

The Alignment of Software Testing Skills of IS Students with Industry Practices – A South African Perspective

*Elsje Scott, Alexander Zadirov, Sean Feinberg,
and Ruwanga Jayakody
University of Cape Town, Cape Town, South Africa*

escott@commerce.uct.ac.za; AleksandrPZ@ulh.ru;
SeanF@nedcor.com; ruwanga@webmail.co.za

Executive Summary

Software testing is a crucial component in the development of good quality systems in industry. For this reason it was considered important to investigate the extent to which the Information Systems (IS) syllabus at the University of Cape Town (UCT) was aligned with accepted software testing practices in South Africa.

For students to be effective in industry, a close alignment between testing practices in industry and those to which a student is exposed to during undergraduate study is essential. For this reason there is a major emphasis on programming and systems development in the IS degree curriculum at the UCT. A requirement for both third and fourth year IS courses is a systems development group project with the primary goal to expose students to real life systems and thus obtain valuable industry related experience.

In order to determine the extent to which alignment exists between testing approaches used in Industry and those used by students in the IS courses, a survey was undertaken and relevant criteria were identified as the basis for this investigation. These criteria were used to examine the data collected from companies in the software testing industry and students at the University of Cape Town. For each one of the criteria a statistical analysis leading to relevant conclusions was performed.

Significant differences were found between software testing skills required by industry and those claimed by students, particularly with regard to the tests being used and the percentage of time spent on testing. The fact that the students' perceptions differed so much from the ratings given by industry emphasized the need for more exposure of students to current testing methods. Although both students and industry perceived testing as an important part of software development,

Material published as part of this journal, either on-line or in print, is copyrighted by the publisher of the Journal of Information Technology Education. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Editor@JITE.org to request redistribution permission.

it became clear that definitions and ratings could vary between individuals and the potential for subjectivity cannot be ruled out. Results also indicated that the depth of treatment of software testing in the IS course at UCT and for new graduates in general, was insufficient in instructing students to test a system adequately. As no testing framework currently exists in the IS course, the challenge

remains to develop one suitable for inclusion in the system development projects.

The conclusion of the study showed that there was significant scope for revision in the IS syllabus of UCT which could bring closer alignment with industry practices.

Keywords: Software Testing, Testing Categories, Testing Framework

Introduction

Software testing is accepted as an integral part of the systems development process. Testing proves that the system works correctly as it was intended to and that it demonstrates that the developers have understood and met customers' requirements (Tayntor, 1998). To ensure quality it is crucial that systems are tested rigorously, as failing to do so can cause serious malfunctions and a software developer can obtain a reputation for delivering inferior-quality products. For students to be effective in industry, a close alignment between testing practices in industry and those to which a student is exposed to during undergraduate study is essential.

There is a major emphasis on programming and systems development in the Information Systems (IS) degree curriculum at the University of Cape Town (UCT). As students progress through the curriculum each year, the percentage of the overall marks allocated to the practical systems development component of the course increases. A requirement for both third and fourth year IS courses, is a systems development group project.

- Third year student groups are expected to go out into industry to find a sponsor whose business specifications match the generic specifications given to them. Project sponsors then allocate time to meet with students and supply them with information about the particular business.
- Fourth year students go out into industry to identify a related business need. The systems they develop have to improve organizations' efficiency and effectiveness. In many cases these new systems will enable businesses to operate more effectively or even create totally new business opportunities.

A vital component of the group project is to expose students to real life systems and, thus, obtain invaluable industry related experience. Since good testing is an essential component in the production of systems of a high caliber, students need to be equipped to play a critical role in such testing.

This paper provides an overview of a research project undertaken by a group of fourth year Information Systems students. The primary objective of the study was to determine how well the software testing skills and procedures of IS students at UCT are aligned with industry norms in South Africa. Consequently the findings can be applied to improve and/or adapt the curriculum of IS students in South Africa. In order to successfully compare the practices of students to those of industry, it was necessary to investigate what software testing methods are currently employed by industry, as well as how, why and when they are implemented. It was also necessary to determine which of these aspects are covered in UCT's syllabus.

