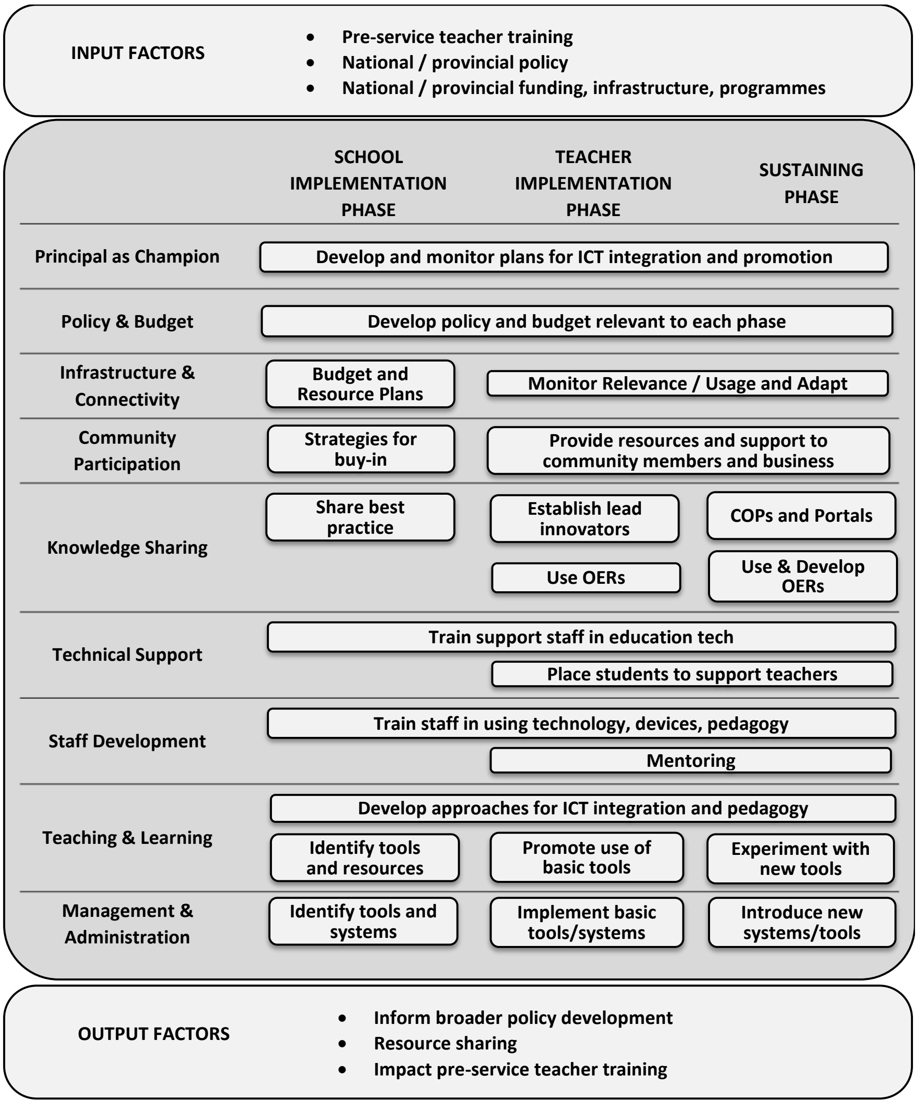

INPUT FACTORS
Pre-service teacher training
National / provincial policy
National / provincial funding, infrastructure, programmes
SCHOOL IMPLEMENTATION PHASE
TEACHER IMPLEMENTATION PHASE
SUSTAINING PHASE
Principal as Champion
Develop and monitor plans for ICT integration and promotion
Policy & Budget
Develop policy and budget relevant to each phase
Infrastructure & Connectivity
Budget and Resource Plans
Monitor Relevance / Usage and Adapt
Community Participation
Strategies for buy-in
Provide resources and support to community members and business
Knowledge Sharing
Share best practice
Establish lead innovators
COPs and Portals
Use OERs
Use & Develop OERs
Technical Support
Train support staff in education tech
Place students to support teachers
Staff Development
Train staff in using technology, devices, pedagogy
Mentoring
Teaching & Learning
Develop approaches for ICT integration and pedagogy
Identify tools and resources
Promote use of basic tools
Experiment with new tools
Management & Administration
Identify tools and systems
Implement basic tools/systems
Introduce new systems/tools
OUTPUT FACTORS
Inform broader policy development
Resource sharing
Impact pre-service teacher training

Fig. 1. Draft framework for university partnerships supporting e-schools

The framework recognizes that university and e-school partnerships take place within an existing content, where national and provincial policy and funding, as well as pre-service teacher training play important roles in promoting system-readiness for ICT integration.

The bulk of the framework is devoted to identifying various activities that can form part of collaboration partnerships between universities and e-schools. By its nature a list of activities could not be exhaustive and the intent here is to describe a range of activities in general terms which can be inclusive of many specific initiatives. It is not intended that all of the described activities are included within a collaboration plan from the outset. Rather, this framework should serve as a guide to developing a collaboration plan that addresses aspects related to the various dimensions, and that the collaboration plan would adapt as it moves through the various stages included in the framework.

The adoption of a structured and planned collaborative approach should be seen to translate to impacts that are broader than the specific school under consideration. Lessons and information gathered could be applied to broader contexts including policy development efforts, funding guidelines, improved resource sharing, and changes to pre-service and in-service teacher training.

Finally, it is worth noting that this paper has not referenced the TPACK model, even though it is a useful (and popular) model for describing the integration of technology in curricula [18]. There are two main reasons for this:

- The TPACK model focuses on contexts specific to curriculum, while this study considers broader factors including policy environment, the school environment, and school activities beyond teaching and learning. As such, the models informing the study were chosen for a broader perspective.
- The TPACK model includes three different contexts as they relate to curriculum development, viz. Technological, Pedagogical and Content Knowledge, as well as the various intersections between the three contexts. The focus of this project is primarily with the Technological Knowledge context, as the disciplinary area underlying this study is the field of Information Systems, with the focus of this project on technology diffusion and adoption (in this case particularly within the schools context). The intent therefore does not include specific, subject-area recommendations for curriculum development, although these could be fertile areas for future inclusion in this framework.

6 Conclusions

The draft framework presented in this paper considers various theoretical perspectives relevant to the integration of ICT into e-schools, including theories of innovation diffusion and technology adoption. The framework also relies on existing policy and literature to derive the set of dimensions, as well as the various stages that must be considered in establishing a university and e-school partnership.

The framework captures the context within which ICT integration takes place through the inclusion of input factors such as national and provincial policy and budget plans, and pre-service teacher training. The context also implies that the process leads to output factors or impacts including the refinement of policy and funding plans, and impacts on teacher training.

The dimensions identified relate to creating and sustaining an enabling environment for ICT integration to take place, and to the various areas where ICT must be used in e-schools. The stages included in the framework recognize the need for a longitudinal aspect that is based on different rates of adoption of new innovations.

The proposed framework is developed based on the South African context, in that an important point of departure is the White Paper on e-Education and the expectations it sets out for e-schools in South Africa (but also the wider continent). However, it is envisaged that the concepts reflected in the framework should be transferable to other educational systems.

7 Recommendations

It is recommended that schools and universities use the framework to guide the inclusion and timing of activities as they jointly develop plans for collaboration for initial introduction, and continued enhancement of ICT integration. Plans for collaboration would be guided by the inclusion of activities related to the various dimensions in the framework. These plans would evolve over time as schools move through the various stages envisaged in the framework, and based on the lessons and experiences from earlier activities.

The focus in this study is on the forming of partnerships between universities and individual schools. However, it must be noted that existing (and future) initiatives identified by universities are naturally targeted towards broader collaboration with a set of schools, such as within a school district. In future this focus could also be included in the framework.

Further research related to this framework could include subject-specific applications / versions of the framework, as well as well as its use for introducing specific technologies into schools. Testing and extending the framework for adaptability to educational systems outside South Africa could also be future topics of interest.

References

1. Department of Education, "White Paper on e-Education: Transforming Learning and Teaching through Information and Communication Technologies (ICTs)," *Government Gazette* , no. 26734, pp. 3 - 46, 2 September 2004.
2. Department of Higher Education and Training, "White Paper for Post-School Education and Training - Building an Expanded, Effective and Integrated Post-School System," *Government Gazette,* pp. 3 - 11, 15 January 2014.
3. New Partnership for Africa's Development (NEPAD), "NEPAD e-Africa Programme," [Online]. Available: http://www.nepad.org/regionalintegrationandinfrastructure/infrastructure/ict. [Accessed 30 September 2015].
4. Ikamva National e-Skills Institute, "South African National e-Skills Plan of Action 2013.," 2013. [Online]. Available: http://www.inesi.org.za/.
5. M. Makgato, "Challenges Contributing to Poor Integration of Educational Technology at Some Schools in South Africa," *Mediterranean Journal of Social Sciences,* vol. 5, no. 20, pp. 1285 - 1292, 2014.
6. T. Vandeyar, "Practice as policy in ICT for education: Catalysing communities of practice in education in South Africa," *Technology in Society,* vol. 35, no. 4, pp. 248 - 257, 2013.
7. I. Sahin, "Detailed Review of Rogers' Diffusion of Innovations Theory and Educational Technology: Related Studies Based on Rogers' Theory," *The Turkish Online Journal of Educational Technology,* vol. 5, no. 2, pp. 14 - 23, 2006.
8. P. S. Hsu and P. Sharma, "A systemic framework for sustaining technology integration in educational settings," *World Future Review,* no. February - March, pp. 41 - 56, 2010.
9. M. Y. Yi, J. D. Jackson, J. S. Park and J. C. Probst, "Understanding information technology acceptance by individual professionals: Toward an integrative view," *Information and Management,* vol. 43, no. 3, pp. 350-363, 2006.
10. U. A. Yucel and Y. Gulbahar, "Technology Acceptance Model: A Review of the Prior Predictors.," *Journal of Faculty of Educational Sciences,* vol. 46, no. 1, pp. 89-109, 2013.
11. P. Surendran, "Technology Acceptance Model: A Survey of Literature," *International Journal of Business and Social Research,* vol. 2, no. 4, pp. 175 - 178, 2012.
12. Y. Lee, K. A. Kozar and K. R. T. Larsen, "The Technology Acceptance Model: Past, Present, and Future," *Communications of the Association for Information Systems,* vol. 12, no. 50, pp. 752 - 780, 2003.
13. M. Chuttur, "Overview of the Technology Acceptance Model: Origins," *Working Papers on Information Systems,* vol. 9, no. 37, pp. 9 - 37, 2009.
14. V. Venkatesh, M. G. Morris, G. B. Davis and F. Davis, "User Acceptance of Information Technology: Toward a Unified View," *MIS Quarterly,* vol. 27, no. 3, pp. 425 - 478, 2003.
15. C. Moersch, "LoTi turns up the HEAT!," *Learning & Leading with Technology,* no. February, pp. 20 - 23, 2010.
16. K. A. Bingimlas, "Barriers to the successful integration of ICT in teaching and learning environments: A review of the literature," *Eurasia Journal of Mathematics, Science and Technology Education,* vol. 5, no. 3, pp. 235 - 245, 2009.
17. A. Jones, "A Review of The Reserach Literature on Barriers to the Uptake of ICT by Teachers," British Educational Communications and Technology Agency (BECTA), 2004.
18. M. J. Koehler and P. Mishra, "What is Technological Pedagogical Content Knowledge (TPACK)?," *Contemporary Issues in Technology and Teacher Education,* vol. 9, no. 1, pp. 60 - 70, 2009.

Using *Minesweeper* to Teach Data Structures and Algorithms: a Problem-based Learning Perspective

Roelien Goede

North-West University, Potchefstroom, South Africa
Roelien.goede@nwu.ac.za

Abstract. The module on data structures and algorithms is typically presented as a final programming module in the second year of a computer science and information systems programme. It aims to integrate programming knowledge and develop efficient and well-structured programming techniques, such as the use of specific data structures, algorithms and the analysis thereof.

Programming is an integral part in preparing computer science and information technology students as lifelong learners for an ever-changing industry. Problem-based learning is an educational approach rooted in constructivism and the ideas of self-directed learning. In problem-based learning students take control over the learning process and ill-defined real life problems are used to guide self-directed investigation to achieve understanding of new concepts.

This paper reflects from a problem-based learning perspective on how the popular game *Minesweeper* can be used to facilitate the learning of some of the concepts covered in the data structures and algorithms second-year module. After a brief introduction to problem-based learning principles, the outcomes of the data structures and algorithms module are linked to specific features of *Minesweeper*. A critical argument is provided, motivating that "pure" project-based learning poses problems in programming education.

Reflection on learning by students is an integral part of problem-based learning. The final sections of the paper presents an analysis of 121 written open-ended questions from an interpretive research perspective. Students reflected on their initial reactions after receiving the assignment, the progress in the individual phases, and their holistic experiences of the problem. From this reflection the researcher identifies aspects to consider in future applications of project based-learning in programming modules.

Keywords: Data structures · Algorithms · Computer science education · Problem-based learning

1 Introduction

The aim of this paper is to demonstrate how the popular game *Minesweeper* can be used to teach some of the concepts required by a module in data structures and algorithms from a problem-based learning perspective.

Problem-based learning (PBL) is a teaching approach that developed in the health sciences during the 1970s (Savery 2015). It is based on the assumption that doing facilitates learning, as expressed by Dewey (1938). It is often viewed as a teaching approach that facilitates the development of students' self-directed learning and lifelong learning skills (Hung et al. 2008; Ben-Ari 2001). These skills are especially important in the information technology industry. Section 2 provides more detail on PBL used in the instructional design of the module on data structures and algorithms.

In PBL, practical problems familiar to the learner are used to facilitate learning. The use of game development has been successful in computer programming education (Papastergiou 2009; Becker 2001; Lawrence 2004; Leutenegger and Edgington 2007; Mathrani et al. 2016). The selection of the "problem" in PBL or the game in game-based learning, is crucial. The specific learning outcomes must be achieved when a student solves the problem (Hmelo-Silver 2004). Programming the popular game *Minesweeper* requires a large number of skills developed in the module on data structures and algorithms. Section 3 provides a description of the main concepts of the module. These concepts can be allocated to features of *Minesweeper* (Section 4).

Section 5 presents the instructional design of the module on data structures and algorithms based on PBL. An integral part of PBL is student reflection. An interpretive written interview was completed by 121 students who were enrolled for the module. Their reflections are presented in Section 6. The paper concludes in Section 7 with reflection and future guidance on the use of popular games or problems to teach data structures and algorithms.

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

The problem addresses by this paper is therefore: How should one evaluate a "game" as problem for problem-based learning in terms of module content, PBL scenario? Although Minesweeper is known to be used in programming education (Becker 2001), this paper demonstrate that it is a good problem in terms of (1) the content of data structures and algorithms (Section 4.2) and in terms of PBL (Section 4.3). The paper also demonstrates how the game may be presented in phases to assist students with different self-directed learning skills.

2 Project-based learning

Most accounts of PBL is found in medical literature, since PBL developed as an alternative to traditional knowledge-first teaching in medical education (Schmidt 1983). The skills required by medical doctors during diagnosis requires integration and prioritization, which can be developed with collaborative learning and the use of scenarios (Schmidt 1983). This section presents PBL as described in literature – although its application, as reported later in the paper, is not pure. The inconsistences between this discussion and the application are identified in Section 5.

The definition provided by Hung et al. (2008) is representative of the accounts of PBL in medical education literature:

"An instructional method that initiates students' learning by creating a need to solve an authentic problem. During the problem solving process, students construct content knowledge and develop problem-solving skills as well as self-directed learning skills while working toward a solution to the problem" (Hung et al. 2008).

This definition supports the characteristics of PBL as discussed by Barrows (1996), who is known for his scholarly development of PBL:

- *"Learning is student-centred – [students takes responsibility for own learning]*
- *Learning occurs in small student groups*
- *Teachers are facilitators or guides*
- *Problems form the organizing focus and stimulus for learning*
- *Problems are a vehicle for the development of clinical problem-solving skills*
- *New information is acquired through self-directed learning" (Barrows (1996).*

Apart from these characteristics, Barrows and Kelson (1995) also accentuate the development of the student's intrinsic motivation to learn.

The nature of the specific problem that should be used in BPL is widely discussed. Hmelo-Silver (2004) describes it as "realistic" and "ill-structured". Lee (1999) argues that although a well-structured problem may be used in PBL, ill-structured projects better reflect the social-constructivist nature of PBL. Therefore, the objectives of the problem should not be pre-determined by instructors. The structure of the problem is influenced by the learning objectives; the students' prerequisite skills and knowledge; the time students have to complete the project; and the specific presentation formats for the solution (Lee 1999). In well-structured problems the procedures are determined by the facilitator. Such problems are helpful to create a basic knowledge domain (Lee 1999).

Wood (2003) provides guidelines for the creation of effective PBL scenarios as summarised in Table 1.

Table 1. Guidelines for the creation of effective BPL scenarios (Wood 2003)

Code ***(Used in Table 3)***	***Description***
Objectives_by_Learner	Learning objectives likely to be defined by the students after studying the scenario should be consistent with the faculty learning objectives.
Appropriate_Curriculum	Problems should be appropriate for the stage of the curriculum and the level of the students' understanding.
Intrinsic_Interest	Scenarios should have sufficient intrinsic interest for the students or relevance to future practice.
Integration	Basic science should be presented in the context of a clinical scenario to encourage integration of knowledge.
Cues_for_Discussion	Scenarios should contain cues to stimulate discussion and encourage students to seek explanations for the issues presented.
Open_Problem	The problem should be sufficiently open, so that discussion is not curtailed too early in the process.
Various_Resources	Scenarios should promote participation by the students in seeking information from various learning resources.

From a constructivist perspective, the prior experience of the student shapes his/her understanding and the material that the learner finds problematic forms the motivation for learning (Savery and Duffy 1995). From our perspective, the role of the module in terms of the curriculum provides the context of the prior experience of the module. The context of this module is discussed in the next section.

Group work and reflection in groups are cornerstones of PBL, since these processes entail communication with peers (Barrows (1996). Students learn from each other and develop the skills to articulate their thoughts and motivations for solutions. Barrows (1996) advocates groups of five to eight members. The practical complications of small groups are evident in large groups of students, as this practice requires many facilitators.

In terms of assessment, Wood (2003) argues that assessment should be done according to the learning outcomes of the knowledge content. Both individual students and groups should be assessed – both formative and summative. The group should be encouraged to reflect on their functioning as a group in terms of communication skills and individual contributions (Wood 2003).

PBL is used from an SDL perspective. Students are in control of their own learning and have to organise their own learning environment (Savery 2015). Grow (1991) argues that the SDL capabilities of students vary and can be developed by means of effective level specific facilitation.

Although PBL is sometimes explicitly linked to programming education (Lykke et al. 2014; Nuutila et al. 2008; Nuutila et al. 2005), the use of problems and scenarios are common (for example O'Kelly and Gibson 2006; Papastergiou 2009; Robins et al. 2003). The presentation of PBL in this paper is from a social constructivist perspective. Although social constructivism is well-documented in science education, it is not often explicitly reported in computer science education (Ben-Ari 2001). Ben-Ari (2001) discusses the metaphysical difference between computer science education and other science education and concludes that constructivism is harder to practice in computer science education, since students find it hard to create mental models of concepts such as object-oriented programming constructs. This paper aims to provide an illustration of how a non-pure version of PBL can be used to teach data structures and algorithms from a social-constructivist perspective, aiding students in creating such mental models.

3 The module in data structure and algorithms

The aim of this section is to present the context of the module in data structures and algorithms. The level of detail provided is guided by the discussion on the applicability of *Minesweeper* as PBL scenario.

A module in data structures and algorithms (DS&A) is part of the Association of Computing Machinery (ACM) standard for computer programming education. It typically covers the following aspects: data structures such as arrays, linked lists, hash structures, queues, stacks, trees and graphs; algorithms for manipulation of the data within these structures; advanced programming skills such as recursion and data-type independence; and analysis of performance of algorithms. It is typically presented in the second year of a computer science programme.

Not all second-year DS&A modules cover all these aspects, as some aspects, such as trees and graphs, overlap with the content of the decision support systems module. The module in DS&A typically follows two semesters of functional and object-oriented programming. Prerequisite knowledge is therefore basic object-oriented programming (OOP) skills in Java or C++, as well as basic functional programming language constructs.

From a constructivist perspective, novice programming students find it hard to create meaningful mental models of programming concepts (Ben-Ari 2001). The focus on the mechanisms (language constructs), especially when starting with OOP, hampers the student’s ability to form a mental model for the logical problem solutions required by programming. Ben-Ari (2001) argues for an initial focus on functional programming before the introduction of OOP, as students are better equipped to form mental models of algorithms than of OOP structures.

The context of the DS&A module at the author’s institution is also affected by the disjointed mental models of the students entering the module. Students have a sufficient theoretical knowledge of programming concepts, but they have not yet internalised these concepts as a tool set that can serve their problem solving skills. Their focus is on the mechanics rather than the role of the concepts in their constructed solutions (programs). The lecturer doesn’t view the content of the module in isolation from the prerequisite content and the module in DS&A serves an integrational purpose to shape the cognitive framework of the programming students. In other words, DS&A helps to form a holistic mental model of computer programming as discussed by (Ben-Ari 2001).

The complexities of teaching programming as identified by Robins et al. (2003) are evident from the decisions related to programming modules at the institution where this research was done and the author concludes that the problems experienced at this institution are not unique.

The module is presented in the second year of a Bachelor of Science programme and requires 160 study hours, assessed with both formative and summative assessment. The author/lecturer uses computer-based practical problems as formative assessment and a hand-written examination as summative assessment. Students are expected to develop complex algorithms in the summative assessment without the availability of a computer to test their algorithms. The class size of about 180 students poses restrictions for the assessment model and the instructional design.

All students are treated as novices at the beginning of their first year, although some had completed secondary school computer science. Some students are self-taught programmers and others have no programming experience at all. These imbalances are still present after two programming modules. The SDL abilities of the students differ greatly as a number of students taught themselves to programme in their high school years.

The selection of *Minesweeper* as problem scenario for DS&A aims at: (1) integrating prior and disjoint knowledge of students; (2) developing the SDL skills of all students; and (3) providing a learning environment for DS&A concepts.

4 *Minesweeper* as scenario for PBL in DS&A

The purpose of this section is to present *Minesweeper* as a scenario for PBL. The operation of the game is first presented, followed by a mapping with DS&A concepts and a reflection in terms of PBL principles.

4.1 Minesweeper

The game *Minesweeper* was distributed as part of the initial Microsoft Windows operating system in 1989. The game presents the player with a matrix of a predefined size containing a fixed number of preplaced “bombs” concealed to the player. Each cell in the matrix either contains a bomb or a value representative of the number of bombs in its eight adjacent cells. The player should then use logical skills to flag a cell as a bomb or to reveal the content of the cell. The game ends when the player has identified all the bombs or mistakenly reveals a bomb. Figure 1 depicts a section of a game in progress.

A specific feature of the game is that when a cell with zero adjacent bombs is cleared, all adjacent cells are revealed as none of them can be bombs. This iteratively applies to the newly revealed cells as well, and therefore a large area is revealed where no bombs are present as indicated by the unshaded cells in Figure 1. By revealing one cell, the user “opens” up a large area with no bombs.

The time it takes for the player to identify all bombs is recorded and a list of shortest time per grid size forms the leader board on a specific computer.

Although *Minesweeper* can be used in education from a solutions perspective (Kaye 2000b, a), the game itself is the focus in this module. The scenario (problem) used in the DS&A module is to recreate the game by using basic Java constructs sensitive to the principles of good OOP programming design. The fact that there are solutions in Java available on the internet, motivated the lecturer to provide some structure to the required solution. Students are allowed to use code from the internet, but they would have to alter so many aspects that it would be easier to write their own code.

It is given as an individual assignment as discussed in Section 5 in order to maximize the benefit to every student. Informal collaboration is expected as it is often present in face-to-face instruction.

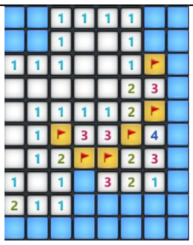


Fig. 1. A section of *Minesweeper* in progress

4.2 Programming concepts present in the recreation of *Minesweeper*

In order to select a good PBL scenario, the required learning concepts should be represented by the solution. In the case of *Minesweeper*, some direction from the lecturer was needed to ensure that students chose the alternative that best represents the learning outcomes. The selection of alternatives was the topic of class discussions as indicated in Section 5.

Table 2 maps the aspects of the game with concepts present in the prerequisite and actual content of the DS&A module.

Table 2. Mapping between *Minesweeper* and DS&A concepts

Minesweeper feature	***DS&A concept***
The graphical user interface (GUI)	Students in the specific module studied GUIs in C# and not in Java. GUIs in Java involves a large number of concepts, such as event listeners, buttons, and layout managers. This is not part of the typical DS&A outcomes. The GUI is used as a final stage in the problem description, given in Section 5.
The cell content	Each cell is an object of a class – constructors, accessors, mutators and general class design knowledge are required.
The matrix containing the cells	A two-dimensional array and the display thereof in terms of the polymorphic display of the cell object value is required. The display of each cell is guided by the state of the game (unrevealed, number, flag, or bomb). In a GUI environment, a data structure containing buttons is required.
Waiting for user input	GUI aspects are used, but alternatively, keyboard activity and the conversion of character data to numbers could be used as array indices are required.
Allocation of cell values	An algorithm is required to compute the digit values of the non-bomb cells. One method of solving this problem is to follow a bomb-centred approach, where all the cell values adjacent to a bomb are incremented when a bomb is allocated. Another method, namely open cell-centred, is to count all the adjacent bombs after their allocation for each open cell. Algorithm design for the manipulation of two dimensional arrays are central to DS&A. Algorithm analysis is performed to compare strategic alternatives (such as using a false border or not)
Clearing open areas	Clearing the adjacent cells of zero values cells can be done by means of different methods, but recursion is intuitive and expected, as it is a central concept in DS&A. A stack can also be used.
Game termination	Functional programming logic is required to manage the flow of the game and its termination when either all the bombs are flag or a bomb is revealed.
Timing of the game	Timing requires an algorithm that determines the length of the game based on the system's time. Students do not have experience in obtaining the system time and doing so extends their mental model of the functioning of the operating system.
High score list	A linked list is used for the best scores (resulting from a class discussion) and file storage is required.
Overall appearance	Various GUI components.

It is evident from Table 2 that not all outcomes of DS&A are required by the solution of the MS problem. Additional problems can be given to facilitate learning of aspects such as stacks, queues, trees, and graphs. *Minesweeper* facilitates the integration of all the programming content of the prerequisite modules with the stated aspects of DS&A. The student should integrate previous and new knowledge. Research into seldom-used aspects, such as the system timer, provides opportunities for development as self-directed learners.

4.3 Reflection on *Minesweeper* in terms of PBL scenario

Minesweeper serves to improve the student's integration and extension of his/her programming knowledge (see Table 2). The aim of this section is to reflect on *Minesweeper* in terms of a typical PBL scenario. From a non-programmers perspective, *Minesweeper* can appear as a well-structured problem. The rules organizing the game are well defined and predictable. The programmer, however, has a large number of alternative strategies to choose from to implement the different features of the game, which makes it ill-structured. Table 2 shows that some alternatives are chosen (in collaboration with students) to ensure the application of desired theoretical components. The reflection is guided by Wood's (2003) principles (Table 1) and are provided in Table 3.

Table 3. Guidelines for the creation of effective BPL scenarios

Code ***(for later reference)***	***Description***
Objectives_by_Learner	Part of the instruction design was the creation of a collaborative chatroom. The reflection of the students (Section 6) showed that although students made an overall mapping, they were not sufficiently able to connect the detailed learning outcomes of the module to the features of *Minesweeper*.
Appropriate_Curriculum	Implementation of *Minesweeper* is appropriate to the stage of the curriculum and the level of the students' understanding. Specific guidance is required to ensure that students choose good strategies. The problems posed by the solution can be solved using their prior knowledge, but the application of the advanced knowledge (module content) is more efficient.
Intrinsic_Interest	The majority of the students had played the game before, and understood the inherent complexities – refer to Section 6.
Integration	The implementation of *Minesweeper* allowed the integration of all prior programming language knowledge. Students had to translate the knowledge obtained in other languages such as C# to Java (GUI components)
Cues_for_Discussion	Although the problem was well-defined, there are different solution strategies with different effectiveness. A key part of the DS&A content is the analysis of algorithms. Different strategies and the analysis thereof are central to the learning outcomes.
Open_Problem	The initial stages of the implementation of *Minesweeper* are well-defined, but the choice to either use a false border or not influences the code in latter stages and can be argued from many perspectives. Students were given total freedom in the latter stages to adapt the standard version as discussed in Section 5.
Various_Resources	Various institutions use *Minesweeper* as programming assignment in different computer languages. Some institutions provide instructional videos on how to implement *Minesweeper*. Although the instructional design (Section 5) is designed to prevent students from copying code, these videos can be used to guide the students. Reflection in terms of the available resources is provided in Section 6.

5 Instructional design

One of the key differences between problem-based learning and project-based learning is the scope of individual assignments. In project-based learning, students typically work on a single project throughout the entire module. The project guides them to reach all learning outcomes of a module (Helle et al. 2006). In problem-based learning the scope of an individual problem is narrower, therefore one problem is used together with others to guide the students to reach the desired outcome (Mills and Treagust 2003). The implementation of *Minesweeper* formed 20% of the examination entry mark for the module in DS&A discussed here. The other 80% was achieved by solving smaller weekly problems focused on linked lists, stacks, queues and trees. The allocation of study hours for *Minesweeper* was a maximum 35 hours. The introduction of the problem was done in a workshop environment with the following actions:

1. Students had to play the game on their smart phones – this was for some a first introduction to the game. The rules of the game are simple and the second-year IT students typically take less than 15 minutes to master the game and they enjoy playing it.
2. Students are then guided to discuss the aspects of the game in terms of programming in small groups of three students each. Although formal direction is given later, it is important to allow the students to first reflect on the programming aspects of *Minesweeper*.
3. Students also had to reflect on the division of the problem into feasible phases in small groups.
4. Students were encouraged to play the game in their free time.

The SDL skills and programming experience of the individual students differ. Some students are relative novice programmers with only two modules in Java completed, while other students have three to five years of programming experience, often self-taught (high SDL). The novice programmers (that DS&A caters for) find the problem overwhelming (refer to Section 6). It is more efficient to be prescriptive at the beginning of the project in terms of foundational data structures used and naming these structures when aiming to provide continuous support to a group of 200 students.

Providing an initial structure achieves three key outcomes:

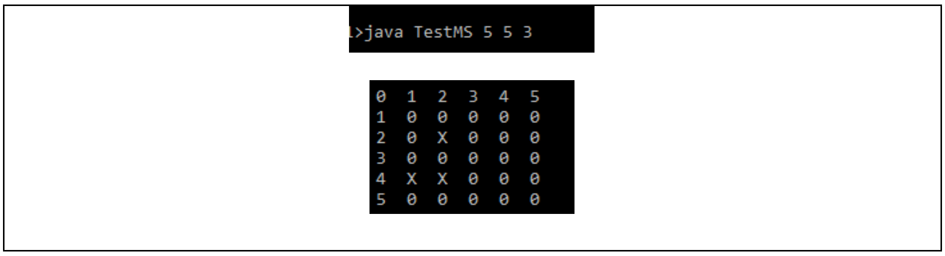
1. Standardised guidance can be provided after each phase of the project in terms of a suggested solution.
2. Available solutions from the internet must be adapted to the prescribed structure, requiring deep understanding of the available solution.
3. Class discussions on different strategies can be based on the initial structures and all students can take part in the discussion.

