MAP-I: Doctoral Program in Informatics
2013-14 Edition

Proposal of a Curriculum Unit in Technologies

Software Test Automation and Quality Engineering

Ana Paiva
João Pascoal Faria
UPorto

Ricardo J. Machado
UMinho

May, 12, 2013

Abstract: This document describes a Ph.D. level course, corresponding to a curriculum unit credited with 5 ECTS. It corresponds to a joint UPorto-UMinho proposal for OPTI (option in technologies) in the joint MAP-I doctoral program in Informatics. It is presented the programmatic component, the lecturing team, and the plans for the coordination with curriculum units in other PhD programs.

A. Programmatic Component

1. Theme, Justification and Context

Motivation

Software is increasingly present in our daily life. The software controls 80% of the functions of military aircraft, motor controls, TVs, mobile phones etc. Moves daily, one billion dollars in the financial market, and is crucial for several businesses. Thus, the quality of software is becoming increasingly important. But assuring the quality of increasingly complex software systems is a challenging task. Failures in software are all too common and have a huge economic impact. According to NSF, the economic impact of inadequate software testing is estimated to represent 0.6% of the USA's GNP.

In the software industry, software testing typically consumes 30% to 50% of the development effort and schedule. Yet, the quality of delivered software products is not satisfactory. Considering the typical defect density of 1 to 7 defects/KLOC in delivered software products, typically with millions of lines of code, leads to a number of defects in delivered products in the order of thousands of defects. Hence, it is of upmost importance to improve testing efficiency and effectiveness, to reduce testing costs and improving product quality, namely through test automation.

Software testing is recognized as a major knowledge area or subject of software engineering, both by the IEEE's Guide to the Software Engineering Body of Knowledge (SWEBOK) and the ACM Computing Classification System.

Improving software testing alone, may not produce the desired cost reduction and quality improvement results, if not complemented with other quality engineering activities, like defect analysis and prevention, static verification and validation techniques (inspections, reviews, and static analysis), quality measurement, risk management, and continuous process improvement.

2. Objectives and Learning Outcomes

This course aims to cover, both from the foundational and the methodological point of view, the concepts and techniques for Software Test Automation and Quality Engineering.

The course is not intended as an introductory survey on Software Test Automation and Quality Engineering, but as an opportunity of exposing students to cutting-edge research topics in this area, although presented in a coherent and integrated way. It is placed at a similar level and covers overlapping material with advanced modules in doctoral programs at leading academic institutions.

Upon successful completion of this curricular unit, students should be able:

- to recognize and explain the economic and social importance of software quality;
- to understand the benefits and challenges of automated software testing;
- to identify test related measures (efficiency, effectiveness, coverage, etc.);
- to understand the principles and techniques for test suite minimization and prioritization;
- to understand the types, benefits and best practices of software reviews and inspections;
- to understand and explain the relationships between process quality and product quality.
- to promote the continuous assessment and improvement of software processes;
- to identify and describe existing process assessment models and methods;
- to understand the importance and techniques for defect causal analysis and prevention;
- to use quantitative techniques in product and process quality assessment and improvement;
- to identify and describe exiting standards and reference models for software process and product certification.

3. Course Contents

The course is organized in two units, corresponding to the software product quality and process quality perspectives.

3.1. SOFTWARE TEST AUTOMATION

This unit covers concepts, methods and techniques for software product quality evaluation and defect detection and removal.

Our society is increasingly dependent on the correct functioning of software systems, so the software industry should strive to deliver essentially defect free software, by using more effective and efficient defect detection and prevention techniques than are in common use today. The goal of this unit is to cover some of the most promising techniques in that direction.

It includes both dynamic techniques (software testing), that involve program execution, and static techniques (manual reviews and inspections and automated static analysis), that involve the analysis of static software representations. Testing is the main defect detection technique in software industry. Manual testing is infeasible because of the ever growing size and complexity of software systems, so the focus in this unit will be placed in the study of test automation techniques, notably automatic test generation. Among others, the value of static analysis is early detection of defects prior to test execution, and evaluation of internal quality properties and adherence to standards.