Software Testing

Understanding quality testing is a key factor towards obtaining a better idea of the overall objective of software testing. "Software testing is the process of executing a program or system with the intent of finding errors" (Myers, 1987). Hetzel suggests that "it involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results" (Hetzel, 1988).

Software testing can also be described as the process of testing the functionality and the correctness of software while it is being run. Software testing is usually performed for one of two reasons: (1) defect detection, and (2) reliability estimation. Software testing can, however, only indicate the existence of flaws, not their absence (unless testing is exhaustive). The difficulty with reliability estimation is that the input distribution used for selecting test cases may be flawed. It becomes clear that the entire software testing process is highly dependable on many different variables. If any of these are faulty, the entire process is compromised (Cigital Labs, n.d.).

Information drives today's organizations. Superior systems allow for more efficient business processes. The ability of organizations to deliver superior customer service is crucial to gaining a competitive advantage over their rivals. In order to control risk, time, cost and quality, organizations have to test their systems thoroughly to ensure that these systems fulfill industry requirements and meet their needs. Thorough testing is therefore beneficial to ensure that newly developed systems contain attributes of correctness, reliability, performance, security and usability (Kates, 1998; Zambelich, 1998).

Testing has become a serious undertaking in most organizations (Musa, 1997). It is not just another phase in the system development life cycle (SDLC); it is part of all phases and starts with a thorough risk analysis, a well-planned budget, and sufficient resources to do the work. From there, it encompasses the implementation of well planned testing strategies and the delivery of a reliable system.

Testing Approaches Used in Industry

From the literature study, it was found that the two most commonly documented approaches used in the testing of systems can be categorized as Black Box and White Box testing. Most tests fall under these two highly generalized categories. To comprehensively test systems, testers should develop test plans involving tests from both approaches.

- “The Black Box approach is a testing method in which test data is derived from the specified functional requirements without regard to the final program structure” (Perry, 1994).
- Contrary to this, in white box testing, the structure and flow of data under test are visible to the tester. Because of the transparent nature of this kind of testing, software is often viewed as a White-Box, or Glass-Box. Testing plans are made according to the details of the software implementation, such as programming language, logic, and styles. Test cases are also derived from the program structure.

The hallmark of successful testing is a test plan. This should be a formal, written plan listing each case to be tested along with the expected results. Without a test plan it becomes difficult to ensure that all possible combinations have been tested and that tests can be repeated.

Twelve tests were identified in the literature as being the most widely used in the software development industry. A brief description of each of these tests is given in Table 1.

Various automated testing tools are available to perform tests. These tools are especially useful to eliminate repetitive tasks and in verifying results against predicted ones. They can however not totally eliminate human effort and should not be relied upon to determine what needs to be tested (Tayntor, 1998). These testing tools are normally very expensive and as a result, smaller development houses tend to write their own customized testing tools (Zambelich, 1998).

Table 1: Software Tests Most Widely Performed by Industry

TYPE OF TEST	DESCRIPTION
Unit Testing	Every module that is developed needs to be fully tested. Unit tests forms an integral part in software testing.
Integration Testing	Testing of combined parts of an application to determine if they function correctly.
Acceptance Testing	Testing performed to determine whether software is developed to the satisfaction of an end-user or customer.
Stress Testing	Testing to determine if a web server can handle excessive load.
Load Testing	Testing to determine if response rates are quick enough when a web server experiences normal load conditions.
Performance Testing	Testing used to isolate the exact parts of the web server or web application that slows execution down, in an attempt to improve execution speed.
Regression Testing	Re-testing of systems after modifications have been made, using the same data, to ensure that reported errors have been fixed.
Usability Testing	Testing conducted manually to determine how intended users will interact with a system.
Recovery Testing	Testing to determine how well a system recovers from crashes, hardware failure, or other catastrophic problems.
Security Testing	Testing performed to see how well a system is protected against unauthorized internal or external access, willful damage and other malicious activities.
Compatibility Testing	Testing conducted to determine how well software performs in a particular hardware environment, in conjunction with other software.
Beta Testing	The testing and evaluation of the final pre-released version of software by selected or independent end-users.