After the initial workshop the lecturer presented *Minesweeper* as a problem to be solved in FOUR phases. The freedom of the students in terms of the solution increases in each phase.

5.1 Phase 1: Grid structure with OOP

Initially the students created a non-GUI solution. This strategy was identified by students during the initial workshop. Students did not have prior experience of GUIs in Java and they identified the use of GUIs early in the process as a stumbling block. The intention of the first phase is to link the solution to programming knowledge of OOP and arrays and to extend the application of prior knowledge to create a two-dimensional array of objects. Variable names of the attributes of the cell object and the minefield is given AFTER a collaborative discussion in class on the required attributes.

Students were required to allocate bombs in the grid with a random number generator. Random numbers are not in the frame of reference of all students, but the level of guidance available on the internet is high. To ensure that the difficulty of the problem is at the correct level and that students integrate new knowledge, a changeable grid size is used. The implementations had to use command line parameters for the size of the grid and the number of bombs. Figure 2 shows the desired output for the first program.



```
l>java TestMS 5 5 3
```

```
0  1  2  3  4  5
1  0  0  0  0  0
2  0  X  0  0  0
3  0  0  0  0  0
4  X  X  0  0  0
5  0  0  0  0  0
```

Fig. 2. Output of Phase 1

A rubric was used for assessment of the problem. Assessment was done according to the OOP constructs.

After the due date of the first assignment, a workshop followed, focusing on the solution of Phase 1. In practice, very few students had problems with Phase 1 (refer Section 6) and they found the suggested solution easy to understand as most of the knowledge required was prior knowledge. From a PBL perspective, students gained confidence and overcame the initial resistance to the implementation of *Minesweeper*.

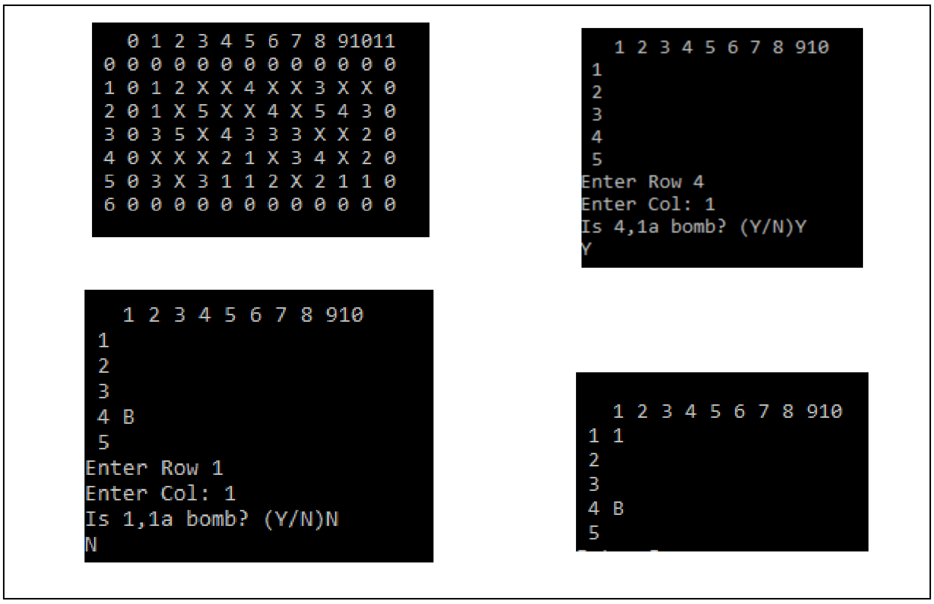
5.2 Phase 2: Cell values and user input

The second phase required algorithm design skills, as indicated in Assignment 2:

1. The value of each cell had to be computed to represent the number of adjacent bombs;
2. A second grid concealing the values had to be displayed;
3. A game loop had to be implemented that controls the game in reaction to user input;
4. A user chosen cell had to be revealed if it did not contain a bomb;
5. The player had to be able to flag a cell as a bomb;
6. The game had to terminate according to *Minesweeper* rules.

The first item on the list requires careful algorithm design in terms of “out of bounds” errors. The frame elements do not have eight adjacent cells to check and should be treated as special cases. The facilitator can expect a large number of questions regarding the handling of frame elements. Students quickly realized that the frame elements affect all the algorithms. A simplification of the implementation is to add a “ghost frame” of zero values around the grid to eliminate the role of boundary values. Most novice programmers chose this option, while more experienced programmers in the class preferred to handle the frame as special cases. A class discussion followed to compare these methods in terms of the module outcomes on method analysis. Since formative feedback was aimed mainly at the less experienced students, the solution distributed after the admission date implemented the “ghost frame”. Figure 3 indicated the output of Phase 2, including the “ghost frame”.

When feedback was given after submission, care was taken to link the components of a long program to the elements of the question and the theoretical concepts they implement. At this stage students experienced satisfaction as they can “play” the game with their own program.



```
  0 1 2 3 4 5 6 7 8 91011
0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 2 X X 4 X X 3 X X 0
2 0 1 X 5 X X 4 X 5 4 3 0
3 0 3 5 X 4 3 3 3 X X 2 0
4 0 X X X 2 1 X 3 4 X 2 0
5 0 3 X 3 1 1 2 X 2 1 1 0
6 0 0 0 0 0 0 0 0 0 0 0 0
```

```
  1 2 3 4 5 6 7 8 910
1
2
3
4
5
Enter Row 4
Enter Col: 1
Is 4,1a bomb? (Y/N)Y
Y
```

```
  1 2 3 4 5 6 7 8 910
1
2
3
4 B
5
Enter Row 1
Enter Col: 1
Is 1,1a bomb? (Y/N)N
N
```

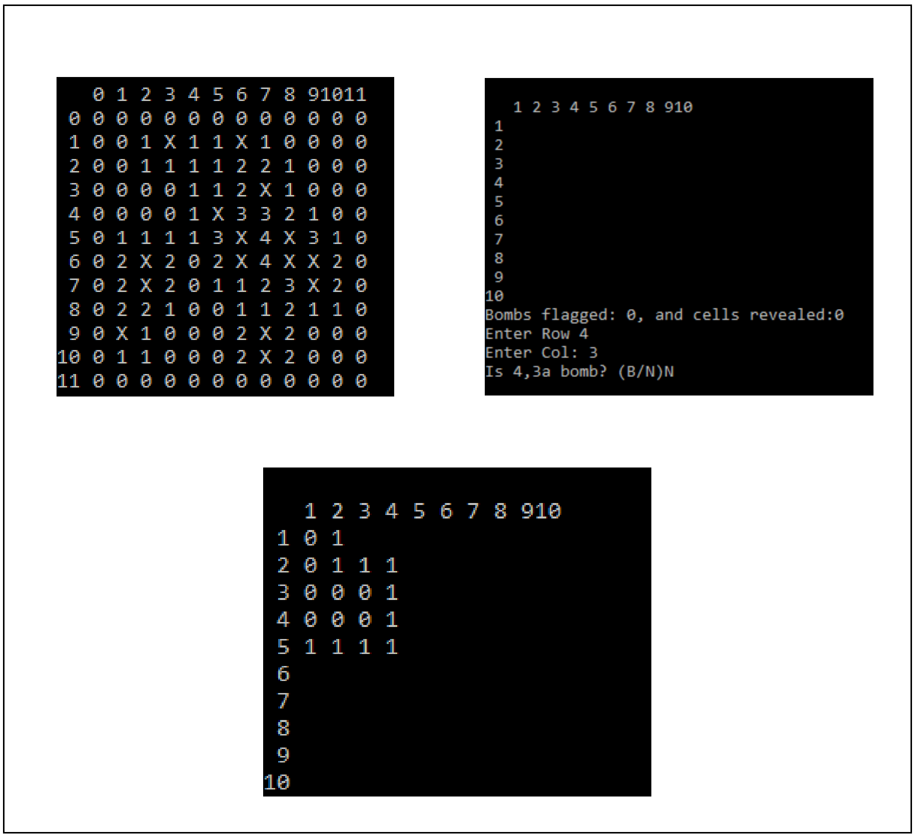
```
  1 2 3 4 5 6 7 8 910
1 1
2
3
4 B
5
```

Fig. 3. Output for Phase 2

5.3 Phase 3: Zero values

When a player reveals a cell with zero adjacent bombs, all the adjacent cells should be revealed as well, also revealing the adjacent cells of the newly revealed zeros. This assignment was given before the theory of recursion was presented in a class session. The more experienced students attempted to solve the problem with loops. Although it is possible, recursion is a more stylish solution. In the third assignment, students were instructed to solve the problem using recursion. Recursion is typically presented in theory a few days after the assignment has been given to students. Very little reference to *Minesweeper* is made in the theoretical discussion. Figure 4 shows the successful solution of Phase 3.

A few students used stacks to implement the solution. This opened up a discussion on the theoretical link between stacks and recursion. Students started a discussion in the collaborative chat room that was used as the basis for a class discussion.



```
   0 1 2 3 4 5 6 7 8 91011
 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 1 X 1 1 X 1 0 0 0 0
 2 0 0 1 1 1 1 2 2 1 0 0 0
 3 0 0 0 0 1 1 2 X 1 0 0 0
 4 0 0 0 0 1 X 3 3 2 1 0 0
 5 0 1 1 1 1 3 X 4 X 3 1 0
 6 0 2 X 2 0 2 X 4 X X 2 0
 7 0 2 X 2 0 1 1 2 3 X 2 0
 8 0 2 2 1 0 0 1 1 2 1 1 0
 9 0 X 1 0 0 0 2 X 2 0 0 0
10 0 1 1 0 0 0 2 X 2 0 0 0
11 0 0 0 0 0 0 0 0 0 0 0 0
```

```
   1 2 3 4 5 6 7 8 910
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
Bombs flagged: 0, and cells revealed:0
Enter Row 4
Enter Col: 3
Is 4,3a bomb? (B/N)N
```

```
   1 2 3 4 5 6 7 8 910
 1 0 1
 2 0 1 1 1
 3 0 0 0 1
 4 0 0 0 1
 5 1 1 1 1
 6
 7
 8
 9
10
```

Fig. 4. Output for Phase 3

5.4 Phase 4: Free range – the GUI

After the assessment of Phase 3, students were given the code of the total solution up to this point and it was explained to them. In Phase 4, students were given an open assignment to convert the black-screen program to a GUI using Java. The use of GUIs in Java is not an explicit learning outcome, but it does requires careful algorithm planning and data structure development in terms of an array of buttons representative of the playing grid. Students were able to achieve a pass mark without the implementation of the GUI (by implementing the first 4 items on the list). In order to manage the time spent on the problem a flexible assessment list was provided as part of the assignment, as shown in Figure 5. This list resulted from a class discussion. Note the addition of an undo function, which is not logical within the rules of *Minesweeper*, since it allows a player to rethink a move, which would normally terminate the game. The students enjoyed the opportunity to be in the design role of the game – they were able to redesign the rules of the game and had fun proposing improbable features.

***Minesweeper* Assignment 4**

Now you are free to choose what you want to do to improve the player's experience:
The following options are available and the marks achievable with the option is indicated in brackets:

- A pop-up input screen without a GUI (10%)
- A timer with best times saved of best scores using a LINKED LIST (30%)
- Save the game to continue later (25%)
- An undo function (20%)
- A fixed-size GUI at least 10 x 10 (30%)
- A user-changeable size GUI (30%+20%= 50%)
- A running timer on a GUI (20%)
- A restart button on a GUI (10%)
- Own idea (verify the marks you will achieve before you start!)

You have to demonstrate your program!

A rubric representative of the marks in Figure 5 was used to assess individual demonstrations. Students were required to explain their programs and all code had to follow the naming conventions of Phase 1. This was the final assignment in the semester and it was not necessary to give students a standard solution as the items on the list did not contain new concepts that are part of the DS&A module content (except for the linked list). The assignment did, however, develop their integration and algorithm design and implementation skills. The choice between an array and a linked list for the best times guided the students to critically reflect on the merits of each of these data structures.

The students had three weeks, which included one week recess, to work on Phase 4. During this time, the students were asked to anonymously reflect on their experience in terms of a written interview completed voluntarily. Analysis of this interpretive data is provided in Section 6. **Fig. 5.** The assignment for Phase 4

6 Reflections of students

A set of questions were given to the students to reflect on their experience. Although 121 of the 183 enrolled students completed the written interview, many gave short answers without the explanation requested in the questions. The aim of the questions were to guide reflection by the students on the learning activity and to understand their overall experience. The interview had three main sections. Section 1 referred to the initial experience; in Section 2 students were asked to reflect on each assignment in terms of the aspects they found easy and difficult; and Section 3 focused on the holistic experience after completion of the first three assignments.

The questions were completed anonymously during a face-to-face class session where the only other activity was to write a small weekly test on binary trees. The lecturer/researcher/author was not present and students were asked by an assistant to complete the questions. A total of 121 students handed in the questions, of which eight only answered the first of the two pages (printed back-to-back). The timing of the written interview was three days after completion of the third assignment, two weeks prior to the final demonstration.

The data were coded using content analysis from an interpretive research paradigm perspective as discussed by Klein and Myers (1999). The answers were mostly cryptic, but useful information could be derived by comparing all the answers from a particular student to understand whether the student had a positive or negative experience. From a hermeneutic perspective, data were coded in a spreadsheet with the data of one participant per row and all the answers for a specific question per column. Rows and columns were analysed individually to gain an understanding of the experiences of individual students and of the group. The answers per participants sometimes had logical discrepancies, as unexpected answers were given. As an example, one student indicated that he was nervous to start the project, then reported difficulty on the first three assignments and then he/she indicated that the fourth assignment would be easy (while all the other students indicated that they have no experience with GUI's in Java). After indicating that the fourth assignment would be easy, the student said his/her overall view was that the problem is too difficult for second-year students. This instance may be a case where the student felt so whelmed that a humorous/sarcastic answer was given.

Entries with discrepancies were isolated and later analysed again to ensure consistency and to minimize errors.

Allocated codes were not mutually exclusive: a code of "excited" and "positive" could be assigned to the attitude of one student, depending on the specific answer. The result is that the total codes for questions on attitude add up to more than the 121 participants.

Additional data from the demonstration of the fourth assignment is added to the analysis. It is not possible to map this data to specific participants of the first interview, but trends are reported.

6.1 The initial reactions of the students

The aim of the first section of the written interview was to better understand the initial reaction of the students. Of the 121 students who completed the interview, 105 indicated explicitly that they had played the game before, the rest had not (14), or did not complete the question (2 students).

The students were asked how they felt after the initial announcement of the problem. When the answers are analysed mutually exclusively with the options of "negative", "positive" or "nervous", 13 was negative, 50 positive and 51 nervous, with seven answers that could not be analysed: one student used the word "indifferent" and another simply said it is "a very easy problem to solve".

A number of "nervous" students gave reasons for their answers: Seven students indicated that they "did not know where to start" and two said that they were not good programmers. Of the "positive" students, 15 answers were coded as “looking forward” and 20 as “excited” (these codes are not mutually exclusive). One of the “negative” students stated that he/she already solved the problem, another said that “this should not be practical module” and two others said they “expected last year’s problem”.

Overall it can be said that 45% (50/114) of the group was positive, 44% (50/114) nervous and 11% (13/114) was negative, which highlights the need for a motivation when the initial problem is given.

6.2 Progress with the individual problems

The written interviews were conducted after the submission date of the third problem (focusing on recursion) and two weeks before the fourth and final problem (focusing on the GUI). Students were asked on their success in terms of the first three assignments, which corresponds with the discussion in Section 5: 108 students indicated that they had successfully completed the first assignment; 107 said they had successfully completed the second assignment and 88 said they had completed the third assignment.

The students were asked how they handled problems they encountered. A total of 28 reported on the role of their friends in solving problems, six referred to the practical assistants and only one referred to the lecturer. A total of six students said they studied theory on recursion in their text book to solve their problems with the third assignment. The majority (68) said they used the internet to solve their problems.

The references to the internet revealed that students used the internet for three different purposes. Some students studied available code on *Minesweeper* in order to use the code. Although the answers are too vague to specifically assign the purpose of using the internet, three students said the available code differed too much from the expected code to be useful. A second use of the internet was to find documentation on concepts such as recursion and GUI development. A third use was YouTube instructional videos on GUI development. Assignment 4 required the students to integrate GUIs in their problems and they had not previously implemented GUIs with Java. A future project will explicitly investigate the purpose of internet usage.

It should be noted that the author is suspicious of a number of students who did not answer the question or who reported that they did not use the internet. These students might have been under the impression that it would cast them in a negative light.

Students were asked to express their feeling on Assignment 4 (the GUI refer Fig. 5) as well. Again feedback was mutually excluded coded as “positive”, “negative” or “nervous”. Of the 112 answers that could be coded, 51 were “positive”, 25 “negative” and 36 “nervous”. The majority of students gave the same motivation: “Never done GUI in Java before”. “Positive” students gave replies such as “I’m looking forward to it as I have never done GUIs in Java before”, while “negative” students said: “I’m not excited since I’ve never done GUIs in Java before” with similar answers by “nervous students”. Four students referred to time constraints. As discussed before, some suspicious (sarcastic) answers were identified.

Of the 180 students initially registered for the module, 144 demonstrated Assignment 4. There were 12 students with valid excuses and the rest were students who dropped out. Of the 144 who demonstrated their project, 130 demonstrated a GUI and 14 chose to do options on the assignment list that did not require a GUI. It is not possible to say how many of the “nervous” or “negative” students did develop a GUI, but from the numbers it is evident that many of them did demonstrate a GUI. As discussed earlier, a GUI in Java as such is not part of the main module outcomes, but the implementation of a GUI in Java demands knowledge of DS&A and therefore falls within the scope of the module.

6.3 Overall student perspective

The ability of the students to identify the learning outcomes is key to successful PBL (Wood 2003). Students were asked whether *Minesweeper* is a good assignment for the module in DS&A. Three questions gave the student the opportunity to reflect on this matter. First, a direct question was asked: “How do you think *Minesweeper* fits the learning outcomes of the module: DS&A?”, secondly: “Do you think it is good practical work for the module? Why?” and later “Do you think I should give something like this again?, Why do you think so?”

Unfortunately many students gave short answers, 91/121 students answered that there is a good fit between *Minesweeper* and the module outcomes. Fourteen students referred explicitly to “recursion” in the answer on the fit with the outcomes. Two students said that it is good to incorporate prior knowledge into a problem. Forty-two students answered that the problem reflects data structures and algorithms. Only two students gave a substantial list of items covered by the problem. Only six students referred to the absence of linked lists in the earlier assignments. Eight students referred to positive aspects concerning the longer duration of the problem. Other assigned codes included “I understand Java better now”; “different form text book problems”; “nice to do larger problem”; “good integration of algorithms” and “good learning opportunity”;

Seventy-eight students said that it should be repeated. Four students said that it was “too difficult” and should not be repeated. Three students said that a problem that better incorporates linked lists should be given and one said that previous answers will be used and a new problem is required. One of the students who answered positively said that it is good mixture of algorithms and data structure, it forces creativity and it tests programming ability.

In order to understand the attitude towards the problem, a question was added on whether they told anyone outside the class about the project. Fifty-two students said they told their family and/or friends about the project. One wrote: “I told and showed it to everybody who wanted to listen.” Eight other students told stories of how they mailed it to family or friends (even overseas) to play and test.

A question on the role of the problem in building programming confidence had mixed results. Seventy-one students report a growth in confidence, while 17 gave a negative response and eight were reserved on the matter. Only six of the 17 substantiated their answers and all referred to either a lack of guidance or a lack of freedom. One student said that his/her own Java knowledge was lacking and that other students enjoyed the project.

A question on which part of the project the students enjoyed most also gave unexpected results. The most common answer (20) was achieving working code. Twelve said the GUI part was most enjoyable and 14 referred to the recursions. Thirteen said the first two problems were the most enjoyable; 11 said they enjoyed everything; nine said playing their own games was most enjoyable; two students referred to collaboration with friends. Only five students said they did not enjoy the assignment at all (stating “lack of guidance” and “too difficult” as main motivations for their answers).

In support of the view that different students have different SDL skills and the need for different levels of support (Grow 1991), students were asked how they made use of the suggested answers they received after each phase of the problem. Half the group (62) indicated that they did make use of the answers. Codes allocated to narrative answers included: “good structure”; “see how it can be done”; and “helped a lot”. Very few students gave reasons on why they did not use the code. If this answer is compared with the second group of questions on which assignments they submitted successfully, it is apparent that roughly 15 students had trouble with the assignments and did not use the suggested answers. Two wrote that the answers were too far removed from their understanding and therefore of no help. Another two complained of the fixed structure and “his/her own style”, referring to the lecturer.

A few reflections follow below based on the analysis of all the data provided by the students:

- Students are nervous when confronted with new work. Some see it as a challenge and some are worried that they won’t be able to complete the assignment.
- The impression is that they are not used to the concept of PBL. They understood that they were supposed to learn by doing, but this made many students more nervous than excited.
- The majority enjoyed the process and were able to link the overall outcomes to the problem.
- Time management and timely support is crucial.

7 Summary and reflection

Minesweeper can be used in conjunction with other small assignments for DS&A teaching. It integrates prior knowledge with new knowledge of specific data structures and challenging algorithms. Students enjoyed the challenge of creating a real-life application, in this case a known computer game.

When applying “pure” SDL, some problems arose. The most important shortcoming relates to the nature of programming as a discipline. Students can present a working version of a program, without implementing good programming design techniques, even without being aware of the existence of such techniques. More guidance than “pure” PBL advice should be given to ensure that students learn and apply good programming techniques. The provision of a structure for a solution also ensures that students are not able to copy available code without understanding it.

Another problem is the collaborative nature of PBL. Often when students work together on programming assignments, one student gains more confidence than the other, mainly because only one program is written and one student takes control of the debugging. Informal collaboration occurred in this project, but since students were required to demonstrate and explain their solutions, most of the students took ownership of “shared” attempts.

Programming students come from different skills sets and different SDL levels. This also implies different levels of readiness for PBL. Some students took a strong grade-oriented approach, posting questions on the collaborative chat room on the number of marks they will lose if a specific feature is not working. A process-orientation, however, is required for deep learning in PBL. Students need to be patient with themselves and should develop a methodology for problem-solving. The lack of such a methodology is evident from the large number of negative responses to the question on their attitude towards the fourth assignment.

E-learning platforms proved successful in providing different levels of support. Dividing the problem into smaller problems for submission enabled the lecturer to provide intensive support without influencing the assessment of the students. Students experienced a sense of renewed confidence each time they used the suggested answer of the previous phase.

Upon reflection, the following is suggested for the future use of PBL:

- An introduction of PBL will be given at the beginning of the project, focusing on the fact that students should strive to link parts of the problem to the learning outcomes of the module.
- Students should be guided in the process-orientation of PBL. This entails the development of own strategies on how to identify and gain knowledge required to solve a real life problem.
- The selection of the problem is crucial. Students enjoyed the game nature of *Minesweeper*. An investigation should be done on the availability and structure of solutions on the internet.
- Structured outcomes, in terms of project divisions and strategy, should be compulsory after the initial workshop. Each student should submit his/her suggestion on the division of the problem and the required programming structures. This should be assessed. There was a lack of ownership in this group in relation to the phases and the structure provided.
- Along with suggested answers for phases, the lecturer will provide a mapping of aspects of the specific phase and the outcomes of the module.
- Clear motivation will be provided at the beginning of each phase for the prescribed use of specific techniques.
- In terms of understanding student reflection, distinctions will be made on the use of the internet to understand whether tutorials on concepts were used or available solutions for the chosen problem.

A final conclusion is that PBL is a suitable teaching/learning approach with many advantages, however one cannot take the students' readiness for PBL for granted. Students must develop PBL skills, similar to developing SDL skills. Students should be encouraged to adopt a process-orientation to learning as opposed to a grade-orientation. It would be very satisfying if all students indicate that they are "excited because they do not know the content" when asked about their attitude towards a new phase of a PBL problem.

The short answers provided by the students in the written interview had a negative impact on the richness of the data and therefor the understanding of the lecturer. Future research should employ a different data collection process, such as a focus group or a written interview after students completed the module. This paper demonstrated that minesweeper is a suitable PBL scenario for data structures and algorithms on the basis of the technical content and the requirements of PBL scenarios. There is an opportunity to develop more classical games listed by (Becker 2001) in this regard.

References

Barrows H, Kelson A (1995) Problem-based learning in secondary education and the problem-based learning institute. Springfield, IL: Problem-Based Learning Institute 1 (1):1-5

Barrows HS (1996) Problem-based learning in medicine and beyond: A brief overview. New directions for teaching and learning 1996 (68):3-12

Becker K (2001) Teaching with games: the minesweeper and asteroids experience. Journal of Computing Sciences in Colleges 17 (2):23-33

Ben-Ari M (2001) Constructivism in computer science education. Journal of Computers in Mathematics and Science Teaching 20 (1):45-74

Dewey J (1938) Experience and education: the Kappa Delta Phi lecture series. Toronto: Collier Books,

Grow GO (1991) Teaching learners to be self-directed. Adult education quarterly 41 (3):125-149

Helle L, Tynjälä P, Olkinuora E (2006) Project-based learning in post-secondary education–theory, practice and rubber sling shots. Higher Education 51 (2):287-314

Hmelo-Silver CE (2004) Problem-based learning: What and how do students learn? Educational psychology review 16 (3):235-266

Hung W, Jonassen DH, Liu R (2008) Problem-based learning. Handbook of research on educational communications and technology 3:485-506

Kaye R (2000a) Infinite versions of minesweeper are Turing-complete. Manuscript, August

Kaye R (2000b) Minesweeper is NP-complete. The Mathematical Intelligencer 22 (2):9-15

Klein HK, Myers MD (1999) A set of principles for conducting and evaluating interpretive field studies in information systems. MIS quarterly:67-93

Lawrence R (2004) Teaching data structures using competitive games. IEEE Transactions on Education 47 (4):459-466

Lee J (1999) Problem-Based Learning: A Decision Model for Problem Selection.

Leutenegger S, Edgington J (2007) A games first approach to teaching introductory programming. ACM SIGCSE Bulletin 39 (1):115-118

Lykke M, Coto M, Mora S, Vandel N, Jantzen C Motivating programming students by Problem Based Learning and LEGO robots. In: Global Engineering Education Conference (EDUCON), 2014 IEEE, 2014. IEEE, pp 544-555

Mathrani A, Christian S, Ponder-Sutton A (2016) PlayIT: Game Based Learning Approach for Teaching Programming Concepts. Educational Technology & Society 19 (2):5-17

Mills JE, Treagust DF (2003) Engineering education—Is problem-based or project-based learning the answer. Australasian journal of engineering education 3 (2):2-16

Nuutila E, Törmä S, Kinnunen P, Malmi L (2008) Learning programming with the PBL method—experiences on PBL cases and tutoring. Reflections on the Teaching of Programming:47-67

Nuutila E, Törmä S, Malmi L (2005) Pbl and computer programming—The seven steps method with adaptations. Computer Science Education 15 (2):123-142

O'Kelly J, Gibson JP RoboCode & problem-based learning: a non-prescriptive approach to teaching programming. In: ACM SIGCSE Bulletin, 2006. vol 3. ACM, pp 217-221

Papastergiou M (2009) Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. Computers & Education 52 (1):1-12

Robins A, Rountree J, Rountree N (2003) Learning and teaching programming: A review and discussion. Computer science education 13 (2):137-172

Savery JR (2015) Overview of problem-based learning: Definitions and distinctions. Essential readings in problem-based learning: Exploring and extending the legacy of Howard S Barrows:5-15

Savery JR, Duffy TM (1995) Problem based learning: An instructional model and its constructivist framework. Educational technology 35 (5):31-38

Schmidt HG (1983) Problem-based learning: Rationale and description. Medical education 17 (1):11-16

Wood DF (2003) Problem based learning. BMJ: British Medical Journal 326 (7384):328

A General Purpose Computer Lab with Linux and Virtual Machines

Petri Jooste

School of IT, North-West University (Vaal Triangle Campus),
Vanderbijlpark, South Africa
petri.jooste@nwu.ac.za

Abstract. This is a practical experience report of the design and implementation of a new computer lab. The computer lab of 180 PCs is mainly used as a lecture room for Computer Science and Information Systems courses but its purpose has been extended to meet the needs of other subjects as well. A new approach was taken by using a lightweight Linux host operating system on which VirtualBox is used to run Windows, Linux and Android in virtual machines. Problems experienced in other computer labs were addressed and our solutions with their advantages are described with reference to some of the technical detail of various open source software programs that was used for this purpose. These include ssh, Clonezilla, uftp, overlayfs and the Epoptes computer lab management and monitoring tool. This report is written after a full year of successful operation.

Keywords: University computer lab, Linux, VirtualBox, Clonezilla, uftp, overlayfs, Epoptes

1 Introduction

In the past all teaching of Computer Science (CS) and Information System (IS) subjects on our campus was done in the general classrooms and computer labs provided and maintained by the campus IT Services department. This began to change a few years ago when the School of IT was given a computer lab (30 PCs) to manage and maintain independently. This lab was utilised for courses especially Operating Systems and Computer Networks in which the practical work was of such a nature that it could not be accommodated in the general campus labs. Basically it was set up to dual-boot Linux and Windows and lecturers could install software themselves. A master PC was set up and cloned to all the rest of the PCs. If a PC became unusable, one or more of the partitions could be cloned from a working PC. Because the lab was shared between courses, a lab assistant was appointed to manage and maintain it. Some lecturers wanted software configurations which remained unchanged and available for their classes and did not appreciate broken configurations that sometimes resulted from the experimental usage. Fortunately the split was mostly between the Windows and Linux users which meant that the Windows partition was supposed to remain stable while the Linux partition was more experimental.

As student numbers grew the lab was moved to a new venue and upgraded to 50 PCs. Gradually more courses of the School of IT, including advanced undergraduate Statistics courses, moved to the self-managed lab. Most of the CS and IS modules were however still using general classrooms for lectures and general computer labs for practical work because of class size.

With student numbers on the whole campus growing and more subjects starting to request computer labs for e-assessment, the pressure was to move all subjects of the School of IT out of the general labs. An additional computer lab with 180 PCs was then planned for the School of IT.

This paper describes the new lab with regard to (a) the design choices and how they addressed problems experienced by CS and IS lecturers in other labs (b) some technical details of the implementation and (c) an evaluation by the lab managers and users.

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

2 Design Choices

2.1 Keep what works in other classrooms and labs

2.1.1 Presenters desk

The installation and maintenance of projectors and loudspeaker systems connected to a presenters desk / podium is done centrally for the whole campus and this seemed adequate for our needs. There is however a big difference between current lecture rooms and labs in that the presenters desk in a lecture room is much bigger and provides devices in addition to a standard PC with (mostly) appropriate software. These include a document camera and audio system with wearable microphone and extra batteries on charge. Since we wanted the new computer room with 180 PCs to function equally well for lecturing and practical work one of the more extensive presenters desks was installed.

2.1.2 Cloning from a master PC

We used the open source Clonezilla software [9] to load all PCs with the same software. To do this a master PC is set up with all software installed and configured and then a complete disk image is copied to our Clonezilla server. The partitions and boot loader can be individually accessed and distributed, or the complete disk image can be utilised. It is called casting because it uses the efficient method of broadcasting or multi-casting from one sender to several receivers at the same time to make the best use of network bandwidth.