This unit will cover the following topics:

- Software testing fundamentals (test levels, test types, metrics, etc.)
- Random, structural, fault-based/mutation and statistical testing techniques
- Automatic test path, test data and test code generation
- Automatic model-based testing (from patterns, algebraic specifications, state-based specifications, UML, Alloy, etc.)
- Automatic GUI testing
- Intelligent monkey testing (machine learning + model-based testing)
- Automatic synthesis of mock objects
- Testing for mobile devices
- Load/stress testing
- Automated evaluation of product quality characteristics
- Self-testing systems
- Test coverage analysis and code instrumentation
- Combining testing with static analysis techniques
- Test suite minimization and prioritization for regression testing
- Test as a Service (TaaS) and SLAs
- Test Maturity Model (TMM)

3.2. SOFTWARE QUALITY ENGINEERING

This unit focuses on approaches to improve the quality of software processes. An underlying premise of this unit is that the quality of software products is largely determined by the quality of the processes used to develop them. Another premise is that it is not possible to manage and improve a process if its performance is not measured.

The unit focuses on the following topics:

- Software quality fundamentals (quality economics, quality principles, quality management)
- Software product quality models and metrics (ISO 9126, ISO 25010, etc.)
- Software inspections and reviews
- Overview of process management and continuous improvement lifecycles
- Process definition and modeling notations (SPEM, etc.)
- Process assessment models and methods (CMMI, SCAMPI, ISO/IEC 15504, etc.)
- Process assurance techniques (audits, checklists, tools, etc.)
- Process performance measurement (metrics, benchmarks, etc.)
- Process performance models (for performance prediction, root cause analysis, etc.)
- Evaluation and prioritization of process improvement proposals
- Defect-prevention based SPI

4. Teaching Methods and Student Assessment

Studies show the need to consider software quality as a multidisciplinary activity that requires systematization. This course will allow:

- learning the fundamental concepts and principles of software quality;
- knowing and understand the solutions and proven practices to improve software quality through exploration of examples and case studies;
- applying the knowledge gained by using and adapting known solutions for a particular problem in an individual project.

No textbook adequately covers the course's range of topics, so a diversity of bibliographic elements (books, journals and conference proceedings) will be used.

Classes

The class meetings are meant to be conversational, and we encourage students to ask questions and make comments. Consequently, the discussion may follow tangents to the prepared lecture, but they should be fruitful, informative, and thought provoking. These classes are conducted by all the elements of the lecturing team.

Panels and talks

Two panels or talks will be organized to complement the topics covered by the formal classes. These panels or talks will involve both the lecturing team and possibly faculty members that are not formally associated with this UC.

Readings

Each week, the students must read papers or some few supplemental materials provided, related to the topics discussed in class. All reading assignments come from journals and conference proceedings. This exposes many students to extensive readings from the research literature for the first time. To help them with their reading, we require them to write a brief summary for each paper, and submit it electronically before the next class. We also ask them to submit a list of questions about the readings, which we try to work into the lecture if possible. During the last few weeks of the course, we no longer require reading summaries, to give students more time to focus on the project.

Individual research project

Whilst the goal of the readings is to develop a critical but shallow view over a broad range of topics addressed in classes, the goal of the individual research project is to develop a more in depth understanding on software quality research topic, matching as much as possible each student interests, as well as the lecturers areas of expertise. Projects are designed to combine a state of the art analyses with an experimental assessment. Grading of individual research projects if based on an oral presentation (for a more methodological project) or demonstration (for a more technological project), and a final written report. No two students can work on the same project. A few weeks into the course, descriptions of possible projects are handed out to students, who are also encouraged to propose projects of their own. Once students complete their project, they must demonstrate it, make an oral presentation, and submit a final written report.

5. Basic Bibliographic References

- Utting M, Legeard M. *Practical Model-Based Testing: A Tools Approach*, Morgan Kaufmann, 2007.
- Burak Turhan and Ayşe Bener, A Multivariate Analysis of Static Code Attributes for Defect Prediction.
- R.M. Poston, Automating Specification-based Software Testing, IEEE Press, 1996.
- Software Verification and Analysis: An Integrated, Hands-On Approach, by Janusz Laski and William Stanley (Nov 5, 2010).
- Ian Somerville, Software Engineering, 9th edition, Addison-Wesley, 2010
- K. El-Emam and N. Madhavji, Elements of Software Process Assessment and Improvement, IEEE Computer Society Press, 1999.
- Christof Ebert, Reiner Dumke. Software Measurement: Establish - Extract - Evaluate – Execute. Springer-Verlag, 2007.
- Marcos Kalinowski, Emilia Mendes, David N. Card and Guilherme H. Travassos. Applying DPPI: A Defect Causal Analysis Approach Using Bayesian Networks, Proceedings of the 11th International Conference on Product- focused Software Process Improvement (PROFES 2010), Lecture Notes in Computer Science, 2010, Volume 6156/2010, pages 92-106.