Testing Approaches Used in the IS Course

Testing was formally introduced into the IS course on a second year level in 2001. Prior to this there was no formal tuition on testing. The focus at this level is on software quality assurance and the role of testing. Understanding the role of testing provides students with an overview into testing objectives and how testing should be integrated into the SDLC. More details on black box and white box testing are also given and students are expected to apply procedures to exhibit these tests in their project development.

Third year students are required to incorporate appropriate software testing techniques into their systems development project. The main components of the testing approach used by the third year students in 2002, consisted of a Test Plan, Test Cases and a Test Base. The test plan was part

of the marked deliverables of the course and the test cases were included in the final documentation. No formal guidelines are provided for fourth year students thus allowing them to address each problem creatively and within its particular unique context.

Test Plan

A detailed test plan was an important deliverable for the third year group project and was used to describe the different types of tests that would be performed during the different stages of development and testing. This test plan had to define the levels and categories (e.g. functional, user interface, usability, configuration, regression etc.) of testing to be conducted. It also included a list of requirements to be tested/verified, related acceptance criteria agreed to by the sponsor (including performance considerations), roles and responsibilities, tools (if any) and techniques to be used, as well as a testing schedule.

Test Cases

A prescribed test case template was used to transform requirements and expected application behavior into documented test cases. These test cases were expected to not only cover business functions, but to also include user interface verification, field validation and stress testing. Students were expected to perform a selected number of these test cases during the final project presentation.

Test Base

Proper test data had to be prepared and stored in a test database to support the test cases.

The Survey

In order to determine the extent to which alignment exists, a survey was conducted using two groups, namely industry and students. In order to obtain an unbiased and high quality list of companies, it was decided to use the books “The Best IT companies in South Africa, a guide to South Africa’s Most Promising Businesses, 2000”, and the “The Top ICT companies in South Africa, 2001”, as the sampling frame. From these books, the e-mail addresses of 102 companies that practice software testing in South Africa were obtained. From this list 22 (21.6%) companies responded. To increase the sample size and thus give more statistically reliable data, e-mails were also distributed to 488 companies listed in the South African IT directory of CITI (Cape Information Technology Initiative). From this list a further 28 (5.74%) responses were received. The low response rate could possibly be attributed to the fact that companies only had a few days to respond and that it was not easy to, in advance, pinpoint companies that perform software testing. The companies that responded to the questionnaire were diverse in terms of their size and nature, ranging from 30 to more than 1000 employees, from software development houses to cellular service providers.

The student sample included all third and fourth year students who attended their lecture on the day of the survey. While the IS Curriculum at some educational institutions is more theoretically focussed, the IS program at UCT is very technical, with a major emphasis on systems development. The IS student body is diverse and many students might already have extensive IT (Information Technology) experience.

Separate research instruments were designed to cater for the different requirements of the two specific sample populations. The basic requirements for the industry questionnaire were: easy distribution, quick completion, a clear purpose and easy submission / return. A web-based survey

method was used to achieve this. In the students' case the requirements were: the assurance of anonymity, avoidance of personal bias on the part of the authors and easy completion. Paper-based questionnaires were distributed amongst all students in the sample group. Details of the questionnaires can be obtained from the author.

Industry Questionnaire

The web-based questionnaire provided a means of distributing and collecting data nationally at a low cost to both the research team and the respondents.

The industry questionnaire was divided into 6 parts:

- General company information – to determine company details for future reference.
- Graduate Information – to determine if the company employed new graduates, and if they did employ new graduates, what the company's opinions of the graduates' software testing skills were.
- General Testing Information – to obtain an overview of the company's testing procedures to be able to compare with current assumptions in curricula.
- Specific Testing Information – to determine more detailed information pertaining to the company's testing procedures.
- Automated Testing – to determine if the company used any automated testing tools and if they did, to what extent.
- Information about specific tests – to determine the exact methods companies employ to test software and what they thought of graduates' understandings of these procedures.