2.1.3 Desks with network connections

The desks for the student PCs was of a new design that would become the new standard on campus and we also used the same supplier of locks and cables for physical security. Network cabling was also done in the standard way. This was different from our smaller 50 PC lab where we were in control of the network cabling and switches, but for the time being it seemed that we could operate without that level of control. This turned out to cause a problem for casting images and we had to use a workaround.

2.1.4 Stable host OS with virtual machines

The dual-boot configuration was still in use in our 50 PC lab, but the experimental nature of the Linux partition was changed. Both Windows and Linux installations were kept stable and protected from permanent changes by students. Empty partitions and unallocated space on the hard disks could be used by students to install and maintain their own operating systems and they could also use VirtualBox [7] in both Windows and Linux to create virtual machines (VMs) there. The use of virtual machines was something that worked well and would become one of the building blocks of the new lab setup.

To keep the installation of Windows stable it was protected by software called Shadow Defender [8]. This protective software layer loaded when Windows booted. All changes would appear to be real but would in fact be made in a temporary storage space and upon reboot this was simply discarded. A similar solution was implemented on the filesystem level in Linux by fsprotect [1]. These solutions worked well during normal use, but there was a weakness in maintaining protection and stability of the dual-boot Windows and Linux systems. Students could boot from removable media to install new operating systems in the unallocated space and this meant that they had access to a very low level and by accident (or on purpose) they could damage the existing partitions or boot loader. Of course we could repair it by simply casting the saved disk images with Clonezilla, but as the content of the partitions increased it was taking longer to cast a complete image or even just one of the partitions.

In the new lab the students are not allowed to boot from removable media. All work that requires them to practice low level configuration such as disk partitioning is done by using virtual machines only.

2.2 Addressing problems experienced in other labs

2.2.1 Power failures

Anecdotal evidence (cursing or crying in the corridors) shows that users of computer labs experience power failures as one of their big frustrations. Our campus invested in powerful emergency generators since the country-wide load-shedding of some years ago, but the problem for a computer is that it needs sustained power. The School of IT requested centrally managed UPS systems for the computer labs during that time, but it never materialised. For that reason we (already in our smaller lab) fitted each PC with a small UPS so that a student has time to save current work and it hopefully keeps the computer running until the campus generator can restore the power supply.

Another big impact of power failures on the use of computer labs is that the network switches, which are typically not on UPS's, go down and network connectivity is lost. This created a problem when our tests were written in general labs not under our control. Before a test the lecturer requested the appropriate number of restricted Novell exam-ids from IT services. These were given to the students at the start of the test and provided the mapped network drives where files are saved for submission. The exam-id allows only one login to help prevent dishonesty and this meant a student could not continue with the test after network connectivity had been lost.

2.2.2 Login access

In the general computer labs on campus a student can only start working after logging in with his/her own credentials. At the start of the year, especially with first year students, there are various reasons why many students cannot log in and the lecturer has no control over the situation. It has the effect that students (and ultimately lecturers) suffer from lost class time and falling behind. Especially in CS and IS courses this has a cumulative negative effect. Therefore it is desirable to have a lab which gives the lecturer control to allow guest logins, so that students can start learning-by-doing from day one.

2.2.3 Improving the process of adding or changing software

The standard practice to load software in computer labs is to install and configure one master PC and then clone it by creating an image of the whole hard disk. The image is then transferred (cast) to all PCs in the target lab. The image usually runs without problems on PCs with matching hardware, but when lab computers are replaced in batches per lab, it could imply that several master PCs have to be maintained. Since the process of casting images can take long and requires the target labs to be completely unavailable for users, the *casting is not done often*. The process also carries a *risk*, because if it does not complete successfully the affected PCs cannot be used at all. For these reasons the IT services department updates the software in the general labs only twice per year. Requests for new or updated software must be made several months before the start of the relevant semester.

The *problem* here is that lecturers experience it as an inconvenience that software changes have to be given long in advance and cannot be changed during the semester. When we only had a single lab with 50 PCs, we afforded our lecturers the luxury of casting images much more frequently to add or update software or make configuration changes. However, we realised that it may not be a sustainable practice when more and larger labs need to be maintained.

An improved process for software changes should be faster, less risky and not disruptive. It should also allow more frequent changes as needed.

Virtual machines running on each lab PC can be used to give students full administration privileges while practising their server, network, security or database administration skills (for example by using the Linux operating system). An alternative approach is to host the virtual machines elsewhere [5].

3 Design and Implementation

The list of requirements for our new labs with proposed or implemented solutions to the problems mentioned above will now be discussed under the following headings:

1. Resilience: power and/or network failure
2. Submit system for CS/IS tests and exams

3. Efficient software installation and maintenance
4. Balance between stability and flexibility
5. Smooth and fast operation
6. Control by lecturer / presenter / invigilator

3.1 Resilience

When a power failure or network failure occurs during a lecture or test, the computers should keep working and still provide access to as many of the required resources as possible. For example the following should still be possible: login; use of software; access to given templates, datasets or skeleton code; saving and submitting of files.

3.1.1 Power failure

The primary measure taken here to supply UPS power to each of the 180 machines will allow students to save frequently and continue until the campus generators kicks in or power is restored. Currently all computers including the presenter's desk have UPS power, but the ceiling-mounted projectors should also be included in future. Emergency lights are installed and although they are very faint, the light from all the computer screens provide enough light for the lecture, practical class or test to continue.

3.1.2 Network failure

All software and files should be available on the hard disks of the PCs and student submissions should be done on the PC hard disks, at least in the form of a backup submission system. In future the need for the lecturer to send control commands from the podium will determine if it justifies the cost to provide the set of local switches with UPS power. This is currently not a priority since power is usually restored within 15 minutes.

3.2 Submit system

All PCs are set up to work as stand-alone workstations with all software installed on local hard disks. A *submit* folder is always available on the desktop and students are instructed to use that as working folder when writing tests. There is also a *given* folder (read-only to students) where files can be placed by the lecturer - typically before the start of the test or class.

Using the local submit folder as the working directory during tests and exams has several advantages for programming and other CS/IS courses. Advantages for lecturers or markers include (1) being able to recreate the exact path and directory structure on the marker's computer; (2) file ownership and permissions are retained; (3) better support for version control systems or development environments which use hidden files or folders. Advantages for students are knowing that the latest version or changes made by the student will be the version submitted for marking and there is no need to follow a manual and error-prone process of copying or zipping the files for submission when the time runs out at the end of the test or exam.

For the lecturer to collect the submissions after the test three mechanisms are provided. The first mechanism is a script that runs before the student logs out. The script checks if there is anything in the submit folder and asks the student to select the subject code and provide a student number while displaying the tree with all files that will be submitted. The student provides the information and allow the submit process to begin or cancel it to go back and make changes. The submit script uploads the files to the PC in the presenters desk. (As a backup measure the files remain on the hard disk of the student PC.) After the last student submitted, the lecturer simply creates a compressed archive of all submissions and copy it to removable media or cloud storage before leaving the lab.

The second and third mechanisms rely on the way that each PC handles a login session and the automatic nightly backups to cloud storage. During each student login, the submit folder of the previous session (if not empty) is automatically timestamped and moved to the backup location on the PCs hard disk. Then the rest of the old session is cleared to a state ready for the next student with files in the given folder still in place. The design is such that even files from a session which ended abruptly (e.g. power failure) so that the submit-script of the first mechanism did not run, will still be collected in the regular backup processes.

Instead of waiting for the automatic nightly backup the lecturer can also use a command on the podium to collect the submissions for a specified date from the backup location of each PC. Before doing this the lecturer usually issues a command to do a fake login on all PCs to ensure all submissions have been moved to the backup location.

3.3 Efficient Software installation and maintenance

Just as with our existing lab, Clonezilla software [9] [2] is used to cast the initial image prepared on a master PC. The design of this initial image (partitions, permissions etc.) is such that it allows for more efficient ways of installing and changing software than having to cast a complete image in future. Of course there will always be an up to date complete image on our Clonezilla server that can be used for example when a PC needs to be replaced due to hardware problems.

Not having complete control over the network in the new lab provided us here with a challenge. For Clonezilla Server Edition to work it is a requirement that the target PCs should boot over the network by getting instructions from the Clonezilla server. However, the network did not forward the boot requests to *our* Clonezilla server but to the server used to cast images to all the other labs on campus and we had no control over this. For the first image casting we had to either use our own temporary network cabling or boot each PC from a USB disk with Clonezilla Live and manually let it join a cast session from our own server. Clearly this was only a once-off solution and only required for the initial deployment. We have developed a different way of deploying software changes to the whole lab using virtual machine images and snapshots which does not use Clonezilla. Also, after doing the first full image cast in an unusual way, we implemented a workaround for potential future use of Clonezilla. Since it is open source we modified the files of Clonezilla Live and placed them on each PC's hard disk so that we could instruct the local boot loader (Grub) to boot with Clonezilla Live. Clonezilla Live is given parameters to find our Clonezilla Server to join a cast session. This process can be orchestrated by issuing commands with ssh to all PCs.

3.3.1 Virtual machines

Instead of setting up a dual-boot system, we decided to use a single lightweight Linux host operating system and provide one or more virtual machines for Windows, Linux and Android with application software as needed by different subjects. VirtualBox was chosen as the hypervisor. The ability to take snapshots of virtual machines made it possible to develop a faster way to update software on all PCs. Instead of distributing a whole disk image, disk partition or even virtual machine image, it was possible to just send the virtual machine snapshot with the differential disk files from the master PC to all PCs.

3.3.2 uftp - casting of files instead of partitions.

Clonezilla and other image casting solutions use network bandwidth efficiently because one sender sends to all receivers simultaneously. However, Clonezilla is designed to transfer complete disk partitions block by block. If most of the files on the partition stays the same and only some files need to be changed, deleted or added then casting the whole partition actually wastes a lot of time. Moreover, if something goes wrong with the process the partition is unusable and it has to be overwritten completely by initiating the cast process again from the start.

We use uftp [4] to transfer large files in a similar bandwidth efficient manner[6]. Files specified in a list are sent block by block from one sender while being received and written to disk by all participating receivers simultaneously. In ideal circumstances each block of data is sent only once from the server. In practice some blocks are missed by some clients but the protocol allows for this and only the missing blocks are sent again until all receivers have all files. This process is enabled by the fact that the files are received and placed into an existing filesystem. This happens while the host OS is running and the filesystem is mounted, consequently there is no downtime during transfer. When the transferred files have to be added to the virtual machines, it requires only a few minutes of downtime during which users cannot be logged in. Files are moved into position and the PCs are rebooted to remount the affected partitions so that the added files take effect.

3.3.3 Overlay filesystem

The ability to use overlay filesystems is built into the Linux kernel [3]. We use it in the following way. Two separate physical partitions are formatted with the ext4 filesystem. The one is named and mounted as *lower* and the other as *upper*. All the directories and files that belong in the home directory of the student user is saved in *lower*. At this time *upper* is empty and is mounted as an overlay on *lower* and the resulting filesystem is then made available as the apparent home directory of the student user (e.g. /home/student). When the student user now logs in all files in *lower* are available. If new files or directories are created they are physically stored in *upper* but appear to be with the existing files in *lower*. If an existing file is deleted it is only flagged as deleted in *upper* so that it is not available when viewed via the overlay /home/student. If an existing file is modified it is automatically first copied to *upper* and then modifications are made there.

When a new student login session starts, all that has to happen is that all items (files and flags) in *upper* are deleted, so that the new session starts off with /home/student looking exactly like the pristine *lower*. Then when the student starts using the Windows VM, only the file of the last snapshot will be copied from *lower* to *upper*. This file is typically less than 100 megabytes in size when copied but then it grows as the student uses the VM. All changes such as pages loaded in a web browser or files copied to the Windows desktop or even new software installed by the student will be added to this file. VirtualBox automatically integrates this new contents with the contents from the VM files which it transparently accesses from *lower*.

The only files that are kept during a new login are those in the *submit* folder in the Linux desktop of the student user. This submit folder is mapped by VirtualBox as a shared folder available to die Windows guest OS and a link on the Windows desktop is also named *submit* and points to the shared folder.

3.4 Balance between Stability and Flexibility

We maintain an identical set of stable virtual machines on each PC. All changes to a VM is discarded at the end of a login session of the host operating system. The new session provides the user with VMs which are identical to those prepared on the master PC. With VirtualBox in place any user can also create new virtual machines which can be rebooted and experimented with during a login session of the host operating system.

3.4.1 Stability

Two virtual machines (Windows 7 and Windows 10) were set up by the lab assistant at the end of 2015 according to the combined needs of lecturers that needed stable environments. During 2016 there were nine updates to the Windows 7 VM and two updates to the Windows 10 VM. Each update simply meant that the new files corresponding to the new VM snapshots were transferred to all the PCs using uftp. Of these files, the smallest was about 1 GB and the largest about 10 GB. After each update the PCs were in a new stable state and die VMs worked correctly when the PC rebooted.

3.4.2 Flexibility

The lecturer for Databases set up his own virtual machine where students could practice administration tasks. This complete VM was also sent to the PCs using uftp. For Operating Systems the lecturer was happy to use the Linux host operating system for some exercises and having the ability to let students experiment in virtual machines that they could set up themselves.

3.5 Smooth and Fast Operation

3.5.1 Login sessions

All students log in to the Linux host OS using a local user account. A guest session mechanism is also available on Linux, but currently disabled, because we address all our current needs with the single user account. This saves valuable class time because it gives students immediate access to use the computers without login problems such as forgotten passwords. The computer lab is actually used as a classroom fitted with computers. On the class timetable almost all sessions are allocated which means that the room is always used under supervision. If there is no class, the room is locked to prevent unauthorised access.

3.5.2 Get a running start

One concern about using virtual machines is that it may cause software to run slower because not all hardware resources are available to the guest OS. We anticipated this issue and therefore chose to keep the resource use of the host OS very low by installing a lightweight Linux desktop environment. The virtual machines are then set up to be given at least 75% of RAM and CPU cores. Hardware acceleration features are also enabled.

Ironically it was the use of VM snapshots that made our lab appear *faster* for users. The reason is that the frustrating slowness in other labs (especially expressed by students when writing tests) is mainly due to long start-up times. To illustrate this, we can compare the sequence of steps that need to happen before a user can begin to use an application in our lab with the sequence in the other labs.

The average times mentioned below were measured on machines with comparable hardware: HP All-in-One PCs with 8GB RAM and 500GB HDD with only the following difference in CPUs: Traditional Windows-based lab has Intel Core i5-4590S 3.0 GHz. New Linux-based lab has Intel Core i5-4570S 2.9 GHz. The similarity in CPU performance can be seen from the following reported "Passmark" values: 6949 (i5-4590S) vs. 6682 (i5-4570S) which is a difference of less than 4%.

The comparative measurements below give an indication of how much time the user spends waiting for the computer to perform actions from the start of a session until the application program can be used. For this reason, and because it can vary greatly, the times for user actions are not reported here. It also does not show differences in booting times of Linux in our lab and Windows in the other labs because when students want to start a session, the computers are usually already booted and waiting at their respective login screens. The version of Windows used in both labs is the same - Microsoft Windows 7 Enterprise 64-bit. Two different application programs were used to represent different courses - the first is MS Visual Studio for a programming course (C#) and the second is MS Excel used in Introductory Computing, Decision Support Systems, Accounting and several Statistics courses. All measurements were taken in first sessions after booting.

A session in our Linux based lab involves the following steps: 1) Login with a generic local user account which loads a lightweight Linux desktop. 2) Double-click a desktop icon to launch a Windows virtual machine. 3) Double-click the application icon. 4) Use the application. 5) Minimise or close the virtual machine window. 6) Logout from Linux

In a Windows-based lab the following typically happens: 1) Login with a Novell username and password. The login script runs and pauses with a dialog. 2) The user accepts the rules of conduct and the Windows desktop is loaded. 3) Double-click the application icon, e.g. MS Excel, Visual Studio or Firefox. The application takes long to load because it needs to load several components, including various plug-ins, from disk into RAM. 4) Use the application. 5) Logout from Windows and Novell.

The times in table 1 are averages given in minutes and seconds, rounded to the nearest second. Only the computer time at steps 1, 2 and 3 is reported after the user input was given at each step until the user can take the the next step. In step 3 the two applications were measured in separate first sessions after booting.

Table 1: Times that user is required to wait

Step	Windows 7 only	Linux with Win7 VM
1) Login	03:13	00:04
2) Windows desktop	01:03	00:26
3) Application: Visual Studio	05:46	00:03
3) Application: Excel	00:39	00:18

Notes on the performance in the Linux based lab: At step 2 the virtual machine resumes from a running snapshot by loading into RAM the complete state of a Windows guest OS which was captured in a running state. Windows does not need to go through a boot process. It is almost like waking up from a hibernated state. At step 3 the application opens quickly because the VM snapshot was taken after these applications had already been opened and closed and therefore Windows already have their core components loaded and initialised in RAM. No logout or shutdown of the Windows VM is required before step 6 (end of the Linux session) because the state of the Windows VM makes no difference, since a new Linux login will discard everything (except the submit folder) of the previous session.

3.6 Control by Lecturer / Presenter / Invigilator

Giving students immediate access with a common password removes barriers for access. However, the lecturer also needs control over utilisation of the computers to ensure that the computers are not a distraction or disruption to the class. To get central control over PCs which have been set up to operate independently (see the discussion on resilience above) we implemented the possibility to issue commands via ssh from the podium PC to all student PCs on the level of the Linux host operating system. We also installed the software Epoptes [10] to give specific control functionality to the lecturer.

3.6.1 During lectures and practical work

There may be times when the lecturer wants to block computer usage to get the students' full attention, Epoptes can be used to temporarily lock all (or a selection of) PCs by displaying a lock screen and disabling keyboard and mouse input. Other remote commands include forced logout, shutdown, reboot and boot. Epoptes can also be used to monitor students. It displays and updates thumbnail images of all connected client PCs to give an overview and the lecturer can then select an individual PC to monitor full screen or control via VNC.

Simple bash scripts have been written to offer file transfer options to lecturers. Rsync initiated over the ssh protocol is used to distribute files for practical work. This can be done before the class or test starts or even when it is in progress. For large file transfers there is the option of using the more bandwidth efficient uftp software

3.6.2 During tests and exams

From the podium the lecturer or invigilator can disable login and force logout on all PCs and when the test should start, login can be enabled. Depending on the type of test or exam the lecturer may want to restrict network access completely or just allow access to a specific server such as the university's learning management system (LMS). Although we do not have control over the network switches and routers, the ssh mechanism can be used to change the firewall rules on all PCs to achieve the desired network restrictions. The submit-system described above is another example of how the ssh protocol is used for collecting submitted files during and at the end of a practical test or exam.

An additional feature is that lecturers have the option of preparing two variants of a practical test. These are then distributed to odd-numbered and even-numbered PCs respectively. Students are thus discouraged from copying what they can see on neighbouring screens because it does not match the values of the questions they have been given.

4 Future work

4.1 Login access

Currently the Linux lab can be used by anyone without login restriction because it assumes supervision is present. The lecturer's PC on the podium however needs a password not known by students. The podium itself is locked and unlocked by swiping an authorised personnel card. If the lab should be available in future for unsupervised sessions it would be better to restrict access during those times to users with a valid username and password.

4.2 Automation

More automation can be done to manage and even deploy a new computer lab from scratch using open source software such as Ansible or Puppet.

4.3 Container-based architecture

The potential benefits of container-based software architecture can be investigated. Some aspects of the design may benefit from this type of architecture. Lecturers may also want to include it in the curriculum of some courses.

5 Conclusion

When the project to set up a new university computer lab was started in 2015 there was only a few months to get it ready for the next year's classes which started in February. This paper was written after a full year of successful operation. It documents some of the key design considerations and implementation details. The lab runs on Linux as host operating system with VirtualBox as hypervisor for virtual machines in which Windows and other operating systems provided a diverse set of subjects with the required applications. Feedback from users and comparative time measurements show that the virtual machine approach gives advantages for reducing user waiting times during session and application program startup. The virtual machine approach also gives flexibility for the configuration and maintenance of software which benefits lab technicians and lecturers. Other software components that are key to the design are ssh, Clonezilla, uftp, overlayfs and the Epoptes computer lab management and monitoring tool.

References

1. fsprotect Ubuntu package (2014), http://packages.ubuntu.com/trusty/fsprotect
2. Bowling, J.: Clonezilla: build, clone, repeat. Linux journal 2011(201), 6 (2011)
3. Brown, N.: Overlay Filesystem, https://git.kernel.org/cgit/linux/kernel/git/ torvalds/linux.git/tree/Documentation/filesystems/overlayfs.txt
4. Bush, D.: UFTP - Encrypted UDP based FTP with multicast (2017), http://uftp-multicast.sourceforge.net/
5. Hardaway, D., Hogan, M.J., Mathieu, R.G.: Outsourcing the university computer lab. Computer 38(9), 100–102 (2005)
6. Nuxref: UFTP - Mass File Distribution Using Multicasting (2014), http://nuxref.com/2014/10/01/mass-file-distribution-using-multicasting/
7. Oracle: VirtualBox (2017), https://www.virtualbox.org
8. shadowdefender.com: Shadow Defender (2016), http://www.shadowdefender.com
9. Shiau, S.: The clonezilla project. http://clonezilla.nchc.org.tw and http://clonezilla.sourceforge.net (2008)
10. Tsamis, F., Georgopoulos, A.: Epoptes - an open source computer lab management and monitoring tool (2017), http://www.epoptes.org

Contributing to the Teaching-Learning Process through the use of Competitive Programming: Experience of School of Computing at the University of Namibia

Ruber Hernández-García[1;2], Yeleny Zulueta-Veliz[1;2] and Tulimevava K Mufeti[1]

[1]School of Computing, University of Namibia, Namibia
[2]Faculty of Computational Sciences and Technologies, University of Informatics Sciences, Cuba
{rhernandez,yzulueta,tndakunda}@unam.na

Abstract. Programming is an essential area of study across the various bodies of knowledge and careers of the computing discipline. Nonetheless, it has proven to be a learning challenge to many students, particularly at the introductory phases. One of the main research areas of computing education is therefore, that of improving teaching and learning of programming. In the School of Computing at the University of Namibia (UNAM), programming is considered to be one of the most challenging and difficult courses to teach and learn. The School is therefore continuously exploring new ways to encourage and motivate students to learn and achieve the required programming skills. Competitive programming is one of the approaches that was recently explored for its ability to improve self-efficacy and motivation of students. This paper presents the work done to prepare UNAM students for competitive programming, from initial preparations to participation in local and regional competitions. Implementing competitive programming as a support to the Teaching-Learning process of Programming has achieved positive preliminary results. This paper demonstrates one way in which researchers took advantage of opportunities to encourage students to overcome barriers in the area of programming.

Keywords: Learning Process, Competitive Programming, ACM-ICPC.

1 Introduction

Programming is a foundational and cross-cutting knowledge area of all Bodies of Knowledge (BoK) of the Computing discipline. It also serves as a fundamental competency for the various careers associated with Computer Science. The possible approaches that can be used to teach programming, both in terms of content (what to teach) and teaching method (how to teach), are very diverse. Teaching of programming has thus often presented a challenge to educators at all levels of education, and researchers are continually searching for strategies to improve the teaching and learning process. At the same time, learning programming is also considered difficult: students have to adapt to a more abstract way of thinking to solve problems algorithmically. For these reasons, the process of teaching and learning programming considered as one of the major challenges of computing education[27].

Usually, within Computer Science undergraduate studies, educators structure programming courses for large and heterogeneous groups of students. This makes it difficult to achieve meaningful forms of personalized teaching, where the educator can ensure that the student has acquired the concepts and attained the desired learning outcomes. It further prevents the development of advanced skills among the more knowledgeable students. Applying traditional teaching methodology in the field of programming is further reported to lead to demoralization of students and loss of interest in the topics covered in the course [29]. It is therefore imperative for educators to find ways that benefit the teaching and learning of programming courses.

Researchers have proposed the use of several methodologies to increase the motivation and commitment of students in an educational environment [16, 42, 20, 29]. There seems to be a general consensus that the use of e-learning applications can break barriers in the programming courses. Using incentives is also reported to result into greater commitment and acceptance by students [34, 7, 25]. This is under the assumption that the human being feels more motivated when his effort is rewarded and noticed in some way by others.

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

The School of Computing at the University of Namibia (UNAM) is striving to be a regional Centre of Excellence in developing computing knowledge and expertise that is appropriate for transitioning Namibia into a knowledge-based economy. Its mission is to equip the graduates with requisite skills in computing that will enable them to start professional careers that will enable them to succeed despite the fast evolving technology and discipline. The School has thus designed a strategy aimed at continuously improving the curriculum and upgrading of skills, to ensure that its graduates attain the expected level of mastery to output inventive products and carry out innovative research that will benefit the national and regional contexts.

Over the past few years, the School of Computing at UNAM has also experienced poor performance and high failure rates among its programming related courses. The School is therefore exploring ways to encourage and motivate students to learn and attain programming skills. Competitive programming is one of the approaches considered [4, 29, 32, 21] in improving self-efficacy and motivation of students. In competitive programming, students participate in a competition to solve a well-defined problem according to provided specifications. The student should be well versed in the general topics and problems of computer science, optimization strategies, as well as in the mathematical concepts that are required for Computing students. Students who participate in competitive programming are reported to have shown better results in related subjects, as each programming challenge involves dedication and dexterity. In addition, participation in competitive programming is also known to develop teamwork skills.

In 2016, the School of Computing had more than 160 students attending modules that are directly related to Computer Programming. Prior to this intervention, students at the School did not have much exposure to programming contests, whether internally arranged or externally organized. For this reason, one of the first goals of introducing competitive programming was to facilitate the participation of our students in the ACM International Collegiate Programming Contest (ACM-ICPC), as a way to promote and motivate students towards programming, and at the same time improve student outcomes in these subjects.

This paper describes the entire process followed in the preparation of our students at the University of Namibia to enable them to participate in the ACM-ICPC. It explains how the students were introduced to competitive programming, from the beginning to the participation in local and regional competitions. It also presents the preliminary results achieved in this exploratory phase, as well as the new opportunities that enabled students to work together to face challenges and overcoming barriers traditionally associated with programming.

The rest of the paper is organized as follows. Section 2 presents related work on the use of competitive programming to support teaching and learning of programming. Section 3 goes deeper into the world of competitive programming and its application in the context of teaching programming. Section 4 introduces the characteristics of the competitive programming program implemented at UNAM. Preliminaries results of the program are given in Section 5. Section 6 concludes the paper.

2 Related Work

Teaching programming requires integration of several disciplines, to ensure that students acquire logical and mathematical skills required to solve problems algorithmically. There are several approaches proposed in literature for teaching programming. Several authors [39, 31, 28, 35, 41] have reviewed the teaching strategies most used in teaching and learning programming. These include problem-based learning, game-based learning, collaborative learning, example-based learning, and project-based learning. From these strategies, many models have been designed and applied in programming courses [40, 9, 23, 30, 32, 36, 3, 37].

One of the most forceful techniques in recent years has been competitive programming. It aims to improve the skills of students to enable them to develop algorithms based on strategies like problem-based learning and collaborative learning. Other researchers [15, 29, 11] have presented their experience with competitive programming in their own universities and institutions. In these cases, the researched made use of an online judge to evaluate algorithms, promote competition and make programming courses more interesting and challenging for students. This approach is also reported to increase student's interest in actual learning content.

A succinct overview of programming competitions is presented by Forisek [12]. In the overview, Forisek highlights the most common and what is considered to be the most important aspect of most traditional programming contests: that of designing efficient algorithmic solutions. He further argued that the focus of contents should be more than the design of efficient algorithms, suggesting instead that it should be comprehensive enough to include the various aspects of problem solving. Coles et al. [5] presented a methodological guide that can be used for the analysis and resolution of problems appropriate for programming contests. Their focus is more on 'how' the problems should be solved rather than in the results. Ragonis [33] explored the large variety of well known problems within the computer science discipline, and discusses the extent at which these problems can be used for different teaching situations and processes.

Using online platforms to support the teaching of programming has become a widely accepted practice in research literature [15, 29, 13, 11]. Some researchers have developed automated programming systems that are aimed at minimizing the complexities of available Integrated Development Environments (IDEs), thereby lowering barriers to programming. Others have also dedicated their efforts to the development of automatic submission and assessment systems of programming code produced by the learners. In most cases however, automatic assessment was successfully implemented, but it did not result in paralleled good quality feedbacks.

Our approach at UNAM is based on the ideas discussed by Hassinen [18]. Hassinen argued that there is no shortcut to learning to program, but that students can improve their skills through extensive practice achieved over a sufficient period of time to become familiar with programming concepts. It is this idea that is advanced in this work. We therefore wanted to experiment with contests, combined with other activities that expose students to programming, while developing algorithmic and self-assessment skills in a fun and challenging way.

3 Context of Competitive Programming

Competitive programming is a methodology that aims to create environments that allow students to compete with each other through solving programming exercises [29]. It enables students to write programs to solve a series of specified problems and to test such solutions with test cases that validate the correct implementation of the involved algorithms. Castillo et al. [4] refer some important objectives of competitive programming, such as:

- Encourage and raise awareness about the importance of algorithm problems and their resolution in an effective and efficient way.
- Encourage self-improvement of students through solving different problems with different algorithm solutions.
- Promote the study of programming, data structures and algorithms among students.
- Help students to develop coding and programming skills.
- Constitute teams of students interested in international programming contest.

These objectives have also been validated by other researchers [34, 8]. Currently there are several competitions around the world, held for both secondary and university students. The objective of each competition may vary, and can include: testing participants' knowledge, improving learning, promoting computer skills and finding talented students.

Due to its relevance and international prestige at tertiary education level, out target and focus at UNAM has been that of participating in the ACM-ICPC contest. The International Collegiate Programming Contest is a well-known event that has been in existence since 1970. It quickly gained popularity as an innovative challenge amongst the top performing students in the United States and Canada. To date, the contest involves a global network of universities across the world, hosting regional competitions that advance teams to the ACM-ICPC World Finals. Participation has grown to several tens of thousands of the finest students and Faculty in Computing disciplines at almost 2,736 universities from over 102 countries on six continents. The contest fosters creativity, teamwork, and innovation in building new software programs, and enables students to test their ability to perform under pressure. Put simply, it is the oldest, largest, and most prestigious programming contest in the world [2].

The ACM-ICPC is a multitier, team-based, programming competition. It consists of solving problems that pass certain test cases. The participating teams are provided with 8 to 12 problems, and have five hours to solve it. Groups of 3 participants and one machine per team are allowed. For each program that passes the test cases, a specific score is assigned, then the time it takes for a team to solve each problem is taken as a penalty when there are teams with the same number of problems solved. The statements of the problems are always in English, which forces the participating groups to be fluent in English. Moreover, language plays a relevant role since teams need to fully understand the problem to code an algorithmic solution.

The competition is organized among teams of students representing institutions of higher education. Teams first compete in local, national, and then progress into regional contests held around the world each year. The winning team from each regional contest qualifies to advance to the ACM International Collegiate Programming Contest World Finals. The organization can invite additional high-ranking teams to the World Finals as wild card teams.

The South Africa Regional Contest represents the geographic area of Southern Africa and West Africa. The regional contest brings together university students from different countries including Angola, Benin, Burkina Faso, Cameroon, Ethiopia, Kenya, Mauritius, Niger, Nigeria, Togo and South Africa. Prior to 2016, students registered at UNAM had not participated in programming contests, let alone the ACM-ICPC.