B. Lecturing Team

1. Team Presentation

This course is supported by a team involving researchers from the University of Porto (Ana Paiva e João Pascoal Faria) and the University of Minho (Ricardo J. Machado).

All team members are working, and have worked actively in the past few years, on topics that are directly related to the subjects covered by this course, as detailed below.

2. Coordinator

The coordinator of the unit is Ana Paiva.

3. Short Presentation of Team Members

In the sequel we introduce a brief presentation of each team member, which includes, for each of them, up to 5 key publications related to the scientific area in which this course is proposed. All CVs are supplied in separate PDF documents.

Ana Paiva is assistant Professor at the Informatics Engineering Department of the Faculty of Engineering of University of Porto (FEUP) where she works since 1999. She teaches subjects like Software Testing, Formal Methods and Software Engineering, among others. She belongs to the group on Software Engineering (softeng.fe.up.pt) which gathers researchers and post graduate students with common interests in software engineering. She has a PhD in Electrical and Computer Engineering from

FEUP with a thesis titled "Automated Specification Based Testing of Graphical User Interfaces". Her expertise is on the implementation and automation of the model based testing process. She has been developing research work in collaboration with Foundation of Software Engineering research group within Microsoft Research where she had the opportunity to extend Microsoft's model-based testing tool, Spec Explorer, for GUI testing. She was a member of the CYTED network on software verification and validation (REVVIS), she is vice-president of the PSTQB (Portuguese Software Testing Qualification Board) board (www.pstqb.pt), member of the Council of the Department of Informatics Engineering, and member of the Council of Representatives of FEUP.

Key Publications:

- Pattern Based GUI Testing Modeling Environment, in 4th International Workshop on Testing & Experimentation Benchmarks for Even-Driven Software (TESTBEDS), Tiago Monteiro, Ana C. R. Paiva, Luxembourg, 18 March, 2013.
- Specification-driven Unit Test Generation for Java Generic Classes , in iFM - 9th International Conference on Integrated Formal Methods, Francisco R. de Andrade, João P. Faria, Antónia Lopes, Ana C. R. Paiva, Pisa, Italy, June 18-21, 2012.
- Test case generation from mutated task models, in the ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS'11), Ana Barbosa, Ana C. R. Paiva, José Creissac Campos, Pisa, Italy - June 13-16, 2011.
- Model-based user interface testing with Spec Explorer and ConcurTaskTrees, in Electronic Notes in Theoretical Computer Science, Volume 208, Issue C, pp. 77-93, ISSN: 15710661, José L. Silva, José Creissac Campos, Ana C. R. Paiva, 14 April 2008.
- Towards the Integration of Visual and Formal Models for GUI Testing, in Electronic Notes in Theoretical Computer Science, Volume 190, Issue 2 SPEC. ISS., pp. 99-111, ISSN: 15710661, Ana C. R. Paiva, João P. Faria, Raul M. Vidal, 31 August 2007.

João Pascoal de Faria is assistant professor at the Department of Informatics Engineering of Faculty of Engineering of University of Porto (FEUP), researcher at INESC Porto, and co-founder of Strongstep. He is vice-president of the "Comissão Sectorial para a Qualidade nas Tecnologias de Informação e Comunicações (CS03)" of the "Conselho Nacional para a Qualidade" (CNQ). In the past, he also worked with several software companies (Novabase, Sidereus and Medidata). He has more than 20 years of experience in teaching, research and consultancy in several software engineering areas. He is the main author of a rapid application development tool, based on domain specific languages, with more 20 years of market presence and evolution (1989-2011). Since 2008, he is a Certified Personal Software Process (PSP) Developer, Authorized PSP Instructor, and Team Software Process (TSP) Coach by the Software Engineering Institute of the Carnegie Mellon University. He is currently involved in research projects and supervisions in the areas of model-based testing, model-driven development and software process improvement.