Student Questionnaire

Paper-based questionnaires that had to be completed and collected after lectures were distributed to 200 third years and 90 fourth year students present at lectures on the day of the survey. This ensured relatively high response rates of 48% and 71% for the third and fourth year student groups, respectively. These groups contained 200 and 90 students respectively.

The student questionnaire was divided into three sections on:

- General Information - to gauge students' level and type of programming experience.
- Testing Methodology – to determine students' sense of testing, the resources they intended allocating towards testing their present systems and the general testing practices they follow.
- Specific Tests – to determine the students' sense of the importance of the most common tests used by industry (identified by the literature review), and also their understanding of each of these tests.

To gain further insight into the testing practices of students, the third year system development test plans were evaluated. These test plans were used to determine which tests the students planned to perform as well as to gauge their level of understanding of software testing, based on evaluation criteria, as shown in Table 2.

Table 2: Evaluation Criteria used to evaluate specific tests in third year test plans.

LEVEL OF UNDERSTANDING	CRITERIA
Low	Did not mention the test at all.
Medium	Mentioned the test briefly, but did not say how it was performed or how it would be applied to the specific system.
High	Mentioned the test in detail, showed good understanding of how it was performed and also produced a plan to apply the test to their system.

Findings of the Study

To fulfill the primary objective of the study, seven criteria were identified for examining the alignment of the software testing skills and procedures of the IS students at UCT with industry standards of South Africa. For each one of the seven criteria listed below, tests were executed to examine the criteria, a statistical analysis was performed, data was summarized and a conclusion was drawn (Zadirov, Feinberg, & Jayakody, 2002).

Student Understanding of Specific Tests

To determine the students’ understanding of specific software tests, ratings of the students’ understanding was compared with industry’s view of the graduate understanding for a number of specific tests.

The students understanding of these specific tests was evaluated in terms of whether students performed the tests and whether or not they understood the tests (regardless of whether they performed the tests or not). To gain an objective perspective of the students’ ability, the third year test plans were collected and analyzed according to the criteria listed in Table 2. The percentage of students possessing skills in each of the specific tests was calculated and compared by means of a t-test with the percentage of companies in the software development industry utilizing these skills.

From these percentages, as shown in Figure 1, and the statistical analysis it can be concluded that:

- students possess relatively high skill levels pertaining to General Validation of data input.
- students possess a lower understanding of Recovery, Compatibility Acceptance, Usability, Security and Beta testing methods.
- statistically there exist significant differences ($p < 0.0001$) between the number of companies performing the tests and the number of students who have the skills to perform the test paragraph formatting.