Competitive programming is an approach that is now widely adopted as a teaching academics in recent years [16, 25, 15, 29, 32, 21]. Several universities have implemented courses of algorithms and programming based on ACM-ICPC exercises [26, 14, 19], which promote competitiveness. They use different programming languages to achieve the desired academic performance [25, 10, 29]. The are many benefits to the adoption of competitive programming as support within a course of programming. For example, the teacher can perform an automatic qualification, which can save time to improve other areas of teaching. From the student's perspective, it allows them to have get their performance grade immediately, to keep up statistics of progress throughout the course, and to receive immediate feedback. This demands rigor and enforces discipline on the part of the student.

There are several, freely available websites where students can download programming problems and present their solutions to be evaluated automatically. Using these freely available resources, students can train in advance, before participating in open programming contests. Among the most used and recommended resource sites are: ACM-ICPC Live Archive, UVa Online Judge, Code-forces, CodeChef, and Caribbean Online Judge. Some universities have also implemented their own practice platforms such as Programming Practice Center [29] and EduJudge [13].

4 Competitive programming program implemented at University of Namibia

In 2016, the School of Computing offered four undergraduate qualifications: Diploma in Computer Science (DCMP), Bachelor of Science in Information Technology (BSCO), Bachelor of Science in Information Technology (BSIT), and Bachelor of Science in Information Systems (BSIS).

The different programming related modules [1] in these three qualifications are shown in Table 1 below.

Table 1. Structure of Programming Discipline in the different programs at School of Computing.

Year	Course	Qualification
First Year	Programming Principles Programming Fundamentals I Programming Fundamentals II	DCMP BSIT / BSCO / BSIS BSIT / BSCO / BSIS
Second Year	Programming I Programming II Object Oriented Programming I Object Oriented Programming II	DCMP DCMP BSIT / BSCO / BSIS BSIT / BSCO / BSIS
Third Year	Data Structures and Algorithms Web Design and Programming	BSCO / BSIS BSCO
Fourth Year	Artificial Intelligence	BSIT / BSCO

The School of Computing decided to organize the participation in the ACM-ICPC as a way to motivate students and at the same time raise student outcomes in programming related courses. The organization and implementation of Competitive Programming was carried out with the assistance of two visiting professors with some experience both in participation and the organization of programming contest in their work at University of Informatics Sciences, in Cuba.

An open invitation call for voluntary participation was posted on the School notice boards and class announcements. In total, 21 students participated, 13 from second year, and 8 from third year. No first year or final year student responded to this call. Interested students were asked to choose and group themselves into three-members teams. Although the call was for any student of University of Namibia, only students from the School of Computing participated.

The actual preparatory program for the competition was organized in weekly meetings as well as individual study by students. The design of the program took into account four fundamental principles that, according to research [14, 15, 32], should be present in every course of competitive programming. These are:

- Motivation, which refers to the desire to learn new concepts and methods, and put them into practice.
- Active learning, where students are involved and are made aware of their own learning processes. This ensures that students are not just spectators, and is necessary to achieve long-term learning
- Independent work, where students practice to solve algorithmic problems individually and create their own programs.
- Feedback of the learning process: where continuous evaluation and feedback of the progress is given to students. Through the use of online judges, it is possible to continuously feedback the process.

From these principles, it was decided that the best way to carry out the program of competitive programming would be to use a teaching - practice - competition strategy described below:

- Theoretical classes: face-to-face sessions, where the professor presents concepts, demonstrations and algorithms that are key to help students understand a certain subject and which will serve as a basis for solving or dealing with a problem in an efficient way.
- Programming practices: the student tests the knowledge acquired during the lectures, as well as his personal study and interaction with his fellow students.
- Local Contests: a local programming contest is organized among students. Exercises in local contests are based on topics which the students have already covered. This way, students get a feel of how much more time they need to dedicate to practice and personal study and to measure how well they did in relation to other participants.

This teaching - practice - competition strategy has several objectives. One of them and perhaps the most important is to generate friendship, cooperation and competition between students. This is in order to provide them with a high problem-solving ability, facilitate easy communication between them, a high level of preparation in programming, and also as training for attending international programming contests.

Cuba-Ricardo et al. [6] explored different learning strategies that programming contestants use. This study was the result of the experience in the topic through systematized interviews and observations by coaching of contest groups. They concluded that the development of contestant's skills depends on the learning process of contents during training sessions, and during competitions. He therefore concluded that influencing the learning of each contestant is a key goal for coaches.

In an effort to modify the contestants learning process, the following actions proposed in [6] were adapted and supported in the methodological work developed during the program:

1. Diagnose the particular skills and learning strategies of the students.
2. Determine the relationship between learning strategies used by students, and the activity of solving programming problems.
3. Address individual differences considering the diagnosis results.
4. Reorient the use of learning strategies that are not suitable for certain processes.
5. Teach new learning strategies that are not used by the students.

The teaching step of the preparatory phase, was generally guided by the book COMPETITIVE PROGRAMMING: Increasing the Lower Bound of Programming Contests [17]. This book is used in Competitive Programming and Data Structures and Algorithms courses in the School of Computing at the National University of Singapore. The authors highly recommend this book to every competitive programmer, if not for enabling participation in the contests, at least during the middle phase of their programming career: when they want to leap forward from just knowing some programming language commands and some algorithms to becoming a top programmer.

Halim's book [17] covers several topics that are also taught in the different programming related courses taught at School of Computing [1]. While the book offers comprehensive coverage of the content relevant to the competition, there was much more needed to achieve the objectives of the program. For example, there are many exercises and programming problems scattered throughout the text of the book which can be skipped first if a solution is not known at that point of time, but can be revisited in latter time after the reader has accumulated new knowledge to solve it. Solving these exercises can strengthen the concepts, as the presented problems often contain interesting twists or variants of the topic being discussed.

The topics covered in [17] are:

- Introduction: Sequence, Selection, Repetition Programming Constructs, Recursion/Backtracking, Ad Hoc Problem.
- Data Structures and Libraries.
- Problem Solving Paradigms: Complete Search, Divide and Conquer, Greedy, Dynamic Programming.
- Graph.
- Mathematics: Ad Hoc Mathematics Problems, Combinatorics, Number Theory, Probability Theory, Cycle-Finding, Game Theory.
- String Processing.
- Computational Geometry.
- Advanced Topics: Advanced Search Techniques, Advanced DP Techniques, Problem Decomposition.
- Others Rare Topics.

The program of competitive programming has been designed so that the knowledge is incorporated continuously. In this way students could improve their skills independent of their particular capabilities, interests and needs.

Self-diagnosis and self-evaluation, together with the professor evaluation, are necessary components of this program. The assessment system of the program was designed from a diagnostic and self-evaluation tool, the use of online judges and developing local contests. The diagnosis and self-evaluation tool was adapted, and an online judge for problem repository and the hosting of local contests was used. The Caribbean Online Judge was successfully used to host both local and regional contests. Prior to the competitions, warm-up activities were carried out with the aim of introducing the students in an environment of competencies and reviewing the concepts that integrate the problems in the competition. The methodology adopted consists of presenting the problem statement, the deadline to solve it, and the submitting options. In the feedback made to each student, as well as indicating if the answer is correct, there is comments about how to improve the submitted solution.

In both local and regional contests, students are organized in teams of three. These type of competitions are however organized in the same manner, and with similar characteristics as regional and international contests. Students must solve algorithmic problems at a progressive level of difficulty. In addition to logic and the use of strategies, it is important to ensure that there is collaboration and communication among the team members. Once a team solves one problem, the solution is sent to the automatic judge who evaluates it with a testing dataset designed for the problem. In case that the proposed solution is correct it informs the score obtained by the group.

5 Preliminary results

This dynamic teaching-learning process reported in this paper was carried out during the second semester of the academic year 2016. The results presented here are therefore, preliminary results. In addition, the acquisition and exchange of knowledge was affected by several factors, such as, the novel context to which the students were incorporated, because this type of experience did not exist before at UNAM and also in the country; individual differences; previous academic difficulties; the social pressure of other students and discipline, among others.

A high degree of interaction was noted amongst the students. Interaction did not only take place in the groups, but it also included students from other groups. Interaction was also noticed between students from second and third years. Prior to this initiative, none of the students was considered to have had advanced programming knowledge. In addition, students just started from the knowledge raised up to that point in their studies. To solve programming problems, students do not only need programming skills, but they also need knowledge of Mathematics and English language. So, their difficulties in other subjects and previous years are also a factor to be highlighted and addressed during the development of the program.

The general feeling among staff was that this had been a positive and successful preliminary implementation of the program. The immediate feedback from the students supported this view too. All students commented that participation in the program and contests had been fun and was well worth repeating. Comments from other faculty members who witnessed the contest was also very favorable. We were reasonably satisfied with the attendance especially among the undergraduate students, considering that it was the first time that this type of experience is implemented in the country. After the positives results and feedbacks, it is now our hope that this practice will be taken up further in the quest to develop the much needed programming skills amongst our students.

As part of the preparation for the South Africa Regional Final, a Local Contest was organized with the aim of selecting the best three teams to represent UNAM in this regional contest. The Local Contest consisted of a set of 12 problems. When constructing the problem set we tried to make it somewhat easier than those that had been used in previous regional and world finals. The reason was that we wanted all the teams to be able to solve at least two or three problems. The top team solved all four problems. The three top teams represented the UNAM on the South Africa Regional Final carried out on 15 October 2016. Participants in the regional contest were from the following countries: Angola, Benin, Burkina Faso, Cameroon, Ethiopia, Kenya, Mauritius, Niger, Nigeria, Namibia, Togo and South Africa (with sites in Cape Town, Grahamstown, Pietermaritzburg and Pretoria). In total, more than 100 teams participated in the contest, which consisted of a set of 8 problems with different levels and topics. The three Namibian teams had only one correct solution with different time penalties. Also, they tried to solve and submitted wrong solutions for another two problems. These resulted in the teams being ranked at numbers 30, 33 and 34 of the general classification of the contest respectively.

It is to be highlighted that the top Namibian teams was the only classified teams amongst the classification that was dominated by South Africa. It is also important to note that the top team in Namibia was a girls-only team. Judging by student feedback, the amount of information given to students prior to the contest was enough in accordance with the time available. It should be noted that the members of the winning team spent considerable more time on preparations than the other teams. Moreover, the three top teams both contained third year students. This shows the importance of both training and experience in performing well in this type of event.

It seems like several students were surprised at how difficult it was to write correctly working programs. One hopes that one lesson learned is that both programming and team-work require practice and planning in order to be efficient. After the contests, discussions were held with students about the problems solutions. In these, it was very encouraging to see the interactivity and enthusiasm among the groups.

Finally, we can conclude as part of the continuous diagnosis and self-assessment carried out during the competitive programming program that the students showed remarkable improvements in the various topics covered. Some of these were: coding skills, algorithmic solutions, Data Structures, String Processing, and Problem Solving Paradigms. Although no conclusive results have been obtained in this sense and a deeper study is necessary in this respect, we were happy to note the improvement in the confidence levels of our students, as well as the motivation and enthusiasm that they now demonstrate towards programming related course. We have also noted a slight improvement in the academic results of the students, but this cannot be directly attributed to this intervention without further research.

6 Conclusions and perspectives

This paper presented the process followed to prepare UNAM students for competitive programming. Implementing competitive programming as support to the Teaching-Learning process of programming discipline has achieved positive preliminary results, and presents new opportunities for students to develop their programming skills. Although the study reported only on preliminary results of the first cycle of implementation, the observed results have given us hope to continue exploring this area of competitive programming as a viable option to motivating our students to learn programming. Our conclusion is that the program functioned as intended. The students were able to enthusiastically participate in the discussion of the problems and to work in teams. Students performed some self-study that they would otherwise not have done while preparing for the contest. Although we only reported on one cycle of implementation, we believe that the results and experience gained so far is positively valued by all participants and program members, revealing the potential of programming tournaments as a strategic resource for teaching programming. As future work and perspectives, we will develop a way to measure the impact of the program on Teaching-Learning process of programming and the motivation to attend contests. In addition, we think to include other higher Education Institutions at Namibia in the local contests, and in that way develop others different levels of contests like the Namibian League of Programming, Training Camps, and the ACM-ICPC National Final.

References

1. Faculty of Science, Prospectus 2016. University of Namibia (2016)
2. ACM-ICPC: About ICPC (March 2017), https://icpc.baylor.edu/
3. Aureliano, V.C., de AR Tedesco, P.C., Caspersen, M.E.: Learning programming through stepwise self-explanations. In: Information Systems and Technologies (CISTI), 2016 11th Iberian Conference on. pp. 1- 6. IEEE (2016)
4. Castillo, J.J., Cardenas, M.E., Serrano, D.J.: Experiencias en el desarrollo de competencias de programacion en utn-frc. In: VI Congreso de Tecnologa en Educaciony Educacion en Tecnologa (2011)
5. Coles, D., Jones, C., Wynters, E.: Programming contests for assessing problemsolving ability. J. Comput. Sci. Coll. 26(3), 28-35 (2011)
6. Cuba-Ricardo, G., Leyva-Figueredo, P.A., Mendoza-Tauler, L.L.: Learning strategies of informatics contestants. In: International Olympiads in Informatics. vol. 8, pp. 35{48. Vilnius University (2014)
7. De Fraja, G., Landeras, P.: Could do better: the effectiveness of incentives and competition in schools. Journal of Public Economics 90(1), 189-213 (2006)
8. Duran, M., Caro, A., Roderiguez, P.G.: Una apuesta por la motivacion al alumnadoen las asignaturas de programacion: el sistema de evaluacion continuada. Actas de las IX Jornadas de Ense~nanza Universitaria de Informatica, Jenui pp. 191-196 (2003)
9. Elgamal, A.F., Abas, H.A., Baladoh, E.S.: An interactive e-learning system for improving web programming skills. Education and Information Technologies 18(1), 29-46 (2013)
10. Enstróm, E., Kreitz, G., Niemelá, F., Sóderman, P., Kann, V.: Five years with kattisusing an automated assessment system in teaching. In: Frontiers in Education Conference (FIE), 2011. pp. T3J-1. IEEE (2011)
11. Estiven Molina, Y., Espinosa Sarmiento, D., Sanchez Trujillo, E.F.: Uso de la programacin competitiva como aporte a la enseanza de la lgica algortmica. Experiencia universidad de la amazonia. Revista Cientca (26), 109-116 (2016)
12. Forisek, M.: Pushing the boundary of programming contests. Olympiads in Informatics 7, 23-35 (2013)

13. Fracchia, C.C., Kogan, P., Alonso de Armino, A.C., Godoy, I., Lopez, L.M.: Realizaci on de torneos de programacion como estrategia para la ense~nanza y el aprendizaje de programacion. In: XX Congreso Argentino de Ciencias de la Computacion (Buenos Aires, 2014) (2014)
14. Garcia-Mateos, G., Fernandez-Aleman, J.L.: A course on algorithms and data structures using on-line judging. In: ACM SIGCSE Bulletin. vol. 41, pp. 45-49. ACM (2009)
15. Garcia-Mateos, G., Fernandez-Aleman, J.L.: Make learning fun with programming contests. In: Transactions on edutainment II, pp. 246-257. Springer (2009) .
16. Guerreiro, P., Georgouli, K.: Enhancing elementary programming courses using elearning with a competitive attitude. International Journal of Internet Education 10 (2008)
17. Halim, S., Halim, F.: COMPETITIVE PROGRAMMING. Increasing the Lower Bound of Programming Contests. School of Computing. National University of Singapore (2013)
18. Hassinen, M., M.H.: Learning programming by programming: a case study. In Proceedings of the 6th Baltic Sea Conference on Computing Education Research (Koli Calling 2006). pp. 117-119 (2006)
19. Heintz, F., Fárnqvist, T.: Pedagogical experiences of competitive elements in an algorithms course. In: LTHs 7: e Pedagogiska Inspirationskonferens, 30 augusti,Lund (2012)
20. Hernandez, B., Montaner, T., Sese, F.J., Urquizu, P.: The role of social motivations in e-learning: How do they affect usage and success of ICT interactive tools? Computers in Human Behavior 27(6), 2224-2232 (2011)
21. Hurtatiz, Y.E.M., Sarmiento, D.M.E., Trujillo, E.F.S.: Uso de la programacion competitiva como aporte a la ensenanza de la logica algortmica. experiencia Universidad de la amazonia. Revista Cientca 3(26), 109-116 (2016)
22. IOI: International olympiad in informatics (March 2017), http://www.ioinformatics.org/history.shtml
23. Iqbal, S., Harsh, O.K.: A self-review and external review model for teaching and assessing novice programmers. International Journal of Information and Education Technology 3(2), 120 (2013)
24. Jam, G.C.: What is code jam? (March 2017), https://code.google.com/codejam/ about
25. Leal, J.P., Silva, F.: Using mooshak as a competitive learning tool. In: The 2008Competitive Learning Symposium (2008)
26. Massey, B.: Experience with a process for competitive programming. Computer Science Department. Portland State University (2001)
27. Mcgettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G., Mander, K.: Grand challenges in computing education. UK Computing Research Committee (2005)
28. Mohorovicic, S., Strcic, V.: An overview of computer programming teaching methods. In: Central European Conference on Information and Intelligent Systems (2011)
29. Moreno, J., Pineda, A., Montoya, L.: Uso de un ambiente virtual competitivo para el aprendizaje de algoritmos y programacion-experiencia en la universidad nacionalde colombia. In: Nuevas Ideas en Informtica Educativa TISE 2013. Brasil (2013)
30. Othman, M., Othman, M., Hussain, F.M.: Designing prototype model of an online collaborative learning system for introductory computer programming course. Procedia-Social and Behavioral Sciences 90, 293-302 (2013)
31. Oztemel, E., Yavuz, E.: Comparison of learning models to build an infrastructure for performance measurement of e-learning systems. In: International Conference on Web-Based Learning. pp. 307-311. Springer (2009)
32. Pineda Corcho, A.F., et al.: Modelo tecno-pedagogico basado en ludicacion y programacion competitiva para el dise~no de cursos de programacion. Ph.D. thesis, Universidad Nacional de Colombia-Sede Medelln (2014)
33. Ragonis, N.: Type of questions the case of computer science. In: Olympiads in Informatics. pp. 115-132. No. 6 (2012)
34. Salamo, M., Camps, J., Vallesp, C., Vernet, D., Llora, X., Bernado, E., Ga rell, J.M., Gonzalez, X.: Iniciativas para motivar a los alumnos de programacion. Actas de las VII Jornadas de la Ense~nanza Universitaria de la Informatica (JENUI 2001) p. 329 (2001)
35. Souza, D.M., da Silva Batista, M.H., Barbosa, E.F.: Problems and weaknesses in the teaching and learning of programming: A mapping review. Brazilian Journal of Computers in Education 24(01), 39 (2016)
36. Sunday, J., Isabwe, G.M.N., Ali, M.U., Schulz, R.P.: Towards personalization of peer review in learning programming. In: International Symposium on Emerging Technologies for Education. pp. 270-277. Springer (2016)
37. Tan, K.S.T., Lee, Y.: An augmented reality learning system for programming concepts. In: International Conference on Information Science and Applications. pp. 179-187. Springer (2017)
38. TopCoder: Competitive programming at topcoder (March 2017), https://www. topcoder.com/community/competitive-programming/
39. Wang, Q., Yu, Y., Yu, Y.: Discussion on computer-based teaching methods. In: Intelligent Ubiquitous Computing and Education, 2009 International Symposium on. pp. 285-287. IEEE (2009)
40. Wang, Y., Li, H., Feng, Y., Jiang, Y., Liu, Y.: Assessment of programming language learning based on peer code review model: Implementation and experience report. Computers & Education 59(2), 412-422 (2012)
41. Xia, B.S.: An in-depth analysis of learning goals in higher education: Evidence from the programming education. Journal of Learning Design 10(2), 25-34 (2017)
42. Yengin, _I., Karahoca, D., Karahoca, A., Yúcel, A.: Roles of teachers in e-learning: How to engage students & how to get free e-learning and the future. Procedia-Social and Behavioral Sciences 2(2), 5775-5787 (2010)

Application of an Action Research Process: Reflections on an Undergraduate Information Systems (IS) Software Development Project (SDP)

Duane Boucher[1] and Roxanne Piderit[2]

Department of Information Systems,
University of Fort Hare, South Africa
[1]dboucher@ufh.ac.za, [2]rpiderit@ufh.ac.za

Abstract. Within the uniquely challenging constraints of the modern university environment, academics are required to adapt to the needs of the students, requirements of the higher education environment, and the expectations of the industry which will employ their students. Subsequently, academics need to constantly reflect on their academic practice for professional growth and adapt the tools used for instruction in order to ensure that their unique students cohort are empowered during the teaching and learning process. This paper details an action research approach undertaken to investigate the impact that a given intervention has on the quality of the software development projects completed by an Information Systems undergraduate class. The intervention introduced was an Agile Methodology with Scrum instructional tool, which provided the process to be followed by the five project groups in the undergraduate class. This paper reports on the third action research cycle with the first cycle intervention being the waterfall approach and the second cycle intervention the agile methodology. The main themes identified from the previous action research cycles inform the lessons learned for this paper. The themes are associated with the determinants of the software quality expected by a client. Qualitative data in the form of project deliverables, project group feedback, tutor reports and lecturer class notes were collected throughout the academic year. The data provided a means to reflect on the impact that the Agile Methodology with Scrum instructional tool had on the quality of the software development projects. It was determined that the intervention had a positive impact on the quality of the group projects compared to previous years, but that a number of problems persisted. Eight lessons learned are discussed and related to the intervention and its impact on the software quality of the student projects. The intervention did not adequately address all the factors impacting on the quality of students software projects and therefore the intervention will have to be refined and another action research cycle completed,

Keywords: Assessment, Curriculum Development, Learning Perspectives, Software Quality, Agile Methodology, Scrum.

1 Introduction

The software development project (SDP) is often recognized as the pinnacle of an information systems (IS) student's education. This is due to the valuable experience gained through the practical implementation of the skills taught in practical and theoretical subjects during the course of their studies. However, the development and management of the SDP by students is quite overwhelming when they are initially presented with the scope for the course. If suitable instructional tools are not provided then the quality of the systems that are developed can be negatively impacted, which results in the final system that is delivered not meeting the expectations of the client. For this reason, it is necessary to ensure that the software development methodology followed for the SDP course is suitable to achieve the objectives of the curriculum and foster the project groups in the software development process. In order to determine a supportive software development methodology for the SDP course, an action research process was conducted.

Action research is one of the most popular methods of professional development for educators [1] and to improve the curriculum [2]. The distinguishing characteristics of action research, specifically within the context of educational research are that it is carried out by people that are directly concerned with the social situation that is being researched; it starts from practical questions arising in everyday teaching practice; and it needs to be compatible with the educational values of the institution [2].

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

The authors detail their experiences while using an action research process to introduce an agile methodology with scrum instructional tool as an intervention with the intention to positively impact the quality of the SDP. The lessons learned by the authors are discussed by reflecting on the qualitative data collected during the action research cycle.

The context of the paper is briefly described and thereafter the theoretical underpinnings of action research are described, which includes a discussion of the stages of action research. A brief description of the software methodologies are presented to provide a point of reference for the discussion of the interventions undertaken. The reflections of the authors on their experiences during the action research process are discussed. Finally, eight lessons learned are presented.

2 Context

Fanghanel [2] states that academics at universities typically operate at a macro, meso and micro level of academic practice. These are explained as follows:

- Macro level comprises the institution *(university)*, external factors *(e.g. higher education regulations, student dynamics, cultural and societal divides, and economic prospects)*, teaching requirements, and the teaching-research engagement.
- Meso level includes the academic department or unit and the discipline specific subject matter.
- Micro level is indicative of the individual lecturer and the personal circumstances which impact on how they conduct their daily work.

The authors are academics *(micro level)* who work in the Department of Information Systems *(meso level)* at the University of Fort Hare (UFH) *(macro level)*, South Africa. The Department of Information Systems offers an IS *(meso level)* undergraduate, three-year degree within the Faculty of Management and Commerce.

UFH is an English-medium university that is located in the Eastern Cape Province and is a historically disadvantaged institution (HDI). As a HDI it faces significant financial constraints resulting from the institutions multiple 'catch-up' improvement plans, which residually impact on the availability of resources for teaching and learning. In addition, a resource challenged rural schooling system in the Eastern Cape Province, which are the main source of UFH students, means that the multitude of the registering students are typically English second language speakers.

Additionally, given the diverse population characteristics, socio-political challenges and historical oppression in South Africa, UFH encourages its educators to adopt a humanizing pedagogy. The underpinnings of a humanizing pedagogy are that educators view the student as a whole person by considering their intrapersonal affects *(feelings, beliefs and experiences)* to assist them in developing their interpersonal relationships with others [4]. Therefore, the educator needs to be aware that the teaching and learning interventions are supportive and culturally relevant, rather than culturally biased or punitive.

Within the context described, the authors undertook action research to reflect on the teaching methods being used in order to improve their academic practice and the realization of the course outcomes by the students. Therefore, the focus was the interventions needed to support the students while they complete the SDP course.

2.1 Background on the IS Software Development Project (SDP) course

The experience of completing a SDP during an IS undergraduate curriculum is intended to provide students with an opportunity to put their foundational knowledge and skills learned during their undergraduate studies into practice in a controlled environment [4,5]. The SDP is typically found in an undergraduate course or is itself the capstone course for an undergraduate degree. During the length of the year-long course students are exposed to a 'real world' SDP, where they learn about the impact of their actions and choices associated with, but not limited to: teamwork dynamics, requirements elicitation, project management deliverables and constraints, system design and the formal nature of software development methodologies. The actions and choices of the students are guided through the instructional model focus *(e.g. documentation, methodologies, business ethos, teamwork, programming, or some hybrid)* adopted for the SDP for the purpose of achieving the expected outcomes of the course [5].

The focus and outcome of the SDP at an undergraduate level are influenced by the educators' perceptions, their academic practice, and their interpretation of the needs and requirements of industry for information communication and technology (ICT) graduates. In South Africa there is an ongoing debate about the misalignment between the ICT-related graduate competencies acquired at tertiary level and those required by industry from graduates

[6]. The main criticisms being levelled are at the graduates' practical application of the skills learned at university and the importance of soft skills competencies *(i.e. business management literacy, verbal and written communication, time management and project management skills)* [6,7].

The SDP contains an academic curriculum that can be used to mitigate the misalignment between industry ICT needs and tertiary education ICT-related content delivery. The mitigation of the misalignment is achieved through the practical application of the course taught in the process of completing the degree. The location and extent of the SDP in the qualification is largely dependent on the intended emphasis placed on it by the academic department.

2.2 The SDP in the IS Degree at UFH

At the UFH, the first-year of study is generic for all commerce students, with discipline specialization commencing from second-year. From second-year IS undergraduate students complete core courses in systems analysis and design, databases, programming and a choice of commerce-related courses. The second-year provides the prerequisite knowledge for an IS focused third-year where students continue with courses in programming, IS project management, professional communications and a yearlong SDP. The SDP is a separate course which is assessed continuously during the course of the academic year. The knowledge and skills acquired from courses done in second-year and being done in third-year contribute significantly to the success of the SDP course and students are repeatedly reminded by the authors that the degree must be viewed as a sum of its parts, rather than viewing courses in silos.

The SDP was previously only done in the 14 weeks allocated to the final semester of the third-year. The short time period allocated to the students meant that the standard and quality of the developed systems was compromised. This was compounded by the fact that the project methodology used at the time was the Waterfall Methodology, which meant that the students often started developing their systems late in the semester. The course assessment at the time was weighted heavier towards documentation (systems analysis pursuits) and minimally on the development of a functional system in order to account for the limited development time.

In order to provide the project groups with more time for development the academic structure was amended to allow the SDP to run over the course of the academic year in parallel with the courses being done at third-year. The increase in the available time allocated to the SDP has allowed the project groups sufficient opportunity to provide equal focus to the documentation and coding of their systems and ultimately enhanced the SDP experience as a project group.

Within an academically controlled environment, students are assigned to a project group of three to five members that carry out the SDP for an identified client. Dependent on the accessibility of suitable clients for the project groups – some of the clients may originate from industry and others may be 'pseudo clients'. The project groups are encouraged to identify their own industry clients based on an identified problem or opportunity that they intend to address. The facilitated interaction with an industry client further assists in developing the soft skills of the project group members, so finding real clients who are willing to volunteer their time is beneficial and highly recommended. For those project groups who cannot identify an industry client, the lecturers/mentors or teaching assistants take on the role of 'pseudo clients' and provide a predefined business scenario to the designated project group. Whether the client is from industry or pseudo, the project group must undertake the SDP using a software development methodology and project management principles to manage their project deliverables for the duration of the course.

The delivery of the final software product is an integral part of the final summative assessment calculation of the course. The final mark assigned for the course is based on a series of formative and summative rubrics which consider not only the final delivered product, but also the associated documentation produced throughout the year, the coding practice evidence, team dynamics, and soft skills mastered.

To ensure that the assessment rubrics and practical experience of the software development projects are complimentary, regular evaluations of the course are an integral part of the process. These evaluations, where the educator reflects on their practice and students provided feedback, serve an additional purpose of ensuring that the course is constantly reviewed so its relevance to industry expectations and academic content delivery can be monitored.

3 Literature Review

The literature review briefly considers the notion of an educator as a reflective practitioner, the use of action research to evaluate academic practice, and the purpose of software development methodologies.

3.1 The Educator as Reflective Practitioner

To be a reflective practitioner, educators need to be cognizant of the fact that their training is insufficient when faced with novel teaching situations or problems, and therefore they need to reflect in order to improve their teaching practice [8]. University academics are notoriously averse to self-reflection and change, and often cite their years of experience in support of their unchanging academic practice [2]. However, given the ever-changing characteristics of the student cohort and the socio-technical-environmental flux, it would be myopic of academics to believe that there is no need for dynamic teaching practices. Reflective practice provides academics an opportunity for self-development through active learning [2] which promotes professional development. The most popular method of professional development of academic practice for educators is through action research [9,10].

3.2 Using Action Research to Evaluate Academic Practice

Defining action research. "Action research is a form of self-reflective enquiry undertaken by participants in social situations in order to improve the rationality and justice of their own practices, their understanding of these practices, and the situations in which these practices are carried out" [11:162]. Elliott [9:69] provides a succinct definition which states that the focus is on "the study of a social situation [learning environment] with a view to improving the quality of action [teaching practice] within it"[1]. Therefore, action research should be undertaken in order to evaluate and inform the future teaching practices (actions) of the teacher (actor) [12,13].

Action research is a commonly used method for the demonstration of the scholarship of teaching and learning (SOTL) [2] and to cultivate professional growth [14], because it provides the means to improve teaching practice through reflection [15]. SOTL is concerned with conducting research, identifying gaps, investigating alternatives, and correcting shortcomings in the teaching practice and learning environment, and specifically sharing these findings with others, so that they can learn from the experiences of other teachers' in educational settings.

