Teaching Software Testing Skills:

Metamorphic Testing as Vehicle for Creativity and Effectiveness in Software Testing

Dave Towey
School of Computer Science
The University of Nottingham Ningbo China
Zhejiang 315100, People's Republic of China
Dave.Towey@nottingham.edu.cn

Tsong Yueh Chen
Department of Computer Science and Software Engineering
Swinburne University of Technology
Victoria 3122, Australia
tychen@swin.edu.au

***Abstract*—In spite of its importance to software quality, software testing is often considered the "poor man" of software engineering processes, left to the end of many projects, and frequently omitted altogether. Compounding this is the reported perception of testing as a menial, low-level job, lacking any need for creativity or ingenuity, and of much lower status and attractiveness than others, such as design and implementation. When teaching about software testing, instructors often face very unmotivated students, professing little or no interest in becoming testers. This tutorial will address some testing misconceptions, and, through Metamorphic Testing – a new approach to testing which alleviates some of the major challenges in the field (including the Oracle Problem) – will provide a fresh and exciting new perspective on software testing.**

Keywords—Metamorphic Testing; Software Testing; Training; Software Quality Assurance.

I. INTRODUCTION

Software testing is a software quality assurance activity that involves execution of the software under test (SUT) in an attempt to identify problems, such as bugs. Traditionally, testing has often been left to later stages of the software engineering lifecycle, and, when time or finances have been constrained, even been omitted entirely. An important aspect of testing is how to identify whether output from the SUT is correct or not, and a mechanism that can do so (such as a previous version of the SUT) is called an oracle. When an oracle does not exist, or exists but cannot feasibly be used, the tester faces the Oracle Problem, a significant challenge to testing.

Another challenge to the testing community is the reported perception of the testing process as being low-level, and requiring little imagination or skill – an issue not helped by many current teaching and training approaches for testing, with their emphasis on extensive checklists and application of repetitive, mundane activities.

Metamorphic Testing (MT) [1] is an approach to software testing that alleviates the Oracle Problem by, rather than addressing individual output correctness, focusing instead on properties and relationships of the SUT, called Metamorphic Relations (MRs). Using multiple executions of the SUT, the impact on the MRs can be used to identify potential bugs. A simple example might be, if implementing the sine function, then using the MR of $sin(x) = sin(2\pi + x)$: Although exact determination of the output's correctness for any arbitrary value of x may not be possible, a comparison of this output with that for $(2\pi + x)$, after consideration of rounding errors, should be relatively easy. With such an MR, if the outputs are not equal, then a bug must be present.

In addition to alleviating the Oracle Problem, and being very effective at finding problems, MT provides a new perspective on testing – one that encourages and rewards creativity and diversity in the tester's approach [2].

II. TUTORIAL OBJECTIVES

By the end of the tutorial, it is expected that participants will be able to apply MT techniques to testing software, and will be able to use creativity and ingenuity in the identification of MRs and the creation of test cases. Where appropriate, participants will leave the tutorial with a clear idea of how to apply MT in their own contexts – professional or academic.

III. TUTORIAL FORMAT

It is expected that the entire tutorial will take about three hours, arranged in four parts, as follows:

The first part will be a presentation of the basic history and terminology related to software testing. Through this, participants with no software engineering or testing background will quickly become familiar with the essentials.

This will be followed by an interactive session introducing MT, during which the instructors will use example programs and algorithms to help participants understand and apply MT, identify MRs, and evaluate which MRs may be best.

A third part will involve participants working in teams and applying the MT skills to a set of different programs, again identifying MRs, and generating test cases. Instructors will be on hand providing clarification and feedback.

The final part may group participants according to academic or professional contexts, and provide practical guidance for they can use MT in their work.

IV. INTENDED AUDIENCE

Although this tutorial is about software testing, there is no need for any special knowledge, skills, or experience on the part of the participants. All necessary information will be provided, and the appropriate skills demonstrated and taught during the tutorial.

978-1-4673-9226-6/15/$31.00 ©2015 IEEE 10-12 December 2015, United International College, Zhuhai, China
2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)

V. INSTRUCTORS

A. Dave Towey

Dave Towey received his BA and MA degrees from The University of Dublin, Trinity College; PgCertTESOL from The Open University of Hong Kong; MEd from The University of Bristol; and PhD from The University of Hong Kong. He was the first foreign academic recipient of the Zhuhai municipal outstanding teacher award, in 2007, and he is currently an Assistant Professor at The University of Nottingham Ningbo China (UNNC). Prior to moving to UNNC, he was with the Beijing Normal University-Hong Kong Baptist University: United International College (UIC), where he had a number of roles, including Deputy Director of the English Language Center and Coordinator (Director) of the Teaching English as a Second Language (TESL) degree programme. In these roles, he oversaw a large number of pre-service and in-service training programs and workshops. His research interests include technology-enhanced teaching and learning, and software testing, especially metamorphic testing.

B. Tsong Yueh Chen

Tsong Yueh Chen, an originator of the metamorphic testing approach, received his BSc and MPhil from The University of Hong Kong, MSc and DIC from the Imperial College of The London University, and PhD from The University of Melbourne. He is currently a Professor of Software Engineering and the Leader of the Software Analysis and Testing Group at Swinburne University of Technology, Australia. Prior to joining Swinburne, he taught at The University of Hong Kong and The University of Melbourne. His main research interests include software testing and debugging, about which he has published extensively.

REFERENCES

[1] H. Liu, F.-C. Kuo, D. Towey, and T. Y. Chen (2014), "How effectively does metamorphic testing alleviate the oracle problem?" IEEE Transactions on Software Engineering 40(1): 4-22.

[2] T. Y. Chen, F.-C. Kuo, D. Towey, and Z. Q. Zhou (2015), "A revisit of three studies related to random testing," SCIENCE CHINA Information Sciences, vol. 58, no. 5, pp. 052104:1–052104:9.