The Emerging Trends and Challenges of Software Testing in the Curriculum

Premalatha Sampath
IT Lecturer
Premalatha.Sampath@whitireia.ac.nz

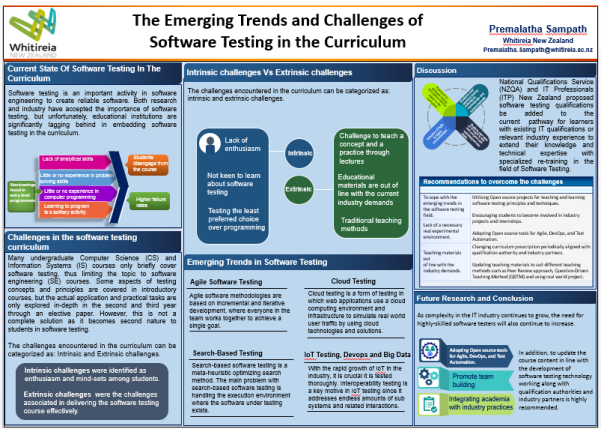
ABSTRACT

Due to the significant importance in software quality, software industries are acknowledging and paying attention to software testing activities. With the increased importance in software quality, the industry requires skilled testers to work in the current, rapidly changing environment. The purpose of this poster is to analyze the emerging trends in the software testing industry and key challenges that polytechnics and universities face when trying to align courses in software testing. The poster also elaborates on ways of improving teaching methods and practices in software testing courses.

Keywords: Software Testing, Teaching methods, Teaching practices.

1. INTRODUCTION

Software testing is important in software engineering to create reliable software (Bertolino, 2007). In 2002 a study conducted by the National Institute of Standards and Technology reported that software bugs cost the US economy an estimated $59.5 billion annually, and found that 1/3 of the cost of all software bugs could be eliminated through improved testing. These numbers would be significantly higher today. Both research and industry have accepted the importance of software testing, but educational institutions are significantly lagging behind in embedding it in the curriculum.



2. CURRENT STATE OF SOFTWARE TESTING IN THE CURRICULUM

It is noted that the subject of software testing is not emphasized in the mainstream of the undergraduate computer science and Information Systems curriculum, despite it being well-recognized in literature and in the industry. Many undergraduate Computer Science (CS) and Information Systems (IS) courses only briefly cover software testing, thus limiting the topic to software engineering (SE) courses. Some aspects of testing concepts and principles are covered in introductory programming courses, but the actual application and practical tasks are only explored in-depth in the second and third year through an elective paper. However, this is not a complete solution as it becomes second nature to students.

2.1 Challenges in the software testing curriculum

Studies have confirmed that educators teaching software testing courses in universities and polytechnics, face some key challenges (Bertolino, 2007; Kurokawa & Shinagawa, 2008; Kaner & Padmanabhan, 2007). The challenges encountered in the curriculum can be categorized as: intrinsic and extrinsic.

Intrinsic challenges were identified as enthusiasm and mind-sets among students. First, the majority of students were not keen to learn about testing and were more interested in software development. Secondly, testing is often viewed as unimportant and as their least preferred subject and therefore they do not learn the purpose of testing to reveal bugs. Thirdly, software testing is not as exciting as development because it is only a subtle feature of an application and requires indirect methods.

Extrinsic challenges were identified as challenges to deliver the curriculum as, in programming courses, testing is not an in-built component to test grades. This is often ignored due to the complexity of grading, which is time-consuming. Finally, it is a significant challenge to teach a concept and a practice through lectures (Kaner & Padmanabhan, 2007). Studies also showed that educational materials are out of line with current industry demands, with teaching methods not aligned to the requirements of current software testing trends (Bin & Shiming, 2013; Sampath, 2015).

3. EMERGING TRENDS IN SOFTWARE TESTING

The emerging trends in the software industry suggest new trends in software testing follow new development strategies and technologies to meet them.

3.1 Agile Software Testing

Agile software testing follows the principles of agile software development. Agile software methodologies

This poster appeared at the 9th annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2018) and the 31st Annual Conference of the National Advisory Committee on Computing Qualifications, Wellington, New Zealand, on July 11-13, 2018 as part of ITx 2018.

are based on incremental and iterative development, where everyone in the team works together to achieve a single goal. Agile software can be represented as:

$$Product_{sw} = \int_{Sprint\ 1}^{Sprint\ n} TW\ (\ TDD^{CI} + PP + \sum R)$$

---formula (1)

In formula (1) *TW* is teamwork, *TDD* is a Team Driven Development, *CI* is a continuous integration, PP is Pair Programming and R is refactoring (Koteska & Dugalic, 2012). Agile testing is to adapt to rapid changes in development and testing with automation the driving force.

3.2 Search-Based Testing

Search-based software testing is a meta-heuristic optimizing search method where the task is automated and the data generated automatically. The main problem with search-based software testing is handling the execution environment where the software under testing exists.