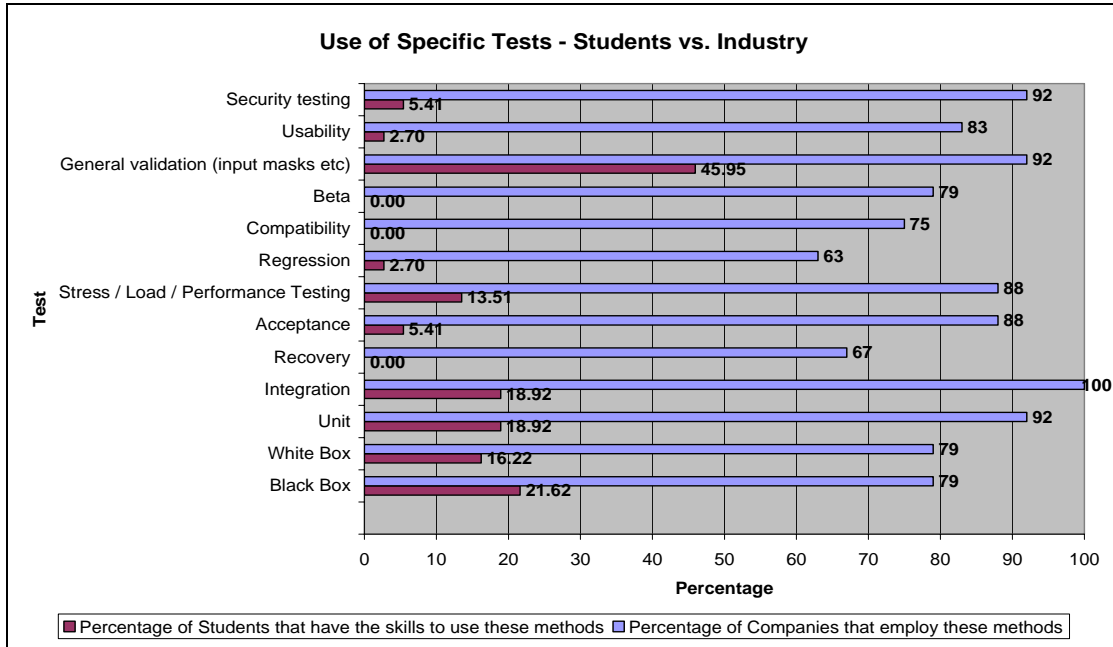


Figure 1: The Percentage of Industry that utilizes certain Software Testing Skills vs. The Percentage of Third Year Students that have the Skill.

The Measurement of Students’ Understanding vs. Industry’s View of Student Understanding

Students were asked to rate their understanding of each test on a scale of 1 to 5. The average understanding of each test was calculated separately for the third and the fourth year classes. Companies who indicated that they hire new graduates were asked to rate these new graduates’ level of understanding for each of the software tests, using the same rating scale.

Results indicated that industry rated students’ understanding significantly lower than students rate their own understanding, as shown in Figure 2. Since this was the case for every test, it can be concluded that extremely poor alignment exists between the students’ perception of their understanding of the tests involved in software testing and that of industry.

The fact that the students’ perceptions differed so much from the ratings given by industry emphasizes the need for more exposure of students to current testing methods.

The Level of Importance of Testing

In order to determine if there is a difference in the perceived level of importance of testing between students and industry, both students and industry were required to rate the importance of testing on a scale from 1 to 5.

Although both the students and industry perceived testing as being an important part of software development, the emphasis that industry places on software testing is significantly more ($p = 0.01$). These results should be seen within the context of the potential for subjectivity when rating importance as one person’s definition of each category of importance might vary, as one person’s rating of extremely important may be equivalent to another person’s rating of very important.