Key Publications:

- César Duarte, João Pascoal Faria. PSP PAIR: Automated Performance Analysis and Process Improvement Recommendation. QUATIC 2012
- Isabel Lopes Margarido, João Pascoal Faria, Raul Moreira Vidal, Marco Vieira. *Towards a Framework to Evaluate and Improve the Quality of Implementation of CMMI® Practices*, in PROFES 2012

- GUI Reverse Engineering with Machine Learning, in RAISE'12 Workshop on Realizing Artificial Intelligence Synergies in Software Engineering, Zurich, Switzerland, 17 Feb, 2012.
- Andrade FR, Faria JP, Paiva A; *Test Generation from Bounded Algebraic Specifications using Alloy*, 6th International Conference on Software and Data Technologies (ICSOFT 2011), Seville, Spain, 2011
- Cruz AM, Faria JP; *A Metamodel-based Approach for Automatic User Interface Generation*, ACM/IEEE 13th International Conference on Model Driven Engineering Languages and Systems (MODELS'2010), LNCS, Volume 6394/2010, pp 256-270, 2010 (best paper award)

Ricardo J. Machado is associate professor at the Department of Information Systems and associate director of the ALGORITMI Research Centre, both from the School of Engineering of University of Minho. He leads the SEMAG research group at the ALGORITMI Research Center and coordinates de EPMQ Laboratory at the CCG/ZGDV Institute. He holds the presidency of the General Assembly of TICE.pt (Pólo de Competitividade em Tecnologias de Informação, Comunicação e Electrónica), since 2009, and holds the presidency of the CT 128 (the Portuguese Technical Committee responsible for analyzing the documents produced by JTC1/SC7 from ISO/IEC and by TC311 from CEN/CENELEC in the software and system engineering domain), since 2004. He coordinated the IEEE Computer Society Chapters and Student Chapters in Region 8 (Europe, Middle East and Africa), in 2007-2008, and coordinated the Portuguese representation in the IFIP TC10: Technical Committee in Computer Systems Technology (the IFIP Committee responsible for the promotion of the State-of-the-Art and the coordination of information exchange on concepts, methodologies, and tools in the stages in the life cycle of computer systems), in 2006-2009. His research focuses on software engineering & management, namely on model-driven development, requirements engineering, and software quality. He has lead several research projects resulting in more than 80 publications. He has been involved in the organization of various international events, including ACSD 2003/10, DIPES 2006, QUATIC 2007/10/12, IEEEXtreme 2008, Petri Nets 2010, and the MOMPES workshops series. He received the 2009 IEEE MGA Achievement Award.

Key Publications:

- Ferreira N, Santos N, Machado RJ, Gasevic D; *Derivation of Process-Oriented Logical Architectures: An Elicitation Approach for Cloud Design*. 13th International Conference on Product-Focused Software Development and Process Improvement (PROFES'2012), Madrid, Spain, LNCS Series, Springer, Jun/2012.
- Borges P, Monteiro P, Machado RJ; *Mapping RUP Roles to Small Software Development Teams*. S. Biffl, D. Winkler, J. Bergsmann (Eds.), Software Quality: Process Automation in Software Development, pp. 59-70, LNBIP Series vol. 94, Springer, Jan/2012.
- Azevedo S, Machado RJ, Bragança A, Ribeiro H; *On the Refinement of Use Case Models with Variability Support*. Innovations in Systems and Software Engineering, 8(1):51-64, Springer Jan/2012
- Ferreira AL, Machado RJ, Paulk MC; *Supporting Audits and Assessments in Multi-Model Environments*. Danilo Caivano, Markku Oivo, Maria Teresa Baldassarre, Giuseppe Visaggio (Eds.), Product-Focused Software Process, pp. 73-87, LNCS Series vol. 6759, Springer, Jun/2011.
- Monteiro P, Machado RJ, Kazman R, Henriques C; *Dependency Analysis between CMMI Process Areas*. M. Ali Babar, Matias Vierimaa, and Markku Oivo (Eds.), Product-Focused Software Process Improvement, pp. 263-275, LNCS Series vol. 6156, Springer, Jun/2010.