3.3 Cloud Testing

Cloud testing is a form of testing in which web applications use a cloud computing environment and infrastructure to simulate real world user traffic by using cloud technologies and solutions. This testing is primarily intended for dynamic, distributed and component-based applications that are flexible and connected to real-time data. Cloud testing is a new approach to simulate real-world user traffic.

3.4 IoT Testing, Devops and Big Data

With the rapid growth of IoT in the industry, it is crucial it is tested thoroughly. Interoperability testing is a key motive in IoT testing since it addresses endless amounts of sub systems and related interactions. Therefore this requires fast, responsive Quality Assurance and testing solutions integrated with agile development.

4. DISCUSSION

Students should be exposed to software testing concepts and techniques throughout various papers in the undergraduate curriculum. Students could start applying testing techniques (even in their introductory courses) early on and begin to test their programs systematically and methodically with well-designed test-cases. The adoption of Agile principles in the curriculum is highly recommended to suit industry expectations.

National Qualifications Service (NZQA, 2017) and IT Professionals (ITP) New Zealand proposed software testing qualifications be added to the current pathway for learners with existing IT qualifications or relevant industry experience to extend their knowledge and technical expertise with specialized re-training in the field of Software Testing.

Table 1: Recommendations to overcome the challenges encountered in the Software Testing curriculum.

To cope with the emerging trends in the software testing field.	Utilising Open source projects for teaching and learning software testing principles and techniques.
Lack of a necessary real experimental environment.	Encouraging students to become involved in industry projects and internships. Adopting Open source tools for Agile, DevOps, and Test Automation.
Teaching materials out of line with the industry demands.	Changing curriculum prescription periodically aligned with qualification authority and industry partners. Updating teaching materials to suit different teaching methods such as Peer Review approach(Smith, Tessler, Kramer, & Lin, 2012), Question-Driven Teaching Method (QDTM) (Mao, 2008) and using real world project (Krutz, Malachowsky, & Reichlmayr, 2014).

5. CONCLUSION

As complexity in the IT industry continues to grow, the need for highly-skilled software testers will also continue to increase. After analysing the industry expectations and current teaching methods, it is recommended to adopt Agile principles and reform teaching practices to effectively deliver this paper. In addition, to update the course content in line with the development of software testing technology working along with qualification authorities and industry partners is highly recommended.

6. REFERENCES

Bertolino, A. (2007). *Software testing research: Achievements, challenges, dreams.* Paper presented at the Future of Software Engineering. FOSE '07. Minneapolis, MN, USA. doi:10.1109/FOSE.2007.25

Bin, Z., & Shiming, Z. (2013). *Curriculum reform and practice of software testing.* Paper presented at the International Conference on Education Technology and Information System (ICETIS 2013). Atlantis Press.

Kaner, C., & Padmanabhan, S. (2007, 3-5 May). *Practice and transfer of learning in the teaching of software testing.* Paper presented at the 20th Conference of the Software Engineering Education & Training: CSEET'07, Dublin, Ireland. doi: 10.1109/CSEET.2007.38

Koteska, B., & Dugalic, B. (2012). *Recent and Future Trends and Challenges of Software Testing.* Paper presented at the 9th Conference for Informatics and Information Technology (CIIT 2012), Macedonia. Retrieved from http://ciit.finki.ukim.mk/data/papers/9CiiT/9CiiT-69.pdf

Krutz, D. E., Malachowsky, S. A., & Reichlmayr, T. (2014). *Using a real world project in a software testing course*. Paper presented at the Proceedings of the 45th ACM technical symposium on Computer science education, Atlanta, Georgia, USA. doi:10.1145/2538862.2538955

Kurokawa, T., & Shinagawa, M. (2008). Technical trends and challenges of software testing. Science & Technology Trends, 10(1), 34–45.

Mao, C. (2008). *Towards a question-driven teaching method for software testing course.* Paper presented at the International Conference on Computer Science and Software Engineering, Hubei, China. Retrieved from https://ieeexplore.ieee.org/document/4722985/

NZQA (2017). *Development of Security and Testing qualifications (2017- 2018).* Retrieved from http://www.nzqa.govt.nz/qualifications-standards/qualifications/information-technology-and-computing-qualifications-at-levels-1-6/development-of-security-and-testing-qualifications/

Sampath, P. (2015, Oct 6-9). *The Emerging Role of Software Testing in Curriculum.* Poster presented at the Computing and Information Technology Research and Education New Zealand (CITRENZ), Queenstown, New Zealand. CITRENZ.

Smith, J., Tessler, J., Kramer, E., & Lin, C. (2012, Sep 09 - 11). *Using peer review to teach software testing*. Paper presented at the 9th annual international conference: International computing education research, Auckland, New Zealand. doi: 10.1145/2361276.2361295