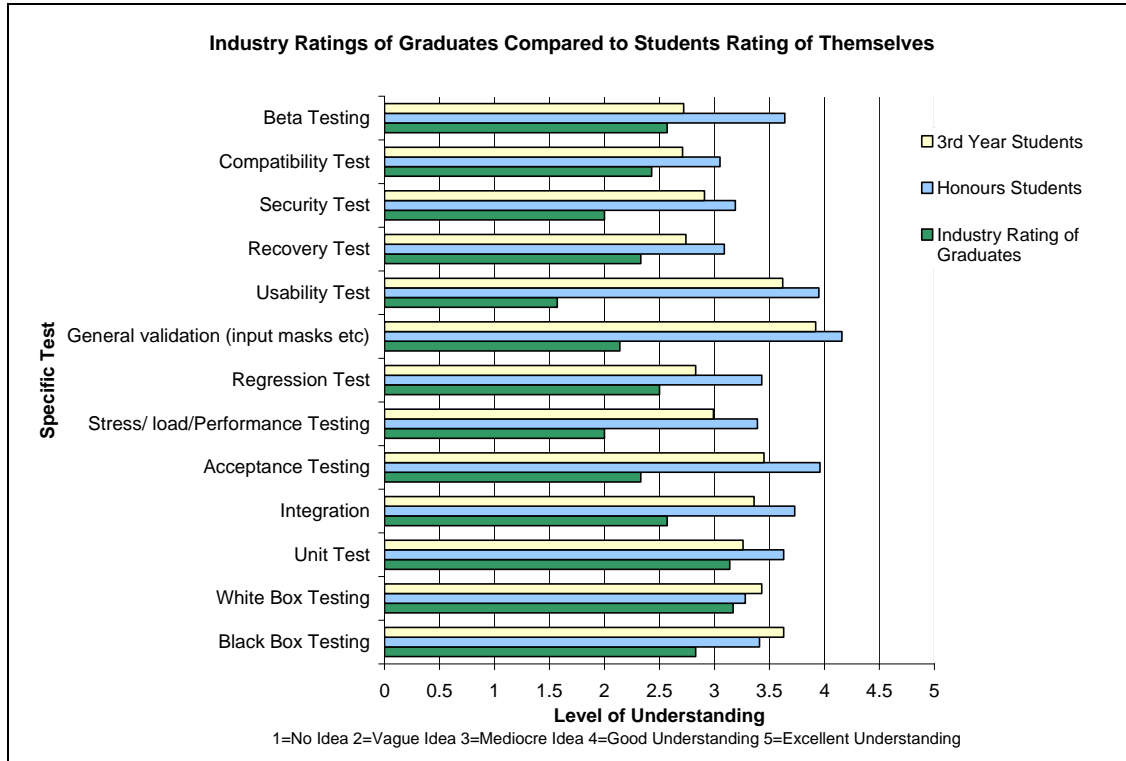


Figure 2: Industry Ratings of Graduates Compared to Students Rating of Themselves

Depth of Treatment of Software Testing

Students were asked what they thought of the depth of coverage of software testing in the IS course at UCT. In addition, companies who indicated that they employ new graduates were questioned as to whether university students are obtaining the necessary skills in software testing, as required by industry.

From the results derived from the above questions it can be concluded that the depth of treatment of software testing in the IS course at UCT and also for new graduates in general, is insufficient in instructing students how to test a system adequately. Results showed that 70% of companies felt that graduates did not know how to adequately test a system. The majority of students do not believe that the depth to which testing is covered in the IS course at UCT is sufficient.

Amount of Time Dedicated to Testing

For both the sample populations it was possible to determine the percentage of software development time allocated for testing. Companies revealed that they spent an average of 40% of development time testing their systems. The percentage calculated for students on the other hand was less than 25%.

From the above result it can be seen that industry allocates a greater percentage of time of their SDLC (software development life cycle) to test their systems than students do.

Provision of Software Testing Frameworks

The necessity for software testing frameworks in systems development courses can be determined from

Alignment of Software Testing Skills

- How widely similar testing frameworks are used in industry.
- The extent to which students feel it is necessary to implement a structured framework to test their systems development project.

A significant proportion (74%) of companies in industry indicated that they use general detailed testing frameworks in testing their software. The majority (83%) of students indicated that they wanted to use a testing framework and would find it useful.

Testing frameworks could be helpful in training students in which tests to do, how to do them and when to do them. This could further ensure that students perform all the required tests correctly. A framework would also be a good educational tool to ensure that the tests are performed correctly. Currently no testing framework exists in the IS courses.

The Use of Automated Testing Tools

The industry questionnaire attempted to determine how many companies used automated testing tools and if so, to what extent. It was found that 82% of them used automated testing tools to perform their tests, and, on average, 47% of the tests used were automated. Students were also asked whether they had ever used automated testing tools. Only about 5% of students indicated that they had ever used these tools. This result is in line with industry's perception that students do not know how to use automated testing tools and need to receive training in them.

It can be concluded that:

- Automated testing tools play a major role in the software testing field.
- A significant difference exists between the percentage of companies and the percentage of students who use automated testing tools.
- Students lack the skills required to use automated testing tools

Summary

Key limitations were identified during the examination of the seven criteria, and can be summarized as follows:

- Students do not test their systems using the same tests as industry and thus their specific software testing skills and procedures are not aligned.
- The majority of students do not have the level of understanding required by industry to test their systems. This proves that their understanding of the tests used by industry in software testing is not in line with those of industry.
- Since students do not place the same value on software testing as industry does, they are unlikely to test their systems with the same rigor that typifies industry practice, and thus are unlikely to gain the necessary skills in software testing.
- Even though students may feel that their skills are aligned with industry, industry responses suggest that their understanding is inferior.
- The depth of treatment of software testing in the IS course at UCT does not provide the students with sufficient knowledge in how to test a system accurately.
- Students spend a smaller percentage of their system development time testing their systems than industry, which further suggests that the testing procedures of students and industry are not aligned.