Kember [2] identifies seven characteristics of action research, namely: social practice; aimed towards improvement; cyclical; systematic enquiry; reflective; participative; and, determined by the practitioners. This paper's authors expand on Kember's seven characteristics of action research as follows:

- *It is a social practice* – therefore the research approach should typically be interpretive, because it requires an immersion into the given setting or context (in-situ), which is influenced by the macro, meso and micro levels of academic practice.
- *It is aimed towards improvement* – the intention of action research is to be a catalyst for change at the macro, meso and micro level of academic practice.
- *It is cyclical in nature* – action researchers frame their enquiry in a set of steps, which typically includes the following of an action research process where they review the context of enquiry (setting), isolate the issue or problem (identify), plan for the research intervention (plan), conduct the required action or intervention (act), complete an analysis and reflective interpretation of the action or intervention (reflect), modify your future practice (change) and report on the findings (share). This process is repeated as needed by the researcher in pursuit of a positive outcome of the action or intervention undertaken.
- *It should contain systematic enquiry* – because action research has the perception of being considered too unstructured and whimsical to have the credibility of established research practices, it is often dismissed by experienced researchers. However, action researchers can draw on the research methodologies employed along the interpretivist-positivist continuum in order to add credibility or validity to their action research process. This provides the action researcher with a link to an established paradigm and assists them in reporting on the research process and data analysis done.
- *It is reflective in nature* – because the action researcher has to reflect on their practice in-situ, they need to guard against researcher bias when reporting on their findings. Keeping field notes (or a research diary) recorded during the action research process provides the action researcher with a longitudinal reference of their perceptions and influence on the impact of the action or intervention.
- *It should be participative in nature* – the individual or group of action researcher(s) should share their findings with their peers. This reflects that the intervention is more than simply curriculum development, but includes a research component that can inform others in their academic practice.
- *It is determined by the practitioners* – the action researcher typically identifies a research problem associated with their academic practice which they want to address [16]. The focus is often on the student learning and determining how an identified learning barrier can be corrected through some form of intervention.

[1] Square bracket wording added by the authors

The provided characteristics of action research assist in clarifying the nature of action research, but the question of the relevance of action research often arises [15]. According to Norton [2] action research, from a pedagogical stance, can serve a number of functions, simultaneously, such as:

- testing the quality of curriculum redesign;
- engaging with continuing professional development [14];
- producing research output [1]; and,
- impacting on student learning.

The action research reported on in this paper takes into consideration the characteristics and functions mentioned above and were carried out by going through the several stages in the action research process.

The Stages of Action Research. Action research is viewed as a spiral process where you generally follow a plan, act, observe and reflect on the process [2]. The repetition of the cycles allows for the application of the evaluations from the previous cycle to the new one, through the planning of new actions and interventions [2,10]. The evaluations which are conducted form the primary data source for action research.

A number of graphical representations of the action research cycle have been discussed elsewhere [1,9,10,14] and Fig.1 is an adaption from literature. The focus of each of the steps is briefly described below. In Section 5 we relate the steps to the context of this research, namely the SDP course.

Stage 1: Setting. Before undertaking action research the researcher needs to be aware of the macro, meso and micro level influences on their academic practice. This is important as it assists in framing the context for the study and assists the action researcher in identifying those internal and external factors in the setting which will impact on the action research process being undertaken. It may even be possible for the action researcher to identify those internal and external factors that may be positively affected during or after the intervention. Similarly, if the action researcher is not aware of the setting and undertakes the intervention regardless, then the intervention may appear to have worked, but the lessons learned may not be easily applied to future practice [10].

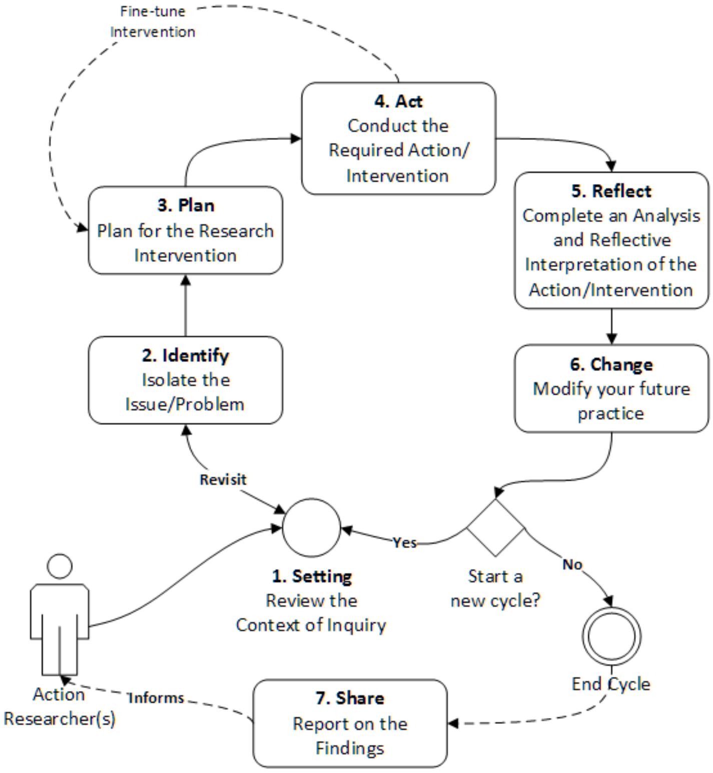
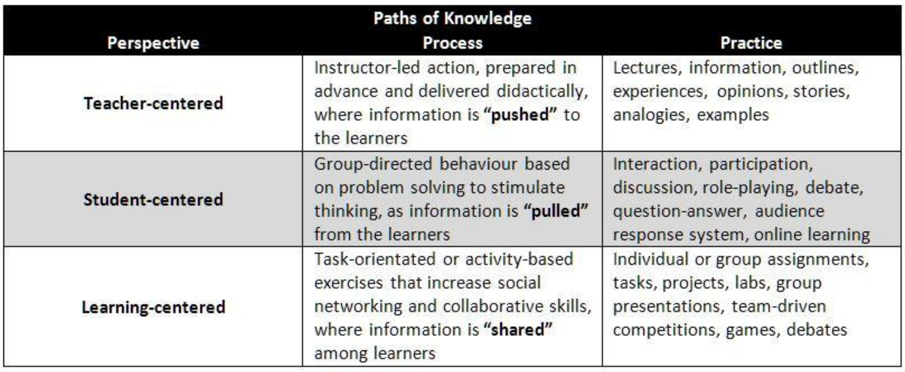

Fine-tune Intervention
4. Act
Conduct the Required Action/ Intervention
3. Plan
Plan for the Research Intervention
5. Reflect
Complete an Analysis and Reflective Interpretation of the Action/Intervention
2. Identify
Isolate the Issue/Problem
6. Change
Modify your future practice
Revisit
Yes
Start a new cycle?
No
1. Setting
Review the Context of Inquiry
End Cycle
Action Researcher(s)
Informs
7. Share
Report on the Findings

Fig. 1. Stages of an Action Research Process (developed from literature)

Stage 2: Identify. In this stage the action researcher needs to identify the problem or issue that is going to be addressed through an intended intervention [16]. The problem may not be immediately evident, or the action researcher may select the first presented problem as the actual problem. Subsequently, it is important that the action researcher clearly defines the problem by making it explicit (putting it in writing) in the form of an objective, or intended outcome, of the action research. This assists in getting the perspectives of others on whether the identified problem is the actual problem, and allows for a deeper interrogation of the problem. The initially identified problem may just be the 'tip of the iceberg' and not the root causes. By understanding the root causes the action researcher can better determine if the primary problem has been identified. Furthermore, the action researcher will need to revisit *Stage 1 (Setting)* to deduce the root causes of the problem by interrogating the effect of the internal and external factors on the problem. One method that can be used in this regard is the Fishbone/Ishikawa Diagram to map the cause-effect relationship to the identified problem. Each cause identified will provide support for the effect, but at the same time it assists in scoping the intervention that will be undertaken.

Stage 3: Plan. This stage involves developing the intended intervention strategy [2]. Often the intervention will be guided by the learning perspective that the action researcher intends to use in the classroom. The perspective chosen to focus the intervention could be teacher-centered, student-centered, and/or learning-centered (*cf.* Table 1). Each of the learning perspectives will take a different path to knowledge and this will be influenced by the intended process and practice that is selected. The action researcher will need to determine if a new strategy or approach to the existing curriculum is needed, or if the existing curriculum can be sufficiently influenced to address the intended objective of their study. For example, for the SDP course the learning *perspective* is learning-centered, which requires a *process* where the project group members must share information in *practice* when working on their group assignments, project, and presentations.

Table 1. Centred learning from different perspectives [17:58]



	Paths of Knowledge	
Perspective	Process	Practice
Teacher-centered	Instructor-led action, prepared in advance and delivered didactically, where information is **"pushed"** to the learners	Lectures, information, outlines, experiences, opinions, stories, analogies, examples
Student-centered	Group-directed behaviour based on problem solving to stimulate thinking, as information is **"pulled"** from the learners	Interaction, participation, discussion, role-playing, debate, question-answer, audience response system, online learning
Learning-centered	Task-orientated or activity-based exercises that increase social networking and collaborative skills, where information is **"shared"** among learners	Individual or group assignments, tasks, projects, labs, group presentations, team-driven competitions, games, debates

Action research can be associated with the process model to curriculum development [18,19], which adopts a more student-centered and learning-centered approach, that emphasizes activities and effects. It is focused on what happens in the learning environment and the exchange between teacher-student as contributors to the process. The student has greater choice over the learning activities and the environment [20]. However, Stenhouse [18] warns that when adopting a process model to curriculum development, educators need to guard against giving the learner complete freedom, as the educator's expected motivators of students behavior and the students' actual motivators of behavior may be vastly different. Therefore, when designing the interventions the action researcher needs to ensure that there is constant monitoring of the intervention through the appropriate assessment methods (formative and summative).

Additional class evaluations can assist in identifying areas where the interventions are not successful to the extent that they are actually detracting from the intended outcome of the study. The assessments together with the evaluations can form part of the data collection methods, which also need to be identified at this stage of the action research process. If the action researcher does not highlight the data collection methods to be used during planning, then they could get to the reflection stage and find that they have little evidence to use for reflective purposes [10].

In completing this stage the following checklist needs to have been addressed:

- Did you decide on the learning perspective that you are going to adopt? How much control are you going to exert on the process? Are the identified assessments appropriate and sufficient?

- Are the practices that you have chosen applicable? Can they be mapped to individual outcomes? Will these lead to the primary objective of the intervention?
- Did you remember to factor in the characteristics of your students and current learning environment identified in Stage 1?
- Are the data sources identified and ideally mapped to a given outcome? These data sources could include, but are not limited to: assessments, observations, questionnaires, interviews, artifacts, field notes. Will you perform some form of triangulation? That is collecting data from different perspectives (yours, students, and peers) and at different times throughout the intervention plan.
- Is the intervention plan realistic and detailed in such a manner that it will not be ambiguous during the teacher-student actions?

Stage 4: Act. During this stage the action researcher implements the planned teaching strategy. The action researcher monitors the process to ensure that the intended interventions are working as planned [10]. If an intervention is not working successfully then the action researcher needs to fine-tune the planned teaching strategy by revisiting the checklist mentioned above to determine what changes are necessary [22]. The fine-tuning is a necessary part of this stage as the action researcher is an active participant in the process with the other stakeholders. The data sources for later analysis and reflection are collected during this stage and can additionally inform the 'fine-tuning' process.

Stage 5: Reflect. This stage has a duel focus. Firstly, the action researcher must complete an analysis of the intervention. The nature of the analysis will be dependent on the primary data collected, which might include interviews, surveys, field notes, observations, or assessment records collected during Stage 4 and planned for in Stage 3. Secondly, the action researcher provides a reflection on the interventions undertaken by explaining the process and including supporting evidence, either qualitative or quantitative, which relates back to the intended purpose of the action research intervention. The primary data collection approach will determine the manner in which the supporting evidence is provided. However, whether qualitative, quantitative, or both, the action researcher needs to reflect (comment) on their role in the process and that of their students.

Stage 6: Change. The intention of an action research activity is to inform the academic practice of the action researcher [2,10,13]. This is accomplished by the action researcher identifying areas where they can make changes to their future practice. If the action researcher feels that they have accomplished the goal of their study then they can leave the action research cycle and report on their findings [22]. Alternatively, they can start a new cycle with the lessons learned from the previous one [10].

Stage 7: Share. The dissemination of the findings of action research is a critical part of an educator's professional development and takes on many forms [1]. These might include changing the academic practice from institutional to international level and/or contributing to knowledge at a national or international level. The outputs can include changes to current course material, workshops, conferences and journals.

Having discussed the stages of the action research process we will now briefly provide an overview of the software methodologies as applied in this research before reflecting on the research conducted.

3.3 Software Development Methodologies as an intervention

Software development methodologies provide a structure that can be utilized during a SDP for structuring, planning, and controlling. The software development methodology in this paper is viewed as the instructional tool which forms the intervention for the action research. There are various methodologies used in practice in software development and only those applicable to this article are briefly described, as follows:

- *Waterfall Methodology* – the primary emphasis of this methodology is on the documentation of the requirements, design and built system. The waterfall methodology follows a set of stages which includes variations related to: planning, requirements analysis, system design, system construction, system testing, and system maintenance. Each phase is completed sequentially, and the documentation is conducted in parallel to the stages in the development process and any changes in the scope are documented accordingly. The benefit of this approach is that there is detailed documentation, although the time to starting the building of the system is delegated to later in the process, which is a significant disadvantage for time sensitive software roll-outs.

Furthermore, there is often a disconnection between the planning, analysis, design, and programming activities [23].

- *Agile Methodology* – it follows the basic notion of the staged approach of the SDLC methodology, but 'boxing' is used to divide the project into multiple mini-projects (sprints) with an expected software output at the end of the iteration. The focus thus shifts more to the delivered product, than on extensive document production. The stages of the agile methodology can include: planning, requirements analysis, system design, coding, testing and documentation. The priorities of the project are reassessed prior to commencing the next iteration, so that the expected outcome for that iteration includes the under or over performance aspects related to the software product or element delivered in that iteration. Where the SDLC produces documentation that must be followed to the letter, in Agile the focus shifts to the communication between the project members working on that iteration [24], with the documentation taking the role of a dynamic working document.
- *Scrum* – is a method that lends project management principles to the agile methodology, such that it focuses on the improvement of the productivity of the project team [21]. The product owner provides a list of system requirements in the form of a product backlog. The team then identifies a portion of the product backlog that they will 'tackle' during a sprint session (two to four weeks), and daily Scrum meetings are conducted to determine the progress made. The process is facilitated by a Scrum Master who ensures that the scope remains focused on the identified goal of the sprint. The sprint is expected to produce a software artifact that is either ready for dissemination, or to be presented to a relevant stakeholder. The final deliverable of a sprint is a sprint review of the sprint process. This cycle then repeats itself by the project team deciding on another set of requirements until such time that all the requirements of the product owner are addressed [21].

These methodologies provide a common understanding for all the stakeholders in the systems development process of what core tasks need to be completed. Additionally, they provide a means to measure the progress of the project through the inclusion of project management processes during the development of a system.

4 Research Methodology

An interpretivist paradigm which adopted an action research approach was utilized in this research study in order to gain insight into a naturally occurring phenomenon in a third year undergraduate SDP class. The phenomenon under consideration was the efficacy of the intervention, that is an instructional tool *(an Agile methodology with Scrum)*, to impact on the quality of the SDP. The main aim of the research study was to determine the impact that the intervention would have on the project groups' delivery of a quality software product during the time available for the course.

The stages of action research, as described earlier (*cf.* Section 3) in this paper formed the research process which was followed to conduct the study. The participants in the research included two principle lecturers, two secondary lecturers, a tutor (teaching assistant), and the students in the class who were divided into five project groups of four members each.

Qualitative data was collected throughout the action research process. The data collection was in the form of anonymous individual student feedback on any difficulties with the instructional tool, focus group discussions with the project groups, classroom discussions, tutor meeting notes, annotations recorded on project group deliverables, and lecturer's field notes.

The data analysis process included the transcribing of all the notes collected during the action research process into pre-identified themes. The primary lecturers utilized the findings from the previous two action research cycles (done in previous years) to inform the intervention to be used in this cycle. The first cycle focused on the impact that the Waterfall Methodology had on the SDP, and thereafter the Agile Methodology was used as an intervention to get the students to develop better quality systems in a more efficient manner. The authors considered the analysis done during the previous action research cycles undertaken and used the findings to develop the themes that the current cycle needs to address. The themes where identified from problems experienced with the efficacy of the previous instructional tools to adequately support the curriculum outcomes in previous years. Specifically, the quality of the final software project delivered.

The authors identified the eight themes which are represented on a Fishbone Diagram (*cf.* Fig 2). The themes *(causes)* listed in Fig 2, had been previously found to impact on the overall outcome of the course curriculum, namely, that the software solution does not solve the client's problem *(effect)*. The implication is that the quality of the system development was not to the satisfaction and standard required by the client.

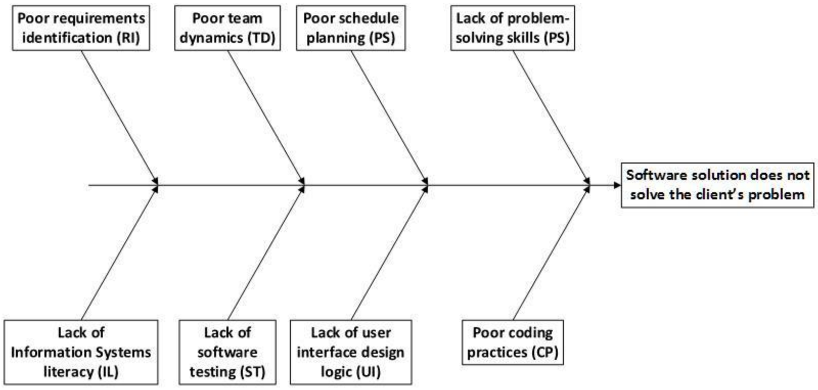

Poor requirements identification (RI)
Poor team dynamics (TD)
Poor schedule planning (PS)
Lack of problem-solving skills (PS)
Software solution does not solve the client's problem
Lack of Information Systems literacy (IL)
Lack of software testing (ST)
Lack of user interface design logic (UI)
Poor coding practices (CP)

Fig. 2. Fishbone Diagram for Cause-Effect Diagnosis

The eight themes were assigned a two character codes (*cf.* Fig 2), which made it easier to group the qualitative data collected during the course of the action research. The theme coding assisted in the eventual reflective purposes by the authors.

For ethical purposes the class was informed by the authors that they were conducting research on the efficacy of the instructional tool to assist in completing the outcomes of the curriculum. The individual students could either comment anonymously into the designated course assignment box or via the eLearning System on their experience with the instructional tool during the course of the year.

5 Findings and Reflective Discussion

The findings and reflective discussion are presented by utilizing the action research stages (*cf.* Fig. 1). The reflections are based on the final in a series of three action research cycles which were undertaken over a period of three years by the authors. The action research cycle discussed in this paper is focused on the utilization of the instructional tool (*cf.* Fig. 3) as the intervention with the undergraduate project groups in order to achieve the outcomes of the curriculum of the SDP, and to improve the quality of the SDP in the allotted time available for its development.

The agile methodology with scrum instructional tool as an intervention is depicted as a single sprint in Fig. 3. However, the process is cyclical and is comprised of three sprints, which each have four compulsory scrums and numerous additional scrums arranged between the project groups and the teaching assistant or lecturers. The initial sprint outcomes are informed by a *project planning document*[2] which is completed as a separate deliverable. On completion of the *project planning document* the students are tasked with dividing their project into three smaller projects with clearly identified deliverables. The deliverables become the intended sprint outcomes, which guide the scrum meetings. The information collected during the scrum meetings assists in developing the sprint documentation, and the system prototype, which is tested and subjected to a code review. At the end of each sprint the project group completes a presentation that identifies any shortcomings in delivering the sprint outcomes. The individual outputs of each sprint should cumulatively contribute to the final system being delivered.

[2] The Scrum Product Backlog, which provides a high-level list of all that needs to be completed (built or done) in order to provide the final system, is included as part of the Project Planning Document.

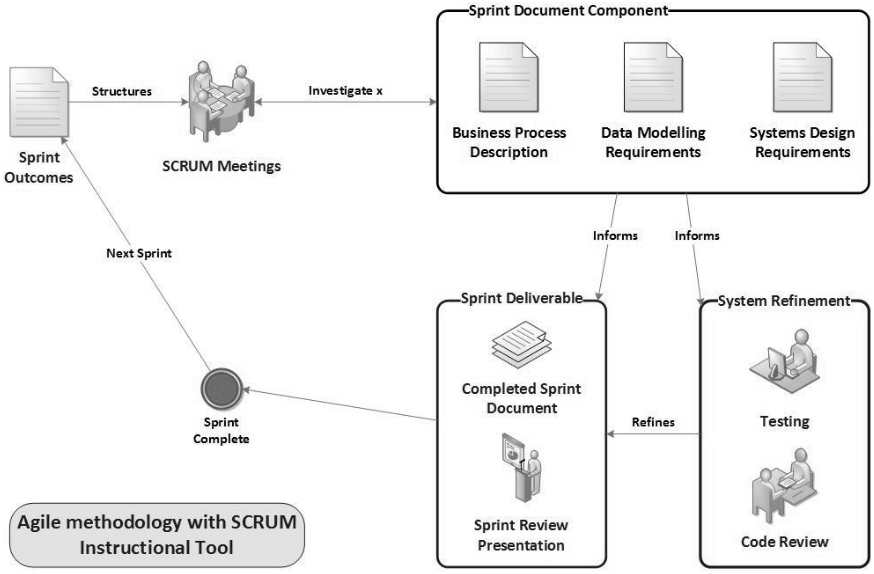

Sprint Document Component
Structures
Investigate x
Sprint Outcomes
SCRUM Meetings
Business Process Description
Data Modelling Requirements
Systems Design Requirements
Informs
Informs
Next Sprint
Sprint Deliverable
System Refinement
Completed Sprint Document
Testing
Sprint Complete
Refines
Sprint Review Presentation
Code Review
Agile methodology with SCRUM Instructional Tool

Fig. 3. Agile methodology with scrum instructional tool

The authors' reflections of the action research process undertaken during the SDP course are discussed further.

Setting. The authors considered the industry expectations for graduates (feedback was received from an advisory board, online employment portals and alumni), the institutional constraints (teaching schedule, funding of teaching assistants, computer resources and meeting room venues), the characteristics of the students, and the previous years' course evaluations, when planning and putting the intended intervention into action. The considerations assisted the authors in identifying a list of internal and external factors, which were recorded and mapped to a risk matrix. The risk matrix was a tool the authors' identified from project management that could provide a visual representation of which factors could most adversely impact on the action research process and the intended deployment of the instructional tool as the intervention. By categorizing the risk factors as high, medium, and low, the authors were able to put measures in place to mitigate the impact of the risks on the SDP.

Identify. The authors initially identified the problem as the fact that Information Systems students lacked a practical application of software development skills for industry and considered interventions to address that shortcoming. However, on further consideration of the internal and external factors impacting on the software development project course, the authors realized that the objective for conducting the action research was - to provide students with a software development methodology which ensures that the presented client problem (opportunity) is addressed by the final software solution developed. We arrived at this objective by completing the Fishbone Diagram in Fig. 2, where the effect was listed as: Software solution does not solve the client's problem. On further reflection, we realized that although the students were able to document the software solution there was often a mismatch between the documentation, the final solution, and the original problem specified. We also realized that more emphasis had to be placed on providing a learning environment where students can better deliver the solution to the presented client problem, as this is the ultimate intention of industry delivered software solutions. Initially we considered introducing scrum as conducted in industry [21], but the academic environment where this action research was conducted does not support daily meetings due to student course commitments. We therefore kept the core ethos of the scrum approach, but made changes to how it was applied in our context. These changes were incorporated into the teaching strategies and techniques, and associated with the agile methodology with scrum instructional tool used as the intervention to improve the quality of the delivered systems. The intended goal was to get the project groups to start their development much earlier in the academic year.

Plan. The authors determined that a learning-centered approach (*cf.* Table 1) and focusing on a process model of curriculum development was the best way to undertake the SDP intervention, as the students are the creators of their own knowledge during the SDP. The project groups collect, analyze, refine, document and implement their findings during the course of the SDP. In order to arrive at the objective of the study it was necessary to determine a suitable intervention which would improve the quality of the systems developed by the project groups for the intended client in the time available. The authors adopted an agile methodology with scrum instructional tool (*cf.*

Fig.3) as the intervention used in order to determine whether it could improve the quality of the delivered systems in the SDP course.

In addition to the structure provided in Fig.3, the authors reflected on their experiences of managing the previous year's SDPs with the undergraduates and determined that a level of control was needed to ensure that the students completed the tasks required. Subsequently, detailed assessment touchpoints were identified, artifacts (documents) were collected from the students for evaluation after each sprint, regular feedback was solicited from the students, and the authors recorded their findings in field notes.

Act. The authors implemented the intervention by providing the project groups and teaching assistant[3] with briefing sessions and documentation on the process that would be followed during the three sprint cycles. An immediate problem that arose when the scrum sessions begun was that the project groups were adopting a teaching-centered approach and waiting for input from the authors and the teaching assistant. The authors fine-tuned the scrum session activity by providing the project groups with sessions that were more structured and focused on the sprint document components *(Business Process Description, Data Modelling Requirements, and System Design Requirements)*. The additional structure provided for the scrums encouraged the students to become active participants in the learning process. The sprint sessions were scheduled to last from 15-30 minutes and the project groups were required to provide the intended agenda for the scrum session in advance. Another area that required fine-tuning was the system refinement activities (system testing and system code review). The project groups needed additional assistance in understanding the need for the testing and code review sessions, and additional information sessions were conducted with the class to explain their importance to providing the required system output for a sprint. These information sessions became regular occurrences where students could raise 'muddy issues' which arose during their scrum sessions or the sprint being undertaken. Many of the issues that were raised were easily mapped to the causes identified in Fig. 2.

The issues of poor coding practices, lack of user interface design logic, and lack of software testing were revisited and fine-tuned throughout the action research process. The authors' together with the undergraduate programming lecturers (secondary lecturers) in the academic department worked at refining the structure of the testing and code review sessions. Detailed rubrics and a *coding pitfalls sheet* of common errors made by project groups were circulated to the project groups in an attempt to reduce the occurrence of these errors. It was anticipated that the project groups would action the common errors and thereby improve the quality of the final system delivered.

Reflect. The process undertaken for Fig. 3 had a number of successes, but there were still problems experienced with the system deliverables in each sprint. The project groups focused their attention on the sprint document and not the coding necessary to complete the specific sprint outcomes. The student feedback received in quotations from a project group indicates that they viewed the sprint document as more significant than the coding of the system.

> "Our group focused all its attention on the sprint document, because it counted 50% of the marks in each sprint."

We revisited the assessment criteria for the course and determined that although we had a 50/50 ratio between the documentation and practical aspects (scrums, testing, code reviews, and scrum presentations), the students focused on the 50% assigned to the documentation, and did not realize the significance of the sum of the remaining marks. The project groups were initially required to produce three separate documents as part of the sprint, but these were combined into a single document during the course of the action research. Although, the students were still completing the same content, they psychologically interpreted the reduction of three documents to one document, as a significant reduction in documentation.

> "We know that the programming is important, but we spent more time on the documentation as it was easier to do."

We did not initially identify the root of this problem, as the students received a solid grounding in programming during the course of their undergraduate studies and their assessments in programming indicated that they possessed the necessary competency. However, after further investigation we discovered that a significant problem

[3] The teaching assistant took on the role of Scrum Master which required him to assist the project groups in identifying those elements that were slowing them down and work towards eliminating them.

was being encountered with the design and development of the database necessary for the underlying system. The project groups were completing the Entity Relationship Diagrams (ERDs) and list of tables and attributes for their documentation, but were then developing different databases when designing the front-end interfaces. Because we were not directly assessing the physical database build we only discovered this problem during the final sprint when we conducted focus discussions with the project groups. It once more highlighted the concern that students have a tendency of viewing activities in silos, rather than in an integrated and holistic view. Additionally, the requirements being documented were not being applied to the system build, which resulted in systems with 'faulty' business logic, and thus not realizing the client's intended system.

Change. The authors determined that the action research intervention started the project groups coding their systems at an earlier stage and produced improvements in the quality of the final system over previous years. However, there is still significant scope for improvement in the quality of the final delivered system. The database problem identified would require a changed to the intervention, so that the focus on the database is articulated better. The lack of information systems literacy problem persists in the fact that students' view their completed undergraduate studies in silos and are not applying the content already learned to their projects. It is therefore necessary to clearly map the activities in the intervention and the outcomes for the SDP course to the undergraduate modules. This will assist in identifying the knowledge students should possess and that which they must investigate further by becoming active participants in their own learning process. In this regard the authors have for many years drawn two horizontal lines on the board in the SDP class – the bottom one was labeled *"This is what you know"* and the top line was labeled *"This is the final project"*. The area between these two lines was colored in and labeled *"These are things you don't know yet, but need to know"* to emphasize the need for not only determining the requirements for the system, but also making students realize they have many skills to develop during the course of the year.

6 Lessons Learned

From the intervention undertaken during this action research study the following lessons learned are provided to inform the academic practices of future SDP courses. Although, each classroom setting will be unique, the causes of poor quality undergraduate software products are significantly generalizable. The need to ensure that student project groups start developing their systems earlier in the academic year has the potential to improve the quality of the systems if a software development methodology is utilized that supports breaking the SDP into manageable tasks.

The following eight lessons learned to mitigate the causes (*cf.* Fig. 2) of software quality are provided with further reflections from the authors:

- *Improve requirements identification by providing a suitable methodology for your students to follow*. We found that our adaptation of the agile methodology with scrum as our intervention provided better software deliverables then we had experienced in previous years of running the course. The primary reason can be attributed to the regular scrum sessions with the teaching assistant in the role as Scrum Master where students were consistently required to interrogate the requirements for that sprint. Additionally, we noted that requiring a system output for each sprint improved the quality of the coded system.
- *Improve team dynamics by providing team-building training at the commencement of the project for students*. We have learned over the years that assuming that students will get along as teams is very short-sighted, especially if students are assigned to groups and are not able to choose to work with friends or acquaintances. Therefore, we complete team building exercises in time management, conflict resolution and communication methods to prepare the students for engaging in a team environment. Additionally, we reinforce these soft skills throughout the SDP course and provide opportunities throughout the academic year for students to voice their concerns about their team members through buddy ratings (where students anonymously rate themselves and their team members) and red flags (where a member of a team can request an intervention in the team). In both cases the project teams are provided with assistance to try and resolve the presenting problem. We have found that those groups who have better interpersonal skills are more likely to produce better quality systems, as they are working together as a cohesive unit.
- *Improve the scheduling of meetings, venues and deliverables by making these visible and known*. Every effort is made to make students aware of project deadlines through reminders on noticeboards and communication via the eLearning System being used. Additionally, we found that having regular scrum sessions meant that we needed to provide a venue scheduling system to avoid clashes between the project groups over the limited meeting venues available. Linked to that was ensuring that project groups arrived promptly for meetings and

did not let them run over their scheduled time-slot. Where groups did not meet during their scheduled times for the scrums we found that the quality of their sprint deliverables was negatively impacted, which in turn impacted on the quality of their final system. The groups who also did not meet in their own capacity on a regular basis also reported that they had significant time management issues with trying to complete the project deliverables.