- The absence of a proper software testing framework in student projects is a further indication of the non-alignment of software testing procedure between students and industry.
- Another significant difference in the testing methods of students and industry is the fact that students are not familiar with automated testing tools.

Conclusion

This study has attempted to determine whether the software testing skills and procedures of the IS students at UCT are aligned with those of industry in South Africa. Although research into the skills used in the software testing industry does exist, there was little literature available as to whether or not these skills are being learnt at UCT.

Research was conducted to investigate the testing approaches of both industry and students. In order to determine whether or not the software testing skills and procedures are aligned, seven criteria were examined.

Although students performed some of the tests described in their test cases during a final project presentation, many of the testing procedures had not been covered in great depth in class, and literature detailing what tests students should perform was limited. The survey also indicated that the number of tests performed by students was less than that found in industry. Automated testing tools were not covered in any of the IS courses, and the students were not provided with a formal testing framework. The fact that the students' perceptions differed so much from the ratings given by industry emphasized the need for more exposure of students to current testing methods. Although both students and industry perceived testing as an important part of software development, it became clear that definitions and ratings could vary between individuals and the potential for subjectivity cannot be ruled out. From the findings listed in the previous section, it can be concluded that the software testing skills and procedures of students and the South African systems development industry are not significantly aligned.

The challenge however remains to develop a suitable testing framework, which can be incorporated into the system development projects of IS students at UCT, not only to improve the quality of student projects, but also to equip our graduates with the skills required by industry.

The research team acknowledges that there are many issues related to the acquisition of knowledge and student learning that had not been addressed and remains an area for further research.

References

- Cigital Labs. (n.d.) [Definition of software testing]. Retrieved November 20, 2002 from http://www.cigitalabs.com/resources/definitions/software_testing.html
- Hetzel, W. C. (1988). *The complete guide to software testing, 2nd edition*. Wellesley, Mass: QED Information Sciences.
- Kates, J. (1998). Hunt for security. *Information Systems Security (Electronic)*, 6 (4), 55-61. Retrieved April 20, 2002 from EBSCO Host: AN: 951180 2002.
- Musa, D. (1997). *Proceedings of the 1997 International Conference on Software Engineering*. New York.
- Perry, N & Roper, M. (1994). *Understanding software testing*. Seattle: John Wiley & Sons.
- Tayntor, C. B. (1998). *Software testing basics and guidelines*. Information Management: Strategy, Systems, and Technologies. Auerbach Publications.
- Zadirov, A., Feinberg, S., & Jayakody, R. (2002). *The alignment of software testing skills and procedures of the information systems students at UCT with that of industry standards in South Africa*. An Empiri-

cal Research Project in partial fulfillment of the requirements for the B.Com (Hons) degree in Information Systems at the University of Cape Town.

Zambelich, K. (1998). *Totally data driven automated testing*. Retrieved April 4, 2002 from http://www.sqa-test.com/w_paper1.html

Biographies



Elsje Scott – Senior Lecturer, Department of Information Systems, University of Cape Town, has 18 years experience in teaching computer programming at tertiary institutions. The author’s main research interests are group projects, software testing, object-oriented programming concepts and general issues concerning the development of efficient computer systems in Information Systems.



Alexander Zadirov – Completed his honours degree at the end of 2002 at the University of Cape Town, with a specific research focus on the software testing skills of IS students.



Sean Feinberg – Completed his honours degree at the end of 2002 at the University of Cape Town, with a specific research focus on the software testing skills of IS students.



Ruwanga Jayakody – Completed his honours degree at the end of 2002 at the University of Cape Town. Ruwanga Jayakody is currently enrolled as a masters student at the same University and busy with further research in the developing of a testing framework for student projects.