- *Improve problem-solving skills by giving project groups a case study to complete as a team prior to starting the sprints*. We have found that students are stuck in a mantra of 'there is only one possible answer', which is problematic when developing software solutions. By giving them a case study (with no single answer) before they start their sprints we challenge them to 'think outside the box. We carry this approach through to the scrum sessions where the teaching assistant is counselled to not provide solutions, but to make suggestions, or challenge project groups to motivate why they have made a given decision. From the teaching assistant's notes we determined that those groups that did not engage during the scrum session tended to produce systems that had problems in their business processes, which impacted on the quality of the final system.
- *Improve information systems literacy by making students aware of what they already know*. Because of the silo effect to studying mentioned earlier in this paper we find that we constantly have to remind students what they already know and provide additional resources for further exploration. Our eLearning System is used extensively throughout the academic year to provide the project groups with sources of knowledge and reference. During focus group sessions with the groups and feedback sessions with the groups we determined that the project groups are seldom challenged themselves by learning new techniques, and tried to develop the systems with only what they had been taught previously. The project groups often reported in the focus group sessions that they abandoned a feature for their system, because it had not been explicitly taught to them. Removing a feature from the systems compromised the business logic of the system in many instances and thereby impacted on the quality of the system.
- *Improve software testing by project groups by making them present a database prototype of the system as a deliverable early in the SDP*. We have found that getting project groups to present their systems for testing is problematic, but have realized that this can in part be attributed to the problems that we experienced with respect to the database builds by the project groups. Our assumption is that if the foundation (database) is solid then the rest of the system quality should improve. We found that the project groups were seldom ready for the testing sessions and when we asked the project groups for feedback they informed us that they did not understand the need for the testing. The lack of testing on the part of the project groups had a direct impact on the quality of the system, as errors in coding and business logic could not be identified timeously.
- *Improve the logic of the user interface design (UID) by providing students with additional resources and training on best practices*. We are sure that every academic who has ever supervised a software project module will have encountered the psychedelic mixture of screen colors which left their eyes seeing sun spots. Added to this is a navigation system that needs a training manual of its own. Early in the project we advise the project groups to choose a color and navigation scheme for their software application and to source similar systems online to determine how they have been designed. Additionally, we provide additional resources on our eLearning System as well as contact sessions on UID during the academic year. We asked the project groups to present screen shots of their screens to 2nd year students and get their feedback. The project groups reported on the feedback, but did not all action the constructive criticism of the 2nd year groups. Those groups that did not make the changes reported that they did not do so because they felt that the 2nd years did not really understand how the system worked. The inability to action the constructive criticism about the user interface design was found to impact negatively on the quality of the final system.
- *Improve the coding practices of project groups by providing them with a set of best practices*. We often find that project groups do not actively comment on their code or make optimal use of their code. With the assistance of the programming lecturers in our department a set of best practices for coding was developed from the problems encountered while conducting the code review sessions during the sprints. The assessment marks collected during the code review sessions and the comments collected from the secondary lecturers indicated that the project groups were not reviewing the best practices provided. The implication was that the project groups made unnecessary errors in their coding, which impacted on the final quality of the system.

These lessons learned and the shortcoming identified can be used to develop a further action research intervention to address software quality in SDPs.

7 Conclusion

The success of this action research project is evidenced by the continued improvement in the systems developed as a result of the systems development project. The ability to move away from a traditional waterfall methodology to an agile with scrum methodology is connected to the success of this study. The key to continued improvement of student projects is to ensure that, as academics, we are actively reflecting on the process and outcome throughout the stages of the SDP.

Since the completion of the last cycle of the action research process undertaken for this study, the impact on the SDP course and students remains positive. While the lessons learned have identified some shortcomings in the intervention that was implemented during the last cycle of the action research process, these can be addressed by refining the current intervention. The refined intervention can then be implemented in a new cycle of the action research process. The primary goal of new action research cycle will focus on determining the optimal *software development methodology to ensure that the presented client problem (opportunity) is developed into a quality software solution.* In doing so, the SDP will be able to provide the students with a relevant and worthwhile learning experience.

References

1. Hine, G. S. (2013). The importance of action research in teacher education programs. *Issues in Educational Research*, 23(2), 151-163.
2. Norton, L. S. (2009). *Action Research in Teaching and Learning – A practical guide to conducting pedagogical research in universities*. New York: Routledge
3. del Carmen Salazar, M. (2013). A Humanizing Pedagogy. *Review of Research in Education*, 37, 121-148.
4. Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker Jr, J. F., Sipior, J. C., & de Vreede, G. J. (2010). IS 2010: Curriculum guidelines for undergraduate degree programs in information systems. *Communications of the Association for Information Systems*, 26(1), 18.
5. Shaw, M. & Tomayko, J. (1991): Models for Undergraduate Project Courses in Software Engineering. *CMU/SEI-91-TR-10 Technical Report*. SEI.
6. Calitz, A.P., Greyling, J.H., & Cullen, M. (2010). The South African ICT works environments and graduate ICT skill requirements. *International Business Conference (IBC) 2010*, Victoria Falls, Zambia.
7. Lotriet, H. H., Matthee, M. C., & Alexander, P. M. (2010). Challenges in ascertaining ICT skills requirements in South Africa. *South African Computer Journal*, 2010 (46), 38-48.
8. Schön, D. A. (1983) *The Reflective Practitioner. How professionals think in action*, London: Temple Smith.
9. Altrichter, H., Feldman, A., Posch, P., & Somekh, B. (2013). *Teachers investigate their work: An introduction to action research across the professions*. London: Routledge.
10. Mertler, C.A. (2009). Action Research: Teachers as researchers in the classroom, 2ed. Sage Publications: Thousand Oaks, California
11. Carr, W. & Kemmis, S. (1996/2004). *Becoming critical*. London: Falmer Press.
12. Gibbs, G. (1995). Changing lecturers' conceptions of teaching and learning through action research. In A. Brew, (Ed.), *Directions in staff development* (pp. 17–20). Buckingham: SHRE and Open University Press
13. Sagor, R. (2005). *The Action Research Guidebook*. California: Corwin Press.
14. Mertler, C. A. (2016). Leading and facilitating educational change through action research learning communities. *Journal of Ethical Educational Leadership*, 3(3), 1-11.
15. Brydon-Miller, M., Greenwood, D., & Maguire, P. (2003). Why action research?. *Action Research*, 1(1), 9-28.
16. Altrichter, H., Posch, P. and Somekh, B. (2005). *Teachers investigate their work*. London: Routledge
17. Jahangiri, L., & Mucciolo, T. (2012). *A guide to better teaching: skills, advice, and evaluation for college and university professors*. London: Rowman & Littlefield Publishers, Inc.
18. Stenhouse, L. (1975). *An introduction to curriculum research and development*. London: Heinemann.
19. Elliott, J. (2015). Educational action research as the quest for virtue in teaching. Educational Action Research, 23(1), 4-21.
20. O'Neill, G. (2010). Initiating Curriculum Revision: Exploring the practices of educational developers. *International Journal of Academic Development*, 15(1), 61-71.
21. Sutherland, J. (2015). SCRUM: The art of doing twice the work in half the time. London: Random House Business Books.
22. Kember, D. (2000). *Action learning and action research: Improving the quality of teaching and learning*. Psychology Press.
23. Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213-1221.
24. Fitzgerald, B., & Stol, K. J. (2014, June). Continuous software engineering and beyond: trends and challenges. *In Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering* (pp. 1-9). ACM.

Evaluating Alumni Satisfaction of an CS/IS Department

Andre P. Calitz[1], Jean H. Greyling[1] and Arthur Glaum[2]

Nelson Mandela University
[1]Dept of Computing Sciences, [2]Business School
Port Elizabeth, South Africa
Andre.calitz@nmmu.ac.za,Jean.greyling@nmmu.ac.za, Arthur.glaum@gmail.com

Abstract. The concepts of a service guarantee, customer satisfaction and perceived value of a service being offered in education are being applied to Alumni satisfaction at Higher Education Institutions (HEIs). Businesses offer service guarantees, however HEIs rarely guarantee the service they offer to students. Literature indicates that offering a service guarantee will increase consumer satisfaction. Service Guarantees in Education have increasingly being introduced by various departments, specifically at post-graduate level when signing service level agreements and learning agreement/contracts. In the academic environment, customer/student satisfaction and perceived value further positively influence graduate/Alumni satisfaction.

The purpose of this research study is to measure the perception and satisfaction of the Alumni of the Department of Computing Sciences at the Nelson Mandela University (NMU) and identify areas for improvement by performing a systematic analysis of the determinants of satisfaction. This research is an exploratory, quantitative study consisting of literature- and case-study components used to test proposed hypotheses. The key concepts related to the topics of Service/Teaching Guarantee, Customer satisfaction and Perceived Value were investigated. The empirical study consisted of a survey completed by the Alumni of the Department of Computing Sciences. A model identified the following factors as having an influence on *Alumni Satisfaction* with the NMU Department of Computing Sciences: *Service/Teaching Guarantee, Customer Satisfaction* and *Perceived Value*. The importance of each factor was identified to understand how to improve the Alumni perceptions and satisfaction with the Department of Computing Sciences.

Keywords: Alumni Satisfaction, Service Guarantee, Perceived Value, Customer Satisfaction.

1 Introduction

The success of courses offered by a university and the effectiveness of the instructors has commonly been measured by student evaluations [22, 29, 38, 39]. The Alumni perception of the extent of learning and the usefulness of the knowledge is a key measure for universities [22, 29, 38]. McCollough and Gremler [37] define a service guarantee as a formal promise made to customers about the service they will receive. Similarly, Kashyap [28], McColl and Mattsson [36] and Sum, Lee, Hays and Hill [51] define a service guarantee as a written promise made by the company through advertising or company literature that will provide compensation if promises are broken.

Stakeholder theory suggests that the purpose of a business is to create as much value as possible for stakeholders and not only for the shareholders. A business should try to meet the needs of everyone that has a stake in both the actions and outcomes of the organisation. In order to be sustainable, the executives must keep in mind the interests of all the stakeholders, such as the customers, suppliers, employees, communities and shareholders, ensuring that is aligned [4, 21, 41].

Alumni are seen as the most important assets of a university. They are important assets as the university is represented in the real world by the Alumni [7, 45, 55]. Historically, Alumni networks were created from regional groups brought together for fundraising purposes. Over time these networks developed, both in their own importance and because the university gained benefits from these networks for the development of the university and thus created an enormous outreach potential for current students in their career paths.

This paper provides additional supporting evidence regarding Alumni perceptions on the value of a service guarantee, customer satisfaction and perceived value of a service being offered in Information Technology (IT)

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

education. The results highlight the importance of Alumni satisfaction relating to the provision of service guarantees, customer satisfaction and perceived value of an academic programme. The research problem and research objective are discussed in Section 2 and Section 3 discusses stakeholder theory. The literature on Alumni satisfaction, service guarantees and perceived value are presented in Section 4. The research methodology (Section 5) and the Alumni survey results are presented (Section 6), followed by conclusions and recommendations relevant to Computer Science (CS), Information Systems (IS) and IT Departments at Higher Education Institutions (HEIs) and future work are discussed in Section 7.

2 Problem Statement

Information Technology (IT) is offered at 26 universities in South Africa by various departments offering IT-related study programmes [6]. In this study, the two disciplines Computer Science (CS) and Information Systems (IS) are grouped together in the Department of Computing Sciences. Alumni surveys are not conducted on a regular basis by the CS and IS departments in South Africa and therefore they do not fully understand the graduate perceptions of the studies offered at their academic institutions.

Alumni are seen as important stakeholders by the Department of Computing Sciences, since they can provide valuable information in order to evaluate and improve programme offerings. There are various programme levels in the Department of Computing Sciences at NMU, starting with the undergraduate students who can complete an undergraduate programme. Examples of undergraduate programmes are BSc in Computer Science or a BCom in Information Systems. Postgraduate programmes on offer are BCom Honours, BSc Honours, MCom, MSc or PhD in Computer Science and Information Systems. The Alumni engaged with postgraduate courses have to work with a supervisor for an honours treatise, master’s dissertation or PhD thesis [5, 6]. This study focused on both the graduate Alumni and postgraduate Alumni in the Department of Computing Sciences at NMU.

The problem statement investigated in this study is that the *Department of Computing Sciences at Nelson Mandela University (NMU) has not considered Alumni perceptions of their study experience at NMU*. The main research objective of this study is *to measure the perception and satisfaction of Alumni of the Department of Computing Sciences and identify areas for improvement*. The Alumni of the department are those who completed their studies at the NMU who can provide direct feedback of their experiences while studying.

3 Stakeholder Theory

The Stakeholders are defined as the individuals or groups who are either affected by the activities and consequences of an organisation or those who have an influence on the organisation [4, 41]. However, this view can be seen to include almost anyone as a stakeholder in the organisation. The stakeholders are therefore narrowed down to the most important, affected individuals or groups. The individuals or groups whose interests are far-removed from the operations of organisations are excluded from the stakeholder theory. If these far-removed individuals or groups are considered as stakeholders, they will have differing interests which will place a financial burden on the organisation and it might not survive [24, 41, 54].

The organisations that take their stakeholders’ interests seriously are regarded as more successful than the organisations that do not. The relationship between the performance of the organisation and stakeholder strategies have been researched and a positive relationship has been found while holding other variables constant. The success of organisations with stakeholder management is stability, growth and profitability [8].

Governments are making substantial cuts in public spending and its contribution to university education declined by 9% from 2000 to 2012 while the burden on students increased by 7% during the same period [44]. This is one of the factors that has caused public education institutions to re-evaluate their relationships with stakeholders in order to remain competitive [43, 47]. Other factors that drive universities to be more competitive are increasing competition due to globalisation and an awareness of international rankings [12, 47]. In this perspective, Alumni are key university stakeholders and forming a strong bond with them is a priority for management of higher education institutions [4, 6, 23, 47].

In the literature, students are regarded as customers of HEIs [12]. The bond formed with students endures after students have graduated and become Alumni. The loyalty that is experienced by students is relevant both during the period when the student was enrolled as an undergraduate at the institution as well as in the period after the student had completed his or her formal education [47]. The benefits of having a strong body of Alumni are additional financial resources through donations to research funding, mentoring, word of mouth and supervising enrolled students [23, 47].

4 Alumni satisfaction

The concepts of service guarantee, customer satisfaction and perceived value in education are being applied to Alumni satisfaction at Higher Education Institutions (HEIs). McCollough and Gremler [37] define a *service guarantee* as a formal promise made to customers about the service they will receive. Similarly, Kashyap [28], McColl and Mattsson [36] and Sum, Lee, Hays and Hill [51] define a service guarantee as a written promise made by the company through advertising or company literature that will provide compensation if promises are broken.

In order to ensure customer satisfaction in a service offering, the quality of service performance needs to be guaranteed [36, 53]. Many service industries promise a level of service performance, with or without conditions attached. Examples of these service industries are retailing, real estate, fast food, airline, telecommunication, transport and leisure as well as professional and financial services and education [36].

An implicit assumption in the services guarantee literature is that offering a service guarantee will increase *consumer satisfaction* [13, 48, 53]. Service guarantees give rise to increased customer satisfaction by setting clear standards with guarantees [53]; feedback is generated from a guarantee [7, 18, 45, 49, 57]; failure is measured against a guarantee creating greater understanding [37, 53]; and there are marketing benefits from guarantees as they reduce purchasing risk and enhance the loyalty of existing customers [37].

Guaranteeing the quality of service performance in a service offering ensures customer satisfaction. The perception of guarantees is that higher guarantees are signals of higher quality and lower guarantees are signals of lower quality [2, 26 28, 36, 53]. Based on customer perception, implementing a service guarantee can either be a positive or negative signal. However, Hogreve and Gremler [25] suggest that providing the service guarantee will result in internal changes, which will improve quality and ultimately improve customer perception and the value of the guarantee. Service Guarantees in Education have increasingly being introduced by various departments by signing service level agreements and learning agreement/contracts. These contracts indicate student and supervisor responsibilities and behaviour requirements.

Unlike businesses, Universities rarely guarantee the service they offer to students. One of the reasons identified is because services, especially education, are seen as the most intangible of products [2, 25, 37]. This intangible quality of education is seen as riskier and failure is more common than with product failure. Firstly, if failure is more common, then service guarantees are also perceived to be more expensive. Secondly, the design and drafting of a service guarantee is more challenging. Lastly, due to co-production it is difficult to separate the roles and responsibilities of the consumer and service provider [37].

Prospective employers of students from HEIs have been offered guarantees that students have understood certain concepts and acquired specified skills. The students of these institutions were essentially treated as products and not as customers. Professor Richard Chase of the University of Southern California was one of the first instructors to offer a broad-based university classroom guarantee that treated students as customers [22, 25, 37]. In the research done by McCollough and Gremler [37], they believe that students should be offered a service guarantee, but are often challenged by their peers and colleagues as to why this is important. In order to understand the significance of offering a service guarantee in education, two theories of guarantees are important, namely: market signalling theory and investment theory.

The *market signalling theory* is used by firms in order to signal to customers the superior quality of their product [11, 17, 37]. Signalling theory originates from the study of information economics and it contemplates a situation where buyers and sellers have different views on the quality of a product or a service. The assumption made by the signalling theory is that the seller of the product has a perfect knowledge of the quality of the product, whereas the knowledge of the customer is inadequate [11]. An example of this is that the quality of many products and particularly services are only known to the customer after consumption. Sellers with high-quality products looking for a competitive advantage distinguish themselves from sellers of low-quality products by sending a pre-purchase marketing signal [11, 17, 37].

One of the methods used to signal the quality of a product to a customer is by using a guarantee [17]. In education, universities with highly intangible products can use a guarantee to signal the teaching quality on offer to a customer. Students can be signalled about the high quality of the classes by being offered a teaching guarantee. This is even more relevant when offering electives which are dependent on sufficient student enrolments [37].

Investment theory interprets a guarantee as an insurance and repair contract that extends the life of a product. The choice lies with the consumer who can purchase a product with or without a guarantee [2]. The product that is offered with a guarantee should be sold at a premium to cover the cost of warranty claims. The choice the consumer has to make is to weigh up the benefit of having a product which is covered by a guarantee, or a lower cost product without a guarantee where the customer carries the risk.

The consumer who decides to carry the risk of product failure has more incentive to keep the product maintained and well looked after. On the other hand, the consumer who selects the product with a guarantee is opting to pay higher up-front cost, but may not be incentivised to maintain the product. Erevelles [16], McCollough and Gremler [37] have empirical support for the investment theory. The investment theory views a guarantee as a risk management instrument. In a university where a service is provided, the student might perceive risk in enrolling for a class and offering a guarantee might be an effective tool in reducing the risk [2, 18, 37].

As the costs of education increase and in most cases, students have to pay for a portion or the full amount of their education, the view of treating students like customers is likely to increase. The evaluation of a university and its instructors is carrying increasing weight for students in their selection of tertiary education [14, 37]. The offer of a teaching guarantee can entice students to enrol with a specific university and help to adopt a student-as-customer focus.

A guarantee offered by a university can generate feedback and provide the necessary insight into the causes of dissatisfaction and failure [37, 50]. Instructors can learn from students who invoke guarantees and adapt a programme to better service future students. The increasing focus on student evaluations and teaching quality highlights the need to understand student dissatisfaction that can be better understood by offering a service guarantee.

Another advantage for universities using service guarantees is that they build *marketing muscle* [37]. This is a rare guarantee that would be difficult to duplicate, valuable for customers and will provide a sustainable competitive advantage. In general, universities have not adopted a competitive strategy, but as funding from government decreases, this may be changing. A service guarantee also provides a competitive advantage by reducing risk and signalling quality [11, 17, 25]. A teaching guarantee is an effective tool to compete for students and gain a competitive advantage over rival universities. *Service/Teaching Guarantees* have been identified and have been proposed to have a relationship with the independent variable, *Alumni satisfaction*, as depicted in Figure 1.

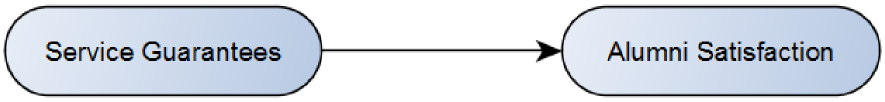


Figure 1: Relationship of Service/Teaching Guarantees on Alumni satisfaction

Customer Satisfaction has been identified and has been proposed to have a relationship with the independent variable, *Alumni satisfaction*, as depicted in Figure 2.

Customer Satisfaction
Alumni Satisfaction

Figure 2: Relationship of Customer Satisfaction to Alumni satisfaction

The conventional method of listening to the customers' needs has been achieved through measuring customer satisfaction [35]. However, Woodruff [56] states that there needs to be a shift from measuring satisfaction towards a better understanding of what customers value. Having a better understanding of what customers' value will help firms to achieve their organisational purpose and goals. This has resulted in an extended view of customer-perceived value which has drawn the attention of researchers [13, 20, 31, 46, 49, 50, 57]. Based on this extended view the focus on customer value then changes to a customer-oriented concept. A business that is customer-oriented must consider the perception of the customer when it defines the value proposition of the business [35].

The *perceived value* of a product or a service will vary from one customer to the next. Tailoring a product or service to a wide audience will result in some customers valuing a certain aspect more when compared to other customers. The value of a product or service will also change over time as customers' needs and wants change. The perceived value by the customer of the product or service on offer is difficult to measure [3, 9, 13, 35, 50].

Alumni are seen as the most important assets of a university. They are important assets as the university is represented in the real world by the Alumni [7, 45, 55]. The achievements of Alumni directly reflect on the university and any improvements to the quality of the education at the university automatically improves the perceived value of the graduate's degree [15]. A synergistic relationship is created by this connection between the Alumni and the university. Alumni are a university's best ambassadors and should be kept informed and involved in the vision and priorities of the university. Alumni bring real-world experience to the modern-day student and increase the perceived value that the university offers [1, 15, 18].

Perceived Value has been identified and it has been proposed that it has a relationship with the independent variable, *Alumni satisfaction*, as depicted in Figure 3.

Perceived Value
Alumni Satisfaction

Figure 3: Relationship of Perceived Value on Alumni satisfaction

In order to test the relationship between the Dependent and Independent Variable the following hypotheses have been formulated:

H_1 = "*Service/Teaching Guarantees* are significantly related to *Alumni Satisfaction*";

H_2 = "*Customer Satisfaction* is significantly related to *Alumni Satisfaction*"; and

H_3 = "*Perceived Value* is significantly related to *Alumni Satisfaction*".

5 Research Methodology

A previous study amongst Alumni Postgraduate students only (n=39), was conducted by the NMU Department of Computing Sciences in 2016 [6]. A quantitative survey was identified in the future research in order to construct a conceptual model on Alumni perceptions of an academic department's offerings.

The questionnaire utilised in this study was divided into seven sections consisting of 50 statements/questions in total. Section 1 of the questionnaire captured the biographical details of the respondents such as *Gender, Age, Education Level, Geographical location and level of education*. Section B through to Section E were designed to capture the respondent's perception *of Service/Teaching Guarantees, Customer Satisfaction* and *Perceived Value* all relating to the Alumni perception. These questions were rated with a five point Likert scale (1 = Strongly Disagree and 5 = Strongly Agree). This section measured a total of 4 variables and each variable was measured using between 6 and 8 items. The sections on the use of social media are not included in this paper.

The primary data were collected using an on-line questionnaire in order to collect responses from the sample group. An email containing a Universal Resource Link (URL) to the questionnaire was sent to the listed email addresses of the Alumni of the Department of Computing Sciences at NMU which contains more than 600 potential respondents. The URL was also posted on the Facebook page of the Department of Computing Sciences. The potential respondents were reminded to respond three times after which a total of 100 responses were received.

The sampling method employed in this study incorporated non-probability sampling in the form of purpose sampling and snowball sampling as the respondents were chosen from the Alumni database of the Department of Computing Sciences at NMU. All respondents furthermore had an equal opportunity of being selected for the study. A representative sample of 100 respondents from this population completed the survey.

Accepting or rejecting the hypotheses proposed in the study by using statistical analysis of gathered numeric data is the purpose of this study [34]. By using statistical analysis, it can be accepted that this is an empirical study and quantitative in nature. A statistician sorted, categorised and cleaned the quantitative data in order to facilitate this purpose. A computer software package named Statistica was used in order to perform quantitative statistical analysis on the data. The analyses performed on the data included descriptive statistics such as measures of central tendency being the mean, median and the mode and inferential statistics specifically using the Pearson's Chi-Square analysis.s.

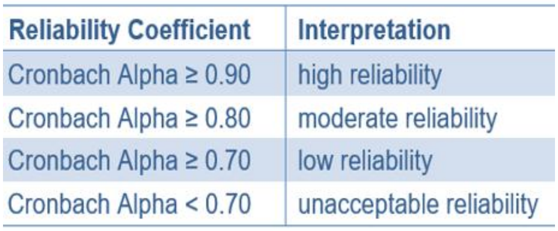
5.1 Reliability and Validity

Reliability is an indicator of the level of consistency that can be attached to a measurement instrument in its capability to correctly measure the variables under investigation [10, 30, 34]. A measurement is said to be reliable when repeated measurements of an unchanged entity return the same result each time [32]. Reliability is vital for positivistic studies. There are two techniques used when measuring reliability namely test and retest reliability. Test reliability is when applying measures of internal consistency; retest reliability is repeating an event to determine if the same or similar results are recorded [27].

A statistical technique named Cronbach Alpha is a common and well-used internal consistency reliability measure in quantitative research. This statistical technique is applied to questions in a questionnaire to statistically determine how reliably they measure predetermined variables [52]. In this study, Cronbach's alpha was used to determine the reliability of the measuring instrument that provided measures of internal consistency reliability.

The Cronbach’s alpha coefficient was used to measure this internal consistency. This coefficient of reliability ranges from 0 to 1 in providing the overall assessment of a measure's reliability. A low coefficient value indicates a low internal consistency while a high value indicates the opposite. For research, the guidelines for reliability coefficient are defined in Table 1.

Table 1. Cronbach’s Alpha Coefficient [10, 40].



Reliability Coefficient	Interpretation
Cronbach Alpha ≥ 0.90	high reliability
Cronbach Alpha ≥ 0.80	moderate reliability
Cronbach Alpha ≥ 0.70	low reliability
Cronbach Alpha < 0.70	unacceptable reliability

A Cronbach’s alpha value of between 0.50 and 0.69 is deemed acceptable for new and experimental research [10, 40]. In order to contribute to the existing body of knowledge, an important consideration in research conclusions is validity. To be valid, the variables measured must accurately be reflected in a manner that allows applications outside of the research environment [34].

In order to describe and summarise the data, descriptive statistics were used. Three types of measures of central tendency were used to describe the data namely the mean, median and mode. This study further tested the relationships between the independent variables, Service/Teaching Guarantees, Customer Satisfaction, Perceived Value and the dependent variable Alumni Satisfaction by applying and analysing the results of the Pearson’s Chi-Square test. The Pearson’s Chi-squared test was used in this study to test for a significant difference between the correlations of independent variables to the dependent variable between two sample data sets.

One of the statistical methods commonly used by researchers to confirm or negate conclusions is correlation analysis. Correlation can be defined as relationships amongst variables or measures of linear association between two variables. The correlation coefficient measures the strength of such a relationship [19]. This correlation coefficient (r) can range from -1 (a perfect negative correlation) to +1 (a perfect positive correlation).

In order to indicate the standardised difference between two means, Cohen's d is used as an effect size. The results of ANOVA and t-tests are often accompanied by Cohen's d. The comparison between two means is appropriately done using Cohen's d as an effect size [33].   2 indicates the interpretation intervals for Cohen's d.

Table 2. Interpretation intervals for Cohen's d

Cohen's d	Interpretation
<0.20	Not significant
0.20 - 0.49	Small significance
0.50 - 0.79	Medium significance
0.80+	Large significance

6 Results of the study

As indicated in Section 4, 100 respondents finally participated. The results indicated that 82% of the respondents were male while only 18% were female (Figure 1). It has been well researched and documented that females are under represented in Computing Sciences [6]. In 2014 only 18% of graduate degrees were awarded to women for Computer Science. This can also be observed in industry where only 18% of Google’s technical employees are female [5].

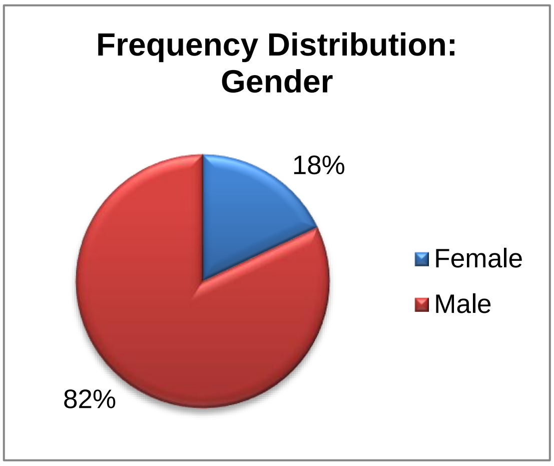


Figure 1. Frequency Distribution of Gender

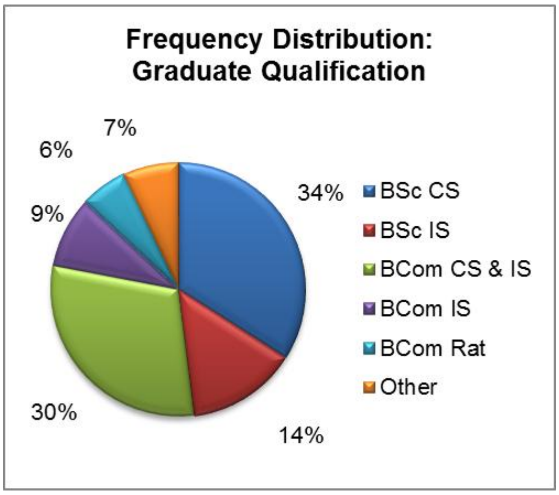


Figure 2. Graduate qualification

As seen in Figure 2, the most of the respondents qualified in BSc CS (34%), 14% in BSc IS, 30% in BCom CS & IS, 9% in BCom IS, 6% in BCom Rat and 7% selected Other.

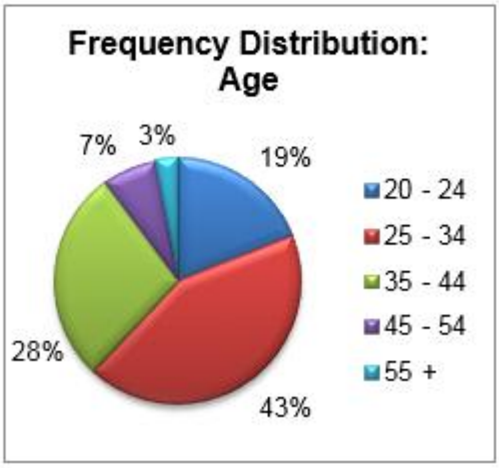


Figure 3. Frequency Distribution of Age

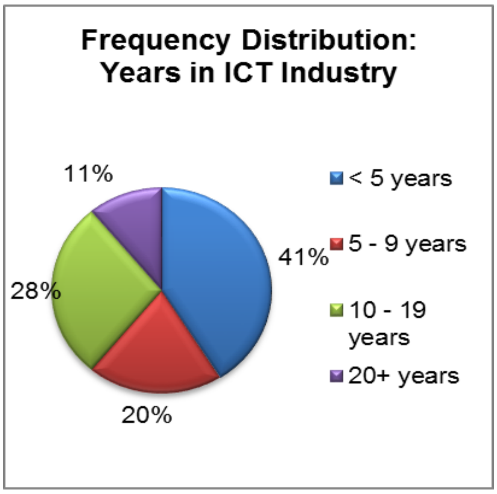


Figure 4. Years in IT Industry

The results, as indicated in Figure 3, show that 19% of the respondents are between the ages of 20 – 24 years of age, 43% between 25 -34, 28% between 35 - 44, 7% between 45 - 54, and 3% are 55+ years of age. As can be seen, there is a relatively prominent age group, 25 – 34 years of age, with the respondents of the study.

Regarding years worked in the industry (Figure 4), the results show that 41% of the respondents have fewer than 5 years' experience in ICT, 20% between 5 - 9 years, 28% between 10 - 19 years and 11% 20 years and more experience in ICT. The responses received covered a wide range of ICT experience that provides valuable feedback on the Alumni of the Department of Computing Sciences and the perception over various generations.

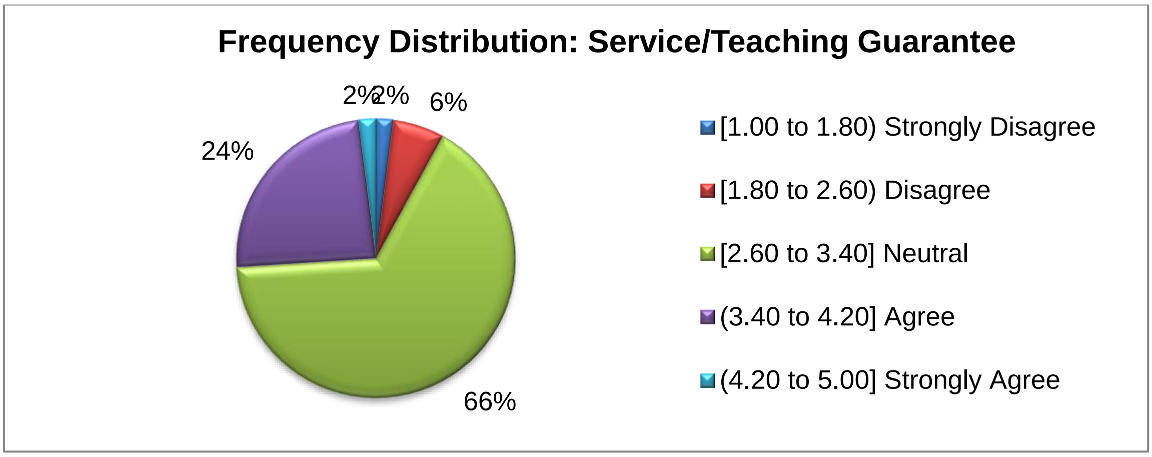


Figure 5. Frequency Distribution of Service/Teaching Guarantee

Figure 5 illustrates the frequency of responses received relating to positive statements made regarding a Service/Teaching Guarantee. A small proportion of responses were positive with 2% strongly agreeing and 24% agreeing. The majority of responses, 66%, were neutral and a very small proportion disagreed (6%) or strongly disagreed (2%) with the statements in the questionnaire. The Cronbach alpha for this factor is 0.69 (Table 3), which is unacceptable reliability for this construct. The reliability for the Service/Teaching Guarantee, however, can still be considered acceptable for exploratory research.

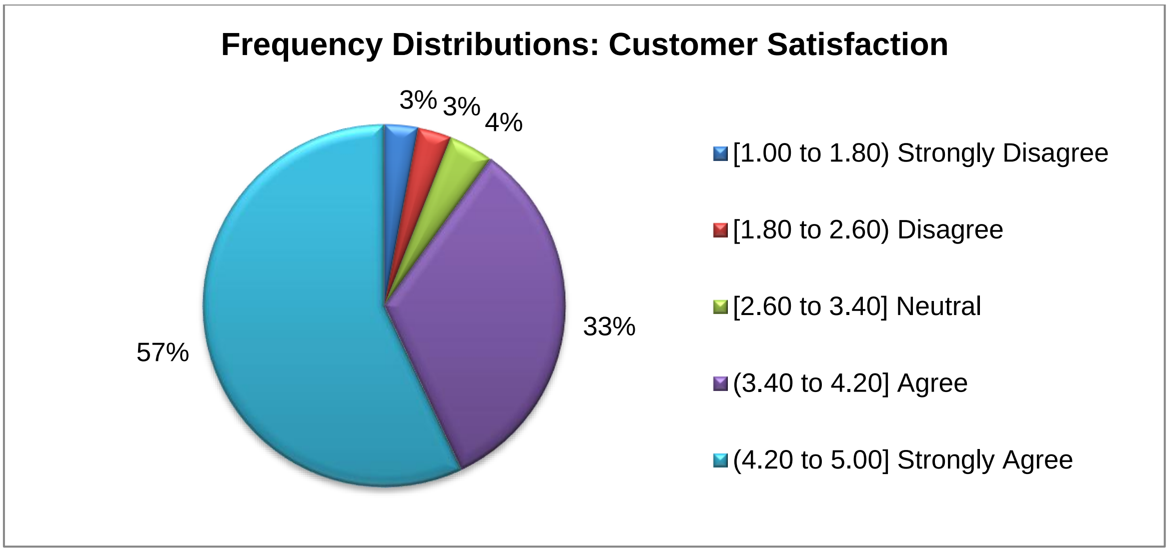


Figure 6. Frequency Distributions of Customer Satisfaction

Figure 6 illustrates the total responses received that relate to Customer Satisfaction. The majority of responses were positive with 57% strongly agreeing and 33% agreeing with statements associated to Customer Satisfaction in the Department of Computing Sciences. Only 4% of the responses were neutral and even a smaller percentage, 3%, disagreed or strongly disagreed, 3%. The Cronbach alpha for this factor is 0.97 (Table 3), indicating high reliability for this construct.

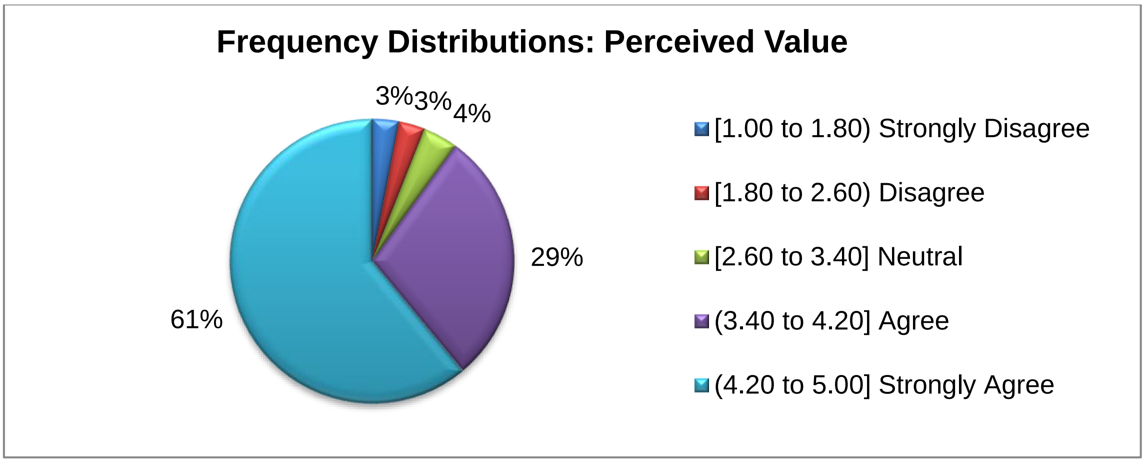


Figure 7. Frequency Distributions of Perceived Value

Figure 7 illustrates the frequency distribution that relates to Perceived Value. The majority of responses were positive with 61% strongly agreeing and 29% agreeing with statements associated to the Perceived Value in the Department of Computing Sciences. Only 4% of the responses were neutral and even a smaller percentage, 3%, disagreed or strongly disagreed, 3%. The Cronbach alpha for this factor is 0.84, indicating a moderate reliability for this construct (Table 3).

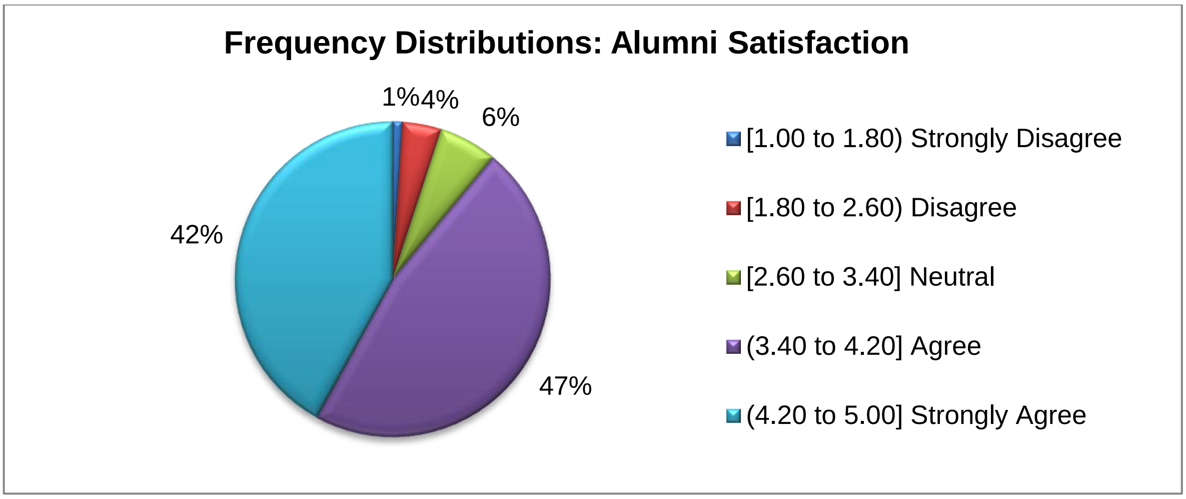


Figure 8. Frequency Distributions of Alumni Satisfaction

Figure 8 illustrates the total responses received which all relate to Alumni Satisfaction. Responses were mainly positive with 42% strongly agreeing and 47% agreeing with statements related to the Alumni Satisfaction of the Department of Computing Sciences. Only 6% of the responses were neutral and even a smaller percentage, 4%, disagreed or strongly disagreed, 1% with the statements in the questionnaire. The Cronbach alpha for this factor is 0.91, indicating high reliability for this construct.

Regarding Service/Teaching Guarantee, a Cronbach alpha value of between 0.50 and 0.69 has been deemed acceptable for new and experimental research [10, 40]. The internal reliability for all other measuring instruments is sufficient, ranging from 0.84 to 0.97. These values are higher than the minimum requirement of 0.70.

Table 3. Cronbach's alpha coefficients for the factors (n = 100)

Factor	Cronbach Alpha
Perceived Value	0.84
ALUMNI Satisfaction	0.91
Customer Satisfaction	0.97
Service/Teaching Guarantee	0.69

A conceptual framework was constructed by the authors based on the literature study. The conceptual framework was used to establish relationships between the Dependent variable, *Alumni Satisfaction*, and the independent variables, *Service Guarantees*, *Customer Satisfaction and Perceived Value.* Table 4 shows these hypotheses, the relevant Pearson Correlation, the correlation strength as well as whether the hypothesis is accepted or rejected.

Table 4. Hypotheses Testing

Hypothesis	Hypothesis Description	Pearson Correlations	Correlation Strength	Hypothesis Accepted or Rejected
H_1	*Service/Teaching Guarantees are significantly related to Alumni Satisfaction*	0.204	Low positive correlation	Rejected
H_2	*Customer Satisfaction is significantly related to Alumni Satisfaction*	0.881	High positive	Accepted
H_8	*Perceived Value is significantly related to Alumni Satisfaction*	0.862	High positive	Accepted

The hypothesised model with the proposed relationships as shown in Figure 9, was tested using Pearson Correlations. Two out of the three hypotheses developed in this research study were accepted by means of statistical analysis through empirical evaluation. The model therefore needs to be adjusted by only removing H_1 as an independent variable of *Alumni Satisfaction*. The correlations in this study proved to be both statistically and practically significant at a 0.05 confidence level when r_{crit} is bigger or equal to 0.300 for all correlations. As shown in Table 5 there are very high positive correlations between *Customer Satisfaction and Perceived Value* with *Alumni Satisfaction.*

Table 5. Pearson Product Moment Correlations with Alumni Satisfaction (n = 100)

Factor	Pearson Correlations	r_{crit}	Correlation Strength
Customer Satisfaction	0.881	0.300	High positive
Perceived Value	0.862	0.300	High positive
Service Guarantees	0.204	0.300	Low positive correlation

Error! Reference source not found. Figure 9 illustrates the correlated strengths of relationships between the dependent variable, *Alumni Satisfaction* and the independent variables, *Service Guarantees*, *Customer Satisfaction and Perceived Value.* An important finding is the strong positive relationship (*r* = 0.881) between *Customer Satisfaction* and *Alumni Satisfaction.* A student, as a satisfied customer, becomes a satisfied alumnus. This relationship is expected to have a strong correlation as both are related to satisfaction. This finding supports the reviewed literature in Section 3 on *Customer Satisfaction.*

The *Perceived Value* on offer at the Department of Computer Sciences also has a strong positive relationship (*r* = 0.862) with *Alumni Satisfaction.* This finding supports the reviewed literature where the achievements of Alumni directly reflect on the university and any improvements to the quality of the education at the university automatically improves the perceived value of the graduate's degree [15]. A synergistic relationship is created by this connection between the Alumni and the university. Alumni bring real- world experience to the modern-day student and increase the perceived value that the university offers [1, 15, 18].

Service Guarantees have a low positive correlation (*r* = 0.204) and any change in this factor is unlikely to influence *Alumni Satisfaction.* The low positive correlation is in contrast with the knowledge gained from the literature study as discussed in Section 3. This concept in education is not well known to the Alumni and many questions were asked during the survey about this concept. The respondents feel that this will add no real value to the *Alumni Satisfaction.* From the literature, *Service Guarantees* offered by a university can still generate feedback if there is dissatisfaction and failure [37, 50]. Instructors can learn from students and adapt the programme to better service future students who invoke guarantees. The increasing focus on student evaluations and teaching quality highlights the need to understand student dissatisfaction that can be better understood by offering a service guarantee.

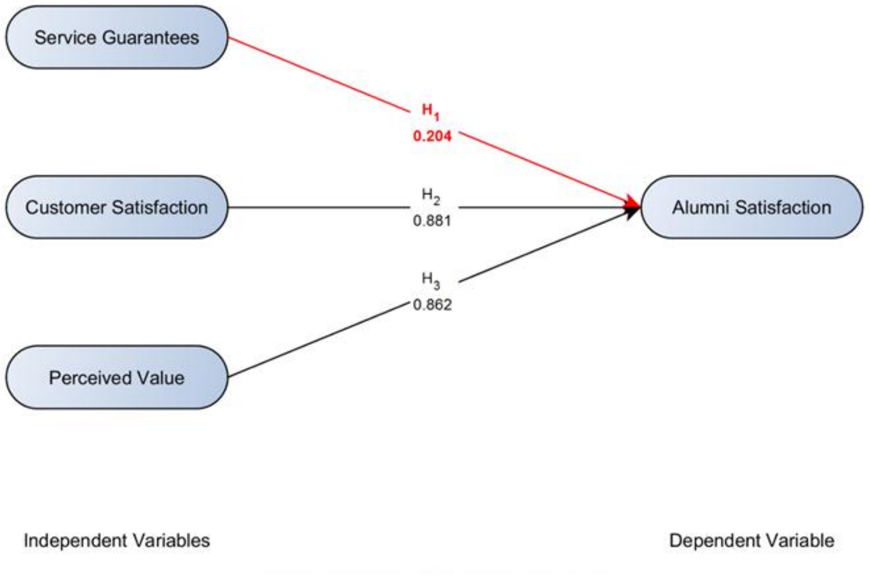


Figure 9. Pearson Product Moment Correlations Factors with Alumni Satisfaction (n = 100).

6.1 Comparison: Graduates and Post-graduates

The study went further to establish whether the two sub-groups (Graduates and Post-graduates) that are defined by different levels of education, have different perceptions of the factors that were measured. If the two educational groups assigned significantly different values to *Perceived Value* for example, it would practically mean that one group perceived the service received to be of higher quality than the perception of the other group. The significance difference between the two sub-groups was tested by performing a Cohen's d calculation. In order to conclude that there is a significant difference between sub-groups there must be both a statistical and practical significant difference. The differences found are highlighted in **Error! Reference source not found.**6.

Table 6. t-Tests: Factors by Postgraduate qualification – Graduate degree (n=36) vs Postgraduate (n=64)

Variable	Postgraduate qualification	n	Mean	S.D	Difference	t	p(d.f.=98)	Cohen's d
Perceived Value	No	36	3.99	0.98	-0.29	-1.74	.085	0.35
	Yes	64	4.29	0.69				Small
Alumni	No	36	3.86	0.74	-0.22	-1.62	.109	0.32
Satisfaction	Yes	64	4.07	0.57				Small
Customer	No	36	4.00	1.10	-0.45	-2.50	.014	0.52
Satisfaction	Yes	64	4.45	0.70				Medium
Service/Teaching	No	36	3.18	0.49	0.09	0.83	.410	0.18
Guarantee	Yes	64	3.09	0.55				Not

From the data in Table 6 it can be seen that there is a significant difference between the perceptions of *Customer Satisfaction* by graduate and post-graduate Alumni. The post-graduate Alumni rated the factors that contribute to Alumni satisfaction higher than the graduate Alumni. A possible explanation is that the post-graduate Alumni have been exposed to a wider range of services in the department due to their extended studies. Based on the services and teaching they have received elsewhere and the quality thereof, their perception of the quality of service received at the Department of Computer Sciences has improved. Figure 13 shows the only factor that had a significant difference between the means, namely Customer Satisfaction. Customer Satisfaction has a medium significant difference, Cohen's d of 0.52 (Table 6).

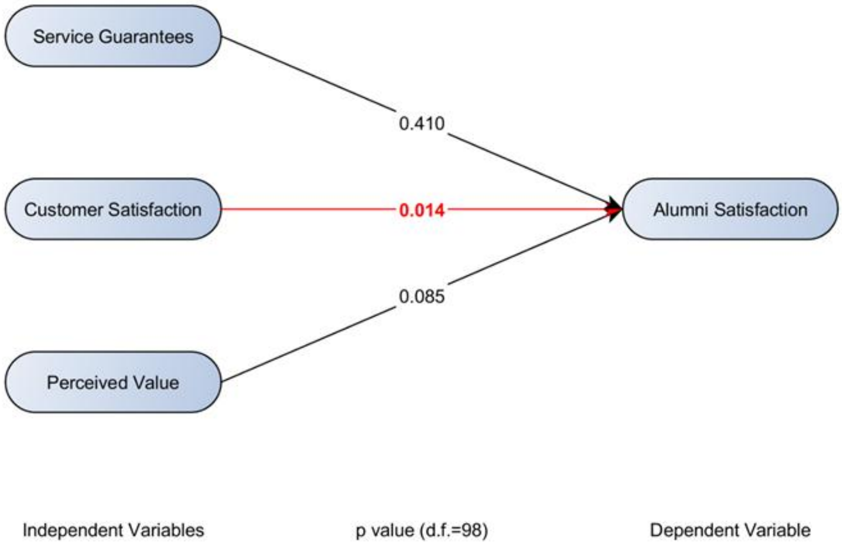


Figure 13. t-Tests: Factors by Graduate degree (n=36)
vs Post-graduate (n=64) on Alumni Satisfaction

7 Conclusions

The Alumni perception of the extent of learning and the usefulness of the knowledge is a key measure for universities to judge the quality of the service and product on offer [22, 29, 38]. McCollough & Gremler [38] defines a service guarantee as a formal promise made to customers about the service they will receive. In a service offering, in order to ensure customer satisfaction, the quality of service performance needs to be guaranteed [36, 53]. Many service industries promise a level of service performance, with or without conditions attached.

The Stakeholder theory suggests that the purpose of a business is to create as much value as possible for stakeholders and not only for the shareholders. Organisations gain a competitive advantage by involving the stakeholders as strategic resources in corporate decisions and this practice is even considered as an ethical requirement [42]. Concepts of service guarantees, consumer satisfaction and how customers perceive these guarantees were introduced in this paper, specifically relating service guarantees in education. The market signalling theory, investment theory and being focused on the customer were discussed. The benefits of including service guarantees in education and treating students as customers were highlighted. The concepts of stakeholder theory were introduced. The benefits of treating students as stakeholders and using feedback from Alumni were identified from the literature. The three factors that might influence the perceptions of Alumni of an educational institution were discussed.

The concepts of customer value and perceived value were then introduced. The section continued by identifying perceived value as well as how Alumni create value for the university. The benefits of delivering value to the customer and of increasing the perceived value of the service through the Alumni were highlighted. The perceived value that the customer gains when purchasing a product or receiving a service was emphasised. The importance to a firm, to the extent to which it delivers value to the customers and how the achievements of Alumni directly reflect on the university were shown.

The main research objective of this study was to measure the perception and satisfaction of Alumni of the Department of Computing Sciences and identify areas for improvement. It was shown that a relationship exists between the Dependent variable *Alumni Satisfaction* and the independent variables, *Customer Satisfaction and Perceived Value* by using Pearson's product-moment correlation coefficient to measure the linear association between the variables. It was further shown that there is no relationship between the Dependent variable *Alumni Satisfaction* and the independent variable *Service/Teaching Guarantees.* The concept of *Service/Teaching guarantees* is not a concept that academia are familiar with and needs to be introduced at an earlier stage.

An important finding in this study was the strong positive relationship between *Customer Satisfaction* and *Alumni Satisfaction.* A student, as a satisfied customer, becomes a satisfied alumnus. The Department of Computing Sciences should assign sufficient resources and prioritise this factor, as any perceived change would have a significant effect on the Alumni Satisfaction.

The *Perceived Value* on offer at the Department of Computing Sciences had a strong positive relationship to *Alumni Satisfaction.* This finding supported the reviewed literature where the achievements of Alumni directly reflected on the university and any improvements to the quality of the education at the university automatically improves the perceived value of the graduate's degree [15]. The recommendation was that the Department of Computing Sciences should maintain this perceived value which is a high quality offering at a reasonable price.

Service Guarantees only had a low positive correlation and any change in this factor was unlikely to influence *Alumni Satisfaction.* The low positive correlation was in contrast with the knowledge gained from literature study. As this concept in education is not well known to the Alumni, many questions were asked during the survey about this concept. The respondents felt that there was no real value added to *Alumni Satisfaction.* From the literature, *Service Guarantees* offered by a university could still generate feedback in cases of dissatisfaction and failure [37, 50]. The Department of Computer Sciences could still learn from students who invoke guarantees and adapt the programme to service future students better.

Limitations of this research study include the limited number of respondents. More detailed statistical information could have been extracted if the study was accessible for a longer period and responses from Alumni that graduated more recently would be more accurate than those from decades ago with a better representation of the current product and service offering from NMU.

Future research that is required is to improve the Alumni Satisfaction model, so that it can be applied to other departments at other universities in South Africa. Further in-depth research can be conducted to understand why there are differences between the perception of graduate degree and post-graduate Alumni and how to address their various needs.

The research conducted in this study contributed to the existing body of knowledge on the subject of Alumni satisfaction of the NMU Department of Computing Sciences. A method to measure the Alumni Satisfaction of a department was developed. Marketing at the Department of Computing Sciences can make use of the Alumni Satisfaction model in order to recognise factors that are important for Alumni and students, as well as factors that are not important or need to be adapted. Factors of the Alumni Satisfaction model which have to be improved on in order to gain a competitive advantage over other universities have been identified.

References

1. Atkinson, J. K., Coleman, P. D., & Blankenship, R. J. (2014). Alumni Attitudes on Technology Offered in Their Undergraduate Degree Program. *Academy of Business Research Journal*, *1*, 53–65.
2. Björlin Lidén, S., & Edvardsson, B. (2003). Customer expectations on service guarantees. *Managing Service Quality: An International Journal*, *13*(5), 338–348.
3. Buys, S. (2014). *Market Needs Analysis For A Turnkey Automation Project Based Organisation In The Eastern Cape*. Nelson Mandela Metropolitan University.
4. Caballero, G., Vázquez, X. H., & Quintás, M. A. (2015). Improving Employability Through Stakeholders in European Higher Education: The Case of Spain. *Long Range Planning*, *48*(6), 398–411.
5. Calitz, A. P., Greyling, J. H., & Cullen, M. D. M. (2014). South African Industry ICT Graduate Skills Requirements. *Southern African Computer Lecturers' Association (SACLA)*, (1), 25–26.
6. Calitz, A. P., Greyling, J. H., & Glaum, A. P. M. (2016). CS and IS Alumni Post-Graduate Course and Supervision Perceptions. In *Southern African Computer Lecturers' Association (SACLA)*, 115–122.
7. Chi, H., Jones, E. L., & Grandham, L. P. (2012). Enhancing Mentoring Between Alumni and Students via Smart Alumni System. *Procedia Computer Science*, *9*, 1390–1399.
8. Clarkson, M. E. (1995). A stakeholder framework for analysing and evaluating corporate social performance. *Academy of Management Review*, *20*(1), 92–117.
9. Coelho, P. S., & Henseler, J. (2012). Creating customer loyalty through service customization. *European Journal of Marketing*, *46*(3/4), 331–356.
10. Collis, J., & Hussey, R. (2009). Business research: a practical guide for undergraduate and postgraduate students. *Palgrave Macmillan UK*, *3rd*, 420.
11. Connelly, B. L., Certo, S. T., Ireland, R. D., & Reutzel, C. R. (2011). Signaling Theory: A Review and Assessment. *Journal of Management*, *37*(1), 39–67.
12. Díaz-Méndez, M., & Gummesson, E. (2012). Value co-creation and university teaching quality: Consequences for the European Higher Education Area (EHEA). *Journal of Service Management*, *23*, 571–592.
13. Dovaliene, A., Masiulyte, A., & Piligrimiene, Z. (2015). The Relations between Customer Engagement, Perceived Value and Satisfaction: The Case of Mobile Applications. *Procedia - Social and Behavioral Sciences*, *213*, 659–664.
14. Durango-Cohen, P. L., Durango-Cohen, E. J., & Torres, R. L. (2013). A Bernoulli–Gaussian mixture model of donation likelihood and monetary value: An application to alumni segmentation in a university setting. *Computers & Industrial Engineering*, *66*(4), 1085–1095.
15. Egizii, R. (2015). Self-directed Learning, Andragogy and the Role of Alumni as Members of Professional Learning Communities in the Post-secondary Environment. *Procedia - Social and Behavioral Sciences*, *174*, 1740–1749.
16. Erevelles, S. (1993). The price-warranty contract and product attitudes. *Journal of Business Research*, *27*(2), 171–181.
17. Fabien, L. (2005). Design and implementation of a service guarantee. *Journal of Services Marketing*, *19*(1), 33–38.
18. Flegg, C. (2012). Alumni, Libraries and Universities: whereto the relationship. *The Journal of Academic Librarianship*, *38*(1), 60–62.
19. Fox, W., & Bayat, M. S. (2010). *A Guide to Managing Research*. (E. Clarke, Ed.) (1st Ed.). Cape Town: Juta and Company Ltd.
20. Fraering, M., & Minor, M. S. (2013). Beyond loyalty: customer satisfaction, loyalty, and fortitude. *Journal of Services Marketing*, *27*(4), 334–344.
21. Freeman, R. E. (2004). The Stakeholder Approach Revisited. *Zeitschrift Fuer Wirtschafts- Und Unternehmensethik*, *5*(3), 228–241.
22. Guevara, C., & Stewart, S. (2011). Do student evaluations match alumni expectations? *Managerial Finance*, *37*(7), 610–623.
23. Hennig-Thurau, T., Langer, M. F., & Hansen, U. (2001). Modeling and Managing Student Loyalty: An Approach Based on the Concept of Relationship Quality. *Journal of Service Research*, *3*(4), 331–344.
24. Hillman, A. J., & Keim, G. D. (2001). Shareholder value, stakeholder management, and social issues: What's the bottom line? *Strategic Management Journal*, *22*(2), 125–139.
25. Hogreve, J., & Gremler, D. D. (2009). Twenty Years of Service Guarantee Research: A Synthesis. *Journal of Service Research*, *11*(4), 322–343.
26. Holttinen, H. (2014). Contextualizing value propositions: Examining how consumers experience value propositions in their practices. *Australasian Marketing Journal (AMJ)*, *22*(2), 103–110.
27. Ihantola, E.-M., & Kihn, L.-A. (2011). Threats to validity and reliability in mixed methods accounting research. *Qualitative Research in Accounting & Management*, *8*(1), 39–58.
28. Kashyap, R. (2001). The Effects of Service Guarantees on External and Internal Markets. *Academy of Marketing Science Review*, *2001*(10).
29. Khatimin, N., Wahab, D. A., & Mohamed, A. (2011). Postgraduate Alumni Survey of the Faculty of Engineering and Built Environment. *Procedia - Social and Behavioral Sciences*, *18*, 110–117.
30. Kumar, R. (2011). *Research Methology: A Step-by-Step Guide for Beginners*. *Igarss 2014*.
31. Landroguez, S. M., Castro, C. B., & Cepeda-Carrión, G. (2011). Creating dynamic capabilities to increase customer value. *Management Decision*, *49*(7), 1141–1159.
32. Leedy, P. D., & Omrod, J. E. (2010). *Practical Research: Planning and Design* (9th Ed). New Jersey: Pearson.
33. Magnusson, K. (2014). Interpreting Cohen's d effect size: an interactive visualization.
34. Maree, K., Creswell, J. W., Ebersohn, L., Eloff, I., Ferreira, R., Ivankova, N. V., … van der Westhuizem, C. (2012).

First Steps in Research. (K. Maree, Ed.) (1st Ed). Pretoria: Van Schaik.

35. Martelo Landroguez, S., Barroso Castro, C., & Cepeda-Carrión, G. (2013). Developing an integrated vision of customer value. *Journal of Services Marketing*, *27*(3), 234–244.
36. McColl, R., & Mattsson, J. (2011). Common mistakes in designing and implementing service guarantees. *Journal of Services Marketing*, *25*(6), 451–461.
37. McCollough, M. A., & Gremler, D. D. (1999). Guaranteeing Student Satisfaction: An Exercise in Treating Students as Customers. *Journal of Marketing Education*, *21*(2), 118–130.
38. McDearmon, J. (2013). Hail to Thee, Our Alma Mater: Alumni Role Identity and the Relationship to Institutional Support Behaviors. *Research in Higher Education*, *54*(3), 283–302.
39. Mcgourty, J., Besterjeld-Sacre, M., Shuman, L. J., & Wolfe, H. (1999). Improving Academic Programs by Capitalizing On Alumni’s Perceptions and Experiences. *Frontiers in Education Conference*, 13A5/9-13A515.
40. Nunnally, J. C. (1978). *Psychometric Theory*. New York: McGraw Hill.
41. Parmar, B. L., Freeman, R. E., Harrison, J. S., Wicks, A. C., Purnell, L., & de Colle, S. (2010). Stakeholder Theory: The State of the Art. *The Academy of Management Annals*, *4*(1), 403–445.
42. Plaza-úbeda, J. A., Burgos-Jiménez, J., & Carmona-Moreno, E. (2014). Stakeholder Integration: Measuring and Knowledge, Interaction Behavior Dimensions Adaptational. *Journal of Business Ethics*, *93*(3), 419–442.
43. Plewa, C., & Quester, P. G. (2008). A dyadic study of “champions” in university-industry relationships. *Asia Pacific Journal of Marketing and Logistics*, *20*(2), 211–226.
44. PWC. (2015). Funding of public higher education institutions in South Africa. Retrieved from http://www.pwc.co.za/en/higher-education/Funding-public-higher-education-institutions-SA.html.
45. Rattanamethawong, V., Sinthupinyo, S., & Chandrachai, E. A. (2015). An Innovation System that can Quickly Responses to the Needs of Students and Alumni. *Procedia - Social and Behavioral Sciences*, *182*, 645–652.
46. Rodrigues, K. F. de C., Nappi, V., & Rozenfeld, H. (2014). A Proposal to Support the Value Proposition in Product Oriented Service Business Model of Product Service Systems. *Procedia CIRP*, *16*, 211–216.
47. Schlesinger, W., Cervera, A., & Pérez-Cabañero, C. (2016). Sticking with your university: the importance of satisfaction, trust, image, and shared values. *Studies in Higher Education*, (February), 1–17.
48. Schofield, P., & Fallon, P. (2012). Assessing the viability of university alumni as a repeat visitor market. *Tourism Management*, *33*(6), 1373–1384.
49. Sheehan, N. T., & Bruni-Bossio, V. (2015). Strategic value curve analysis: Diagnosing and improving customer value propositions. *Business Horizons*, *58*(3), 317–324.
50. Song, H., Cadeaux, J., & Yu, K. (2015). The effects of service supply on perceived value proposition under different levels of customer involvement. *Industrial Marketing Management*.
51. Sum, C. C., Lee, Y. S., Hays, J. M., & Hill, A. V. (2002). Modeling the effects of a service guarantee on perceived service quality using alternating conditional expectations (ACE). *Decision Sciences*, *33*(3), 347–383.
52. Tavakol, M., & Dennick, R. (2011). Making sense of Cronbach’s alpha. *International Journal of Medical Education*, *2*, 53–55.
53. Tucci, L. A., & Talaga, J. (1997). Service guarantees and consumers’ evaluation of services. *Journal of Services Marketing*, *11*(1), 10–18.
54. Walsh, J. P. (2005). Book Review Essay Taking Stock of Stakeholder Management. *Academy of Management Review*, *30*(2), 426–439.
55. Watt, T. T., Norton, C. L., & Jones, C. (2013). Designing a campus support program for foster care alumni: Preliminary evidence for a strengths framework. *Children and Youth Services Review*, *35*(9), 1408–1417.
56. Woodruff, R. B. (1997). Customer value: The next source for competitive advantage. *Journal of the Academy of Marketing Science*, *25*(2), 139–153.
57. Wouters, M., & Kirchberger, M. A. (2015). Customer value propositions as interorganizational management accounting to support customer collaboration. *Industrial Marketing Management*, *46*, 54–67.

Perceptions of Postgraduate Students on the Writing of Reflective Journals as a Means for Personal, Professional and Research Development

Liezel Cilliers and Roxanne Piderit

University of Fort Hare, East London, South Africa
lcilliers@ufh.ac.za; rpiderit@ufh.ac.za

Abstract. Increasingly it has been found that students entering postgraduate studies are not able to easily adapt to the requirements of the research project they need to undertake as part of the programme. In order to address this problem, reflective journals can be used to manage the postgraduate student's personal, professional and research development during the early stages of their research project. Reflective journaling helps to develop the critical thinking and problem solving skills of the student which is especially important for new researchers. The aim of the study is to investigate the postgraduate students' perception of reflective journal writing as a means for personal, professional and research development. A quantitative and descriptive survey was conducted in February 2017. Nineteen participants were recruited from an Information Systems honours class and asked to complete a structured questionnaire. In addition, relevant passages from the reflective journals which indicated development in one of the three areas are provided to illustrate the student development. The students indicated that they generally experienced writing of reflective journals to be a valuable tool in enhancing personal development, professional growth and their research skills. The barriers that must be considered by supervisors when they make use of a reflective journaling activity with their students include the time spent and ability to write in English. The recommendation of the study is that supervisors incorporate reflective journaling for their new postgraduate students to aid in the development of higher-level skills such as critical thinking.

Keywords: Journalling, Postgraduate Research, Reflection.

1 Introduction

Higher education institutions have the responsibility to develop students' ability to critically assess and examine the theories and concepts related to their academic discipline [1]. The possession of critical thinking skills is important when students enter postgraduate studies, as they are expected to reflect on and understand the research process as it relates to their own beliefs, values and knowledge base [2]. Research problems typically present as complex real-life problems which can only be evaluated making use of higher-level thought processes. Although reflection is a critical skill for postgraduate students, they often struggle with this concept as they have not received enough exposure to the activity during their undergraduate courses. Lecturers make use of tests and examinations to test theoretical knowledge at undergraduate level, but are unable to monitor this higher level development of the students due to large class sizes [3,4]. Thus, the existence of these skills is first evaluated when students enter postgraduate studies and have the sole attention of a supervisor.

Reflection is defined as 'active, persistent and careful consideration of any belief or supposed form of knowledge in the light of the grounds that support it and the further conclusion to which it tends' [5:9]. When students are encouraged to reflect on their learning experiences, it provides the student with the opportunity to actively engage in the learning process and acquire a deeper understanding of what they have learned [6,7].

Reflective journaling is a popular technique used to teach and guide students to develop higher order thinking skills such as reflection, critical thinking and problem solving [8]. The student can write in their journal at any time which provides flexibility while the journal entries provide a platform for students to voice their opinions, feelings and issues [9]. Journaling can be done with guided questions from the lecturer or as a free-writing activity [10]. Reflective journals allow the students to compare and contrast their experiences with their existing

SACLA'17, July 3–5, 2017, Magaliesburg, South Africa.

knowledge, values and beliefs which promote higher order thinking. The skills that the students learn when journaling will help them to identify and analyse their own strengths and weaknesses, problem solve and pursue questions on their own which are all valuable tools that can be applied in their research projects and future professional careers [11].

This paper outlines the perceptions of students on the use of reflective journaling for their own personal, professional and research development, and specifically with regards to the skills required to undertake a research project. The key lessons learned through the student feedback are discussed to provide insight into the value of such an exercise in developing high-level skills in postgraduate students.

The research problem which underpins this paper is briefly described and thereafter the relevant literature review is outlined. The methodology, study setting and data analysis follow. The discussion of the findings provides insight into the value of a journaling exercise in the personal, professional and research development of a postgraduate student. Factors which act as barriers to the effective use of journaling are also described and discussed.

2 Problem Statement

One of the main characteristics of a good researcher is the ability to critically reflect on the literature that they are reading [20]. Postgraduate students often struggle to develop this skill as they have not been exposed to academic writing during their undergraduate degree. New and innovative strategies, such as reflective journaling, can be utilised by supervisors to help the student develop on a personal, professional and research level [24]. The aim of this research article is to investigate the postgraduate students' perception of reflective journal writing as a means for personal, professional and research development. The article will contribute to the body of knowledge in the scholarship of teaching and learning, and provide valuable insight for supervisors as to how to use reflective journaling to help their postgraduate students to develop critical thinking skills, and similar high-level skills necessary for research.

3 Literature

Journaling has become popular during the past two decades in higher education and has been used in a variety of academic fields, ranging from nursing, science, business studies and social sciences [24]. Journaling can be used to stimulate the cognitive, affective and psychomotor domains. The cognitive domain, also called the knowledge domain, includes the student's ability to recall, think and understand a concept. The affective domain represents the student's ability to express values or attitudes. This domain also includes leadership, social responsibility, culturally competent research, research advocacy and autonomy, ethical considerations and personal values. The last domain, the psychomotor domain, enables the student to reflect on their research skill and how they will improve it in future [12]. According to these three domains the purpose of journaling is threefold [13]:

- It helps the students to connect and apply theory learned in the classroom to real-life problems;
- It helps students to acknowledge their own strengths and weaknesses; and
- It helps students to plan on how to improve in the future.

A model that is often used to discuss student self-reflection is the DEAL Model for Critical Reflection. 'DEAL' stands for Describe, Examine, and Articulate Learning. The student must first describe the learning or research experience, examine the issues or goals that they have found in the research problem and then decide how to improve the research problem in the future [14]. The reason why reflective journaling has been encouraged in post graduate studies is that it provides the student with the opportunity to explore and describe a research problem internally first, by making use of their prior experience, which helps them to solve and clarify the research problem by using a scaffolding approach [4]. Postgraduate students are then able to achieve a deeper level of understanding of content and context for research purposes [15, 16].

The advantages of reflective journaling have been reported in literature as contributing to the professional, cognitive and affective development of the student; increased and active awareness of research issues; and being able to connect with others in the community [9, 17]. However, literature also reported that there are barriers to the use of reflective journaling such as low level of critical thinking exhibited by students that 'write for the supervisor', poorly structured assignments, lack of time management for journal exercises, negative perceptions of reflective assignments, issues of trust or ethics and assigning marks to journal entries [18,19].

With this in mind, the authors set out a set of journaling exercises for 22 new postgraduate students. This was used to not only identify issues related to curriculum, teaching or assessment, but to provide the students with the opportunity to reflect and develop critical thinking skills.

4 Methodology

The primary data collected for this study was obtained through a quantitative survey method used to elicit participants' perceptions about reflective journaling. Therefore, a descriptive research design was deemed suitable to explain and provide a detailed analysis of the state of the research problem which was clearly defined and made use of formal measures to collect primary data [20].

A structured questionnaire, which was previously developed by Mahlanze and Sibiya [4], was used as the data collection tool. The data collection tool was structured according to the DEAL model described in Section 3, which provides for the personal growth, professional development and academic (research) enhancement of the student [14]. The questionnaire consisted of a demographic section (4 questions), and sections that investigated the personal growth (5 questions), professional development (6 questions) and research development (6 questions) of students. Lastly a section was included that referred to the barriers that prevent successful reflective writing among postgraduate students (5 questions). The sections, with the exception of the demographic section, all made use of a Likert type rating scale with 4 options (Strongly Agree, Agree, Disagree, and Strongly Disagree).

In addition, the comments provided in the process of the journaling exercises are included to provide further detail to certain aspects of the findings. The journal entries were required after 5 milestones within the first two weeks of the research project, namely: research project introduction; library training; formatting, writing and presentation workshop; submission of the initial essay; and after their first meeting with their supervisor. The journaling process is described in detail in Section 5.

4.1 Study Setting

The study was conducted at a traditional university in the Eastern Cape, namely the University of Fort Hare. The study population consisted of the Information Systems honours class starting their degree in 2017. These students have no prior research experience as the research module is only offered for the first time during their Honours year. For this reason, the use of journaling to introduce or orientate them in Information Systems research was preferred. While all the students participated in the reflective journaling exercise, only 19 of the 22 students completed the questionnaire, meaning a response rate of 86% was achieved.

Reflective journaling was used to encourage postgraduate students entering the Honours degree programme to document their thoughts and experiences about the research process during the initial weeks of their research project. The research project at Honours level is a 32 credit subject which spans 10 months of the year. Each student is assigned to a supervisor who facilitates their research project. The outcomes of the module include presentations to the class and staff regarding the progress of the project, a treatise, an article and research poster.

From prior experience, the staff involved in the initial stages of the subject recognised the difficulty the students experienced when choosing a topic, finding a research problem and writing the proposal. Reflective journaling was identified as a possible method to encourage students to improve their critical thinking skills while detailing the difficulties they experience and allowing them to reflect on the research process.

The staff members explained the purpose of the reflective journaling exercise to the students at the start of their orientation week, which was aimed specifically at orientating the students with the research requirements for their Honours year. As mentioned previously, journal entries were required after 5 milestones within the first two weeks of the research project, namely:

- *Research Project Introduction*: During the first week of orientation, the students were given a list of possible research topics and asked to choose two of these topics which could potentially become their research project. The students were also given a list of the required outcomes of the research project.
- *Library Training*: The library training was specifically tailored to introduce the students to the skills and resources they would need to complete their research project. This included database searching and referencing management tools.
- *Formatting, Writing and Presentation Workshop*: This workshop is designed to ensure all students are able to meet the technical formatting and presentation requirements of the research project.
- *Submission of the Initial Essay*: The students were required to identify and write a 2 page academic essay on the problem within the research topic area that they identified from the possible research topics. These essays

were used to assign students to supervisors who have an interest in that area and to evaluate the critical thinking and writing abilities of the students.

- *First Meeting with their Supervisor*: The first meeting between the student and supervisor is the first opportunity to establish a working relationship between the two parties. This is also the first opportunity for the students to get feedback and input directly related to their research project.

In order to guide the students through the reflective process, for each journaling opportunity, students were required to reflect on what they had learnt, what they did not understand, what the academics could do differently to aid understanding, how they felt about the research project at that point, and what step will they take next to work on the project. In addition, specific questions were asked related to the specific milestone.

After the two week period, a questionnaire was distributed amongst the students. The students were informed that the questionnaire would be confidential, that their participation was voluntary and that they could withdraw from the study at any time without prejudice. A pilot study was conducted to test the survey questionnaire. Two staff members were asked to evaluate the questionnaire in terms of ambiguity and user friendliness. Both staff members expressed satisfaction with the questionnaire. The findings from this study are presented and discussed in the next section.

4.2 Data Analysis

The survey questionnaire was analysed making use of descriptive statistics (mean, standard deviation and percentages) which is represented in the tables in the sections that follow. The Statistical Package for Social Sciences (SPSS 24) was used to analyse the statistics. The Cronbach alpha coefficient (α) was used to test for internal consistency of the measuring instrument [21]. A high Cronbach alpha coefficient suggests that the scale used is reliable. Values of 0.70 and above represent a good level of reliability, whereas values between 0.50 and 0.69 are considered to indicate an acceptable level of reliability [21]. The Cronbach alphas were calculated for the various sections of the questionnaire and found to be acceptable in all four cases. Table 1 provides the results of the empirical reliability testing for the scales used in this research.

Table 1. Cronbach alpha coefficients for the scales comprising the measuring instrument

Section	Cronbach Alpha
Professional development	0.694
Personal development	0.597
Research development	0.861
Barriers to reflective learning	0.767

5 Research Findings: Presentation, Analysis and Discussion

A total of 19 honours students participated in the study. The students indicated that 9 (47,4%) are younger than 25 years of age, 6 students (31.6%) are between the ages of 26 and 35 years and 3 students (15.8%) are older than 35 years of age. One student declined to provide their age.

The majority of the class are new fulltime students (14 students or 73,7%) while 5 students (26.3%) were returning part time students. The class consisted of more females (63.2%) than males (36.8%).

The students were also asked if English was their first language. Only 6 students (31.6%) indicated that English was their first language while 13 students (68.4%) considered English to be their second or third language. This demographic is representative of the student population of the traditional university where the first language of the majority of the students is isiXhosa.

5.1 Overall Descriptive Statistics

The questionnaire made use of a Likert type scale consisting of 4 options (Strongly Agree, Agree, Disagree, Strongly Disagree). A score of 2 or below indicates a positive impact while a score of above 2 indicates a negative impact. Personal development had an average score of 1.78, while professional development yielded an average score of 1.76. The third section, research development had an average score of 1.76 while the barriers to reflective writing recorded an average score of 2.61.

These results indicate that the first three sections were regarded as positive and beneficial in terms of personal, professional and research development. The last section, the barriers to reflective writing, was regarded as negative, which means that the barriers presented in the questionnaire were not regarded as a challenge to reflective writing by the students. The rest of the article will discuss the various findings.

5.2 Results for Personal Development

Participants were asked to consider the following variables regarding personal development when reflecting on the journaling exercise (see table 3):

- Empowerment towards examining own attitudes and perspectives to a given situation;
- Awareness of strengths and weaknesses;
- Assistance in developing coping skills towards research experiences;
- Consciousness to own feelings and emotions to deal with self and others;
- Afforded a voice for self-expression; and
- Improved confidence.

A significant number of participants (18 or 94.7%) agreed that reflective journaling empowered them to examine their own attitudes and perspectives while deciding on a research topic, while all 19 students (100%) agreed that the exercise helped them to become aware of their strengths and weaknesses during the research process. Seventeen students (89.5%) stated that they were able to develop coping skills that will help them in their research project and that they were more conscious of their own feelings and emotions which subsequently improved their confidence levels. An additional benefit of personal development was the ability of the students to express themselves better (18 or 94.8%).

Table 2. Test statistics for personal development

Variables	N	Mean	% Agree
Empowerment towards examining own attitudes and perspectives to a given situation	19	1.89	94.7
Awareness of own strengths and weaknesses	19	1.68	100.0
Assistance in developing coping skills towards research experiences	19	1.78	89.5
Consciousness to own feelings and emotions to deal with self and others	19	1.89	89.5
Afforded a voice for self-expression	19	1.74	94.8
Improved confidence	19	1.79	89.5

The postgraduate student is the writer of the journal and must take ownership of their work [10]. While the supervisors in this study provided guided questions, it remains the student's responsibility to use these questions to stimulate critical thinking about their research project.

A journal entry provided in quotation marks below, indicates that at this stage of the research project students are recognising the enormity of the task, yet still optimistic of a successful completion. The reflection on the value of the Honours programme as a whole at this point indicates an early appreciation for the personal development opportunities of the programme.

> "It is exciting but I am still intimidated and overwhelmed. I have realised that I made the correct decision in pursuing this programme as even at this stage it inculcates a sense of self-preservation, self-worth and provides a culture of learning that is different from the norm."

If the student further reflects on their own growth during this project, the journal entries will have substantially more depth and benefit to the student. The development of personal skills has the ability to improve the self-awareness and emotional maturity of the postgraduate student.

5.3 Results for Professional Development

This section consisted of 5 questions and focused on aspects of professional development. The students answered the following variables when answering the questionnaire (see table 2):

- Ability to correlate theory with practice;
- Ability to improve in problem solving skills;
- Ability to redefine experiences and problems;
- Ability to make proactive decisions and take corrective actions; and
- Ability to use past experiences to prevent future occurrences.

A significant number of students (18 or 94.7%) agreed that reflective writing helped them to correlate the theory they had learned in class with practice, while the same amount of students (18 or 94.7%) thought that reflective writing improved their problem solving skills. All 19 of the students (100%) stated that the writing of reflective journals impacted positively on their ability to redefine experiences and research problems while 18 students (94.7%) thought that the activity would help them to make decisions proactively in future. Only 2 students (10.5%) felt that journaling will not help them to learn from past experiences.

Table 3. Test statistics for professional development

Variables	N	Mean	% Agree
I am able to correlate theory with practice	19	1.84	94.7
My problem solving skills have improved	19	1.84	94.7
I am able to redefine experiences and problems	19	1.84	100.0
I am able to make proactive decisions and take on the spot corrective actions	19	1.74	94.7
I am able to use past experiences to prevent future occurrences	19	1.63	89.5

Literature suggests that professional development of the student can be greatly enhanced by reflective journaling [22,23]. Professional development includes skills such as problem solving, making proactive decisions and using past experiences to prevent future occurrences. These are all important skills for the postgraduate student to learn as each of the students enter the research process with their own beliefs and values.

A journal entry provided in quotation marks below, indicates that this student has recognised the required steps for him/her to meet the requirements for the research project. Thus, the reflective journaling can give us an insight into the state of the professional development of the students.

> "I learn a lot about research, how to write it, how to research and what is expected from me so that I will have a well written paper…. I have to find articles and journal that have information that are related to my research topic. I will read all articles so that I will know what to write about."

Reflecting through journaling was found to increase over time, and several studies suggest that higher levels of reflection are associated with better performance in examinations [24]. It is therefore encouraging that the students in this study are of the opinion that reflective journaling is a worthwhile activity that can benefit students in their professional development in academia.

5.4 Results for Research Skill Development

Participants were asked to consider the following variables regarding research skill development when reflecting on the journaling exercise (see table 4):

- Enhanced writing and recording skills;
- Promoted understanding of learning outcomes;
- Increased active involvement and ownership of learning;
- Increased ability in reflection and thinking;
- Stimulated to search for more knowledge to prepare for future research experiences; and
- Helped to improve observation skills.

The participants were very positive about the use of reflective journaling to promote the development of research skills. All the participants (19 or 100.0%) stated that journaling would promote the understanding of learning outcomes in the research process and would increase the ability to reflect and think about their research topics. A significant number of participants (18 or 94.7%) agreed that reflective journaling enhanced their writing and observations skills, the ownership of the research project and how to search for more knowledge to prepare for future research experiences.

Table 4. Test statistics for research development

Variables	N	Mean	% Agree
Enhanced writing and recording skills	19	1.89	94.7
Promoted understanding of learning outcomes.	19	1.79	100.0
Increased active involvement and ownership of learning.	19	1.84	94.7
Increased ability in reflection and thinking.	19	1.68	100.0
Stimulated to search for more knowledge to prepare for future research experiences.	19	1.68	94.7
Helped to improve observation skills.	19	1.79	94.7

The students perceived the reflective journaling exercise to be beneficial and of value to increase their research skills. This is in line with previous studies that suggested that the ability to apply theory to practical experiences motivates postgraduate students to make use of higher levels of reflections [22]. This is important during the research process as postgraduate students can apply the knowledge that they have learned during their undergraduate degree.

A journal entry provided in quotation marks below, indicates that this student is actively taking responsibility for his/her studies and has decided on a path forward. Thus, the initial development of research skills has begun.

> “I am still struggling with conceptualization of my research problem especially relating to the South African context, but I know that I need to read lots of articles based on the South African context so that I can write a cohesive essay based on valid points.”

Students also value journaling as a learning activity to reach specific learning outcomes [13]. These results are important as it suggests that supervisors can make use of reflective journaling to increase the reflection and critical thinking skills of their students.

5.5 Barriers to Effective Journaling

Participants responded to the following variables in the questionnaire (see table 5):

- My command of language limited my writing ability;
- Writing the reflective journal took too much time;
- The benefits of journaling are not clear to me;
- I feel uncomfortable writing my personal feelings in the reflective journal; and
- I may be victimized when I express my real feelings.

Table 5. Test statistics for barriers associated with reflective writing

	N	%	N	%	N	%	N	%
	Strongly agree		Agree		Disagree		Strongly Disagree	
My command of language limited my writing ability.	1	5.3	10	52.6	5	26.3	3	15.8
Writing the reflective journal took too much time.	3	15.8	8	42.1	6	31.6	2	10.5
The benefits of journaling are not clear to me.	1	5.3	7	36.8	8	42.1	3	15.8
I feel uncomfortable writing my personal feelings in the reflective journal.	2	10.5	6	31.6	9	47.4	2	10.5
I may be victimized when I express my real feelings.	0.0	0.0	7	36.8	7	36.8	5	26.3

The majority of the students indicated that these barriers are not important when they participated in the critical reflection exercise. Most of the students that participated in this study indicated that English is not their first language, therefore it was expected that students would feel that this would be a limiting factor (11 or 57.9% of students). While the official language of the traditional university is English and the student completed their undergraduate degrees in English, they are still insecure about using English for academic purposes. In this regard journaling can also help to develop the student's writing skills as they learn to transfer thoughts into written information.

Students prefer activities that take less time as can be seen by the results of this study [25]. The students indicated that the reflective journaling exercises took too much of their time (11 or 57.9% of students). Miller [10] reports that students tend to view journaling activities as cumbersome as they see it as ‘busy work’. Some students even fabricate entries at the last minute in order to have something to hand in. previous studies also suggest that student have reported that journaling during the semester does not feel purposeful or meaningful [13].

Although the purpose of the reflective journaling exercise was explained to the students at the start of the orientation week, 8 students (42.1%) still indicated that they did not understand the value of the activity. This can be due to the inexperience of the postgraduate students with this activity as the researchers are not aware of any other course in the university that requires students to keep a journal.

Miller [10] reports that it is important to explain the purpose of journaling to the students in order to understand what the role of the supervisor and student are in the exercise. The supervisor has to explain their expectations with regard to the content, level of reflection and length of the journal entry. If appropriate the supervisor also provides guided questions, as was used in this study, and has to review the student’s completed journal. The purpose of this review is to help students to merge ideas that have remained unconnected and help the student to reflect at a higher level [13].

Literature has reported that students prefer anonymity when writing or report a distrust in sharing the journal entries with supervisors [25,13]. Eleven students (57.9%) indicated that they did feel comfortable writing their feelings in the journal entries. By definition journaling provides an emotional outlet where students can write down and explore their emotions, but these feelings are very personal [13]. If the student does not trust their supervisor, they may feel vulnerable or embarrassed to write truthfully or feel it is an invasion of privacy.

In line with the previous question, a third of the class (36.8%) indicated that they fear they may be discriminated against if they write their true feelings. Literature has consistently identified issues such as confidentiality of entries and the ethics of grading personal reflections as controversial [26]. If the student feel that they cannot trust the supervisor, they will reduce the activity to a simple word count and ‘write for the supervisor’ meaning that they do not reflect on their own work, but simply writes what the supervisor wants to hear in order to obtain better marks. Recognizing what students perceive to be barriers to this learning activity can lead to arranging the activity to be more acceptable.

6 Recommendations for the use of Journaling in Information Systems Research Projects

For these Information Systems Honours students, with no prior research experience, journaling was used to provide a communication platform between lecturing staff and the students. The prior writing experience for these students consisted of essay writing assignments in undergraduate subjects, thus the level of writing required for the research project is significantly higher. The core purpose of the journaling exercises was to allow lecturing staff to gain insight into the areas students struggle with while beginning a research project, in order to provide assistance if necessary or adjust the teaching and/or mentoring activities to assist the student.

In terms of personal development, journaling has allowed the students to be reflective of their own strengths and weaknesses. This includes being honest about their own level of effort expended on the research project, instead of levelling criticism on the supervisors for their own shortcomings. Students were also able to increase their confidence in their abilities to perform the research tasks.

Professional development was achieved through the guidance of students from the theoretical to practical aspects of research, and the development of problem-solving abilities. Research skills developed included enhancement of writing and observation skills.

The barriers to participation in the journaling were interesting to note. Given the focus in higher education of language issues, it was surprising to note that the use of English as a medium for the exercise was not concerning to the students, even though they were not first language English speakers. However, a significant barrier was the length of time taken to complete the journal entries. The Honours students feel they are under significant time pressure and therefore felt this exercise did not make best use of their time. It is important that these journal entries be kept short, and that guiding questions are used in order to minimize the time effort required by students.

7 Conclusion

The success of this reflective journaling approach is evidenced by the continued development of the students participating in the Honours research project. Staff and students need to have a common understanding that they are jointly responsible for the holistic development of the student. The reflective journal can help the student to understand the research process better, while helping the supervisor to identify the problems of the student and addressing those for the individual.

Since the completion of the last reflective journal entry described here, further journal writing has been encouraged. It has been found that reflective journaling can be useful to both the supervisor and postgraduate student, but must be implemented correctly in order to be effective.

The study findings cannot necessarily be generalised as the study was conducted with one Information Systems Honours class at a single university. However, this study was based on a similar study with Nursing students and produced comparable results. For this reason, it is expected that similar results could be obtained in comparable settings.

Future research may broaden the study population to different Honours classes at the university to provide a useful comparison between degrees or even universities. In addition, the journaling exercise will continue throughout the research project year to evaluate the effectiveness and usefulness across the different phases of the research project. Further papers can include an analysis of the students' journal entries throughout the research process.

References

1. Thorpe, K. (2004). Reflective Learning Journals: From Concept to Practice. *Reflective Practice*, 5(3): 327–343.
2. Colley, BM., Bilics, RA. & Lerch, CM. (2012). Reflection: A Key Component to Thinking Critically. *Canadian Journal for the Scholarship of Teaching and Learning*, 3(1), http://dx.doi.org/10.5206/cjsotl-rcacea.2012.1.2.
3. O'Connell, TS. & Dyment, JE. (2016). 'I'm just not that comfortable with technology': student perceptions of and preferences for Web 2.0 technologies in reflective journals. *Journal of Further and Higher Education*, 40:3, 392-411, DOI: 10.1080/0309877X.2014.984594.
4. Mahlanze, HT. & Sibiya, MN. (2016). Perceptions of student nurses on the writing of reflective journals as a means for personal, professional and clinical learning development. *Health SA Gesondheid*, 22, 79-86, doi.org/10.1016/j.hsag.2016.05.005.
5. Dewey, J. (1933). *How We Think*. Boston, MA: D.C. Heath.
6. McDevitt, P. (2000). Using journals to enhance computer simulation based learning. *Developments in Business Simulation & Experiential Learning*, 27, 57-62.
7. Hermansyah, L. (2016). *Reflective learning journal: teacher guide.* Available from https://www.scribd.com/doc/299831122/Reflective-Learning-Journal-Teacher-Guide (Accessed 25/02/2017).
8. Thorsen, CA. & DeVore, S. (2013). Analysing reflection on/for action: A new approach. *Journal of Reflective Practice*, 14(1), 88-94.
9. Bulman, C. & Schultz, S. (2008). *Reflective practice in nursing (4th ed.)*. Hoboken: Blackwell Publishing Ltd.
10. Miller, LB. (2017). Review of Journaling as a Teaching and Learning Strategy. Teaching and Learning in Nursing, 12, 39-42.
11. Harris, LL., Warren, JL. & Ashleigh, M. (2010). Small Steps Across the Chasm: Ideas for Embedding a Culture of Open Education in the University Sector. *Education* 16(1). http://ineducation.ca/index.php/ineducation/article/view/47/510.
12. University of Connecticut (2016). Assessment primer: Learning taxonomies. Available from http://assessment.uconn.edu/assessment-primer/assessmentprimer-learning-taxonomies/ (26/02/2017).
13. Ruiz-López,M., Rodriguez-García,M., Villanueva, P., Márquez-Cava,M., García-Mateos, M., Ruiz-Ruiz, B. & Herrera-Sánchez, E. (2015). The use of reflective journaling as a learning strategy during the clinical rotations of students from the faculty of health sciences: An action-research study. *Nurse Education Today*, 35(10), 26–31, http://dx.doi.org/10.1016/j.nedt.2015.07.029.
14. Ash, S. & Clayton, PH. (2009). Generating, deepening, and documenting learning: The power of critical reflection in applied learning. *Journal of Applied Learning in Higher Education*, 1, 25–48.
15. Hubbs, D. & Brand, CF. (2010). Learning from the inside out: A method for analyzing Running Head: DON'T WASTE MY TIME reflective journals in the college classroom. *Journal of Experiential Education, 33*(1), 1-8.
16. Lee, O. (2010). Facilitating preservice teachers' reflection through interactive online journal writing. *Physical Educator*, 67(3), 128-139.

17. Harris, M. (2005). Is journaling empowering? Student's perceptions of their reflective writing experience. *Health SA Gesondheid*, 10(2), 1-8.
18. Dyment, JE. & O'Connell, TS. (2011). Assessing the Quality of Reflection in Student Journals: A Review of the Research. *Teaching in Higher Education*, 16(1), 81–97.
19. O'Connell, TS. & Dyment. JE. (2011). The Case of Reflective Journals: Is the Jury Still Out? *Reflective Practice,* 12 (1): 47–59.
20. Babbie, E. & Mouton, J. (2014). *The practice of social research.* Cape Town: Oxford University Press.
21. Pallant, J. (2011). *SPSS Survival Manual*. Berkshire, McGraw Hill.
22. Langley, ME. & Brown, ST. (2010). Perceptions of the use of reflective learning journals in online graduate nursing education. *Nursing Education Perspectives*, 31(1), 12-17.
23. Callister, LN., Luthy, KE., Thompson, P. & Memmott, RJ. (2009). Ethical reasoning in Baccalaureate nursing students. *Nursing Ethics*, 16(4), 499-510.
24. Schuessler, JB., Wilder, B. & Byrd, LW. (2012). Reflective journaling and development of cultural humility in students. *Nursing Education Perspectives*, 33(2), 96–99.
25. Hendrix, TJ., O'Malley, M., Sullivan, C. & Carmon, B. (2012). Nursing student perceptions of reflective journaling: A conjoint value analysis. Available from http://search.proquest.com/. (25/02/2017).
26. Ghaye, T. (2011). *Teaching and Learning through Reflective Practice: A Practical Guide for Positive Action*. New York: Routledge.

~Soli Deo Gloria~