FIT4004
System validation and verification, quality and standards

Unit Guide

Semester 1, 2015

The information contained in this unit guide is correct at time of publication. The University has the right to change any of the elements contained in this document at any time.

Last updated: 24 Feb 2015

Table of Contents

FIT4004 System validation and verification, quality and standards - Semester 1, 2015

This unit covers the fundamental products, processes and techniques for system validation and verifications including testing methodologies, static program analysis and code quality measurement and monitoring. Open-source tools will be used to apply in practice knowledge learnt about software testing from a theoretical perspective. Inspection and testing methodologies, analysis of artefacts, robustness, performance analysis configuration management, quality assurance plan and standards, compliance, assessment, certification issues are covered. It shows how to predict, analyse and control defects in complex software systems.

Mode of Delivery

Clayton (Day)

Workload Requirements

Minimum total expected workload equals 12 hours per week comprising:

(a.) Contact hours for on-campus students:

- Two hours of lectures
- One 1-hour tutorial

(b.) Additional requirements (all students):

- A minimum of 2-3 hours of personal study per one hour of contact time in order to satisfy the reading and assignment expectations.

See also Unit timetable information

Additional workload requirements

You will need to allocate one hour per week for unsupervised lab/tute activity in the MUSE Lab to get familiarised with tools, work on assignments and for self study.

Unit Relationships

Prohibitions

CSE4431

Prerequisites

MAT1830 and FIT2004 and FIT2024 and FIT3077

Chief Examiner

Dr Robert Merkel

Campus Lecturer

Clayton

Robert Merkel

Consultation hours: TBA - check Moodle page

Tutors

Clayton

Hong-Li Song

Your feedback to Us

Monash is committed to excellence in education and regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through the Student Evaluation of Teaching and Units (SETU) survey. The University's student evaluation policy requires that every unit is evaluated each year. Students are strongly encouraged to complete the surveys. The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

For more information on Monash's educational strategy, see:

www.monash.edu.au/about/monash-directions/ and on student evaluations, see:
www.policy.monash.edu/policy-bank/academic/education/quality/student-evaluation-policy.html

Previous Student Evaluations of this Unit

Previous SETU feedback indicated problems with practical relevance and the technology stack used in assignments.

The course material has been completely rewritten from scratch with a greater practical focus, and the technology stack has been changed and simplified to bring the focus back on the testing and quality aspects rather than dealing with the limitations of unreliable tools.

If you wish to view how previous students rated this unit, please go to
https://emuapps.monash.edu.au/unitevaluations/index.jsp

Academic Overview

Learning Outcomes

On successful completion of this unit, students should be able to:

1. explain the importance of quality assurance in software engineering;
2. articulate the role of validation and verification methods in the system development life cycle; key issues in software testing, testing levels and testing techniques;
3. categorise and apply selection and combination of techniques and test related measures;
4. measure, evaluate and analyse software under test using different quality and complexity metrics;
5. develop adequate test cases to help detect software system defects using industry-strength IDEs, unit testing frameworks such as JUnit, code coverage tools such as Cobertura, and other similar products;
6. implement continuous integration (CI) at unit, integration & system testing level using a CI server such as Jenkins to automatically run regression test suites on the system under test.

Unit Schedule

Week	Activities	Assessment
0		No formal assessment or activities are undertaken in week 0
1	Overview, Fundamentals of software quality assurance	
2	Quality Assurance Plans	Weekly assessed Tutorial Participation starts in Week 2
3	Human review	
4	Types of testing	
5	Black box testing	Artifact Review due Week 5
6	White box testing	
7	Documenting testing	
8	Unit testing	Manual System Testing due Week 8
9	Unit testing II	
10	Integration Testing	
11	Metrics	
12	Frontiers of testing	Automated Unit Testing due Week 12
	SWOT VAC	No formal assessment is undertaken in SWOT VAC
	Examination period	LINK to Assessment Policy: http://policy.monash.edu.au/policy-bank/academic/education/assessment/assessment-in-coursework-policy.html

*Unit Schedule details will be maintained and communicated to you via your learning system.

Teaching Approach

Lecture and tutorials or problem classes

This teaching and learning approach provides facilitated learning, practical exploration and peer learning.

Assessment Summary

Examination (2 hours): 50%; In-semester assessment: 50%

Assessment Task	Value	Due Date
Artifact Review	10%	Week 5
Manual System Testing	15%	Week 8
Automated Unit Testing	15%	Week 12
Tutorial Participation	10%	Weekly (starts in Week 2)
Examination 1	50%	To be advised

Unit Schedule

Assessment Requirements

Assessment Policy

Faculty Policy - Unit Assessment Hurdles
(http://intranet.monash.edu.au/infotech/resources/staff/edgov/policies/assessment-examinations/assessment-hurd

Academic Integrity - Please see resources and tutorials at
http://www.monash.edu/library/skills/resources/tutorials/academic-integrity/

Assessment Tasks

Participation

Students are expected to attend all tutorials and participate in active discussions. Tutorials will carry 10% assessment and will be assessed on participation.

- **Assessment task 1**

  Title:
  Artifact Review
  Description:
  Students will review provided software engineering artifacts, such as (but not limited to) requirements documents, design documents, or source code, using a structured review process. They will develop checklists for reviewing the artifacts, conduct the review according to the recommended process, and provide a written report of the review.
  Weighting:
  10%
  Criteria for assessment:
  This assignment will be assessed on:

  - ♦ appropriateness of checklists based on the artifact type and the context of the provided artifact.
  - ♦ conformance with review process.
  - ♦ thoroughness of review process.
  - ♦ appropriateness of recommendations.

  Due date:
  Week 5

- **Assessment task 2**

  Title:
  Manual System Testing
  Description:
  Students will be provided with a software system, and associated artifacts, for which they will design and perform manual system testing.

  Students will choose, and document, an appropriate testing strategy based on the quality goals for the system.

  They will then devise a detailed suites of tests for the system, and document these for others to carry out.

Students will then swap test suites, and conduct manual system testing based on a fellow student's test suite, and report the results of testing. They will also be asked to comment on the quality of the test suite that they used to conduct the manual testing.

Weighting:
15%

Criteria for assessment:

♦ Appropriateness of chosen test strategy.
♦ Selection of test cases according to the strategy.
♦ Comprehensiveness and accuracy of documentation
♦ Effectiveness of test suite.
♦ Reporting of test suite execution.

Due date:
Week 8

- **Assessment task 3**

Title:
Automated Unit Testing

Description:
Students will devise and document a unit testing strategy for a provided software artifact. A test suite, using mocking frameworks to simulate other parts of the system where appropriate, will be devised according to the testing strategy and implemented using an automated unit testing framework. Continuous integration will be set up so that the automated testing takes place as the software code base evolves. Metrics for the comprehensiveness of the test suite and the quality of the codebase will be calculated as part of the CI process, and students will analyse the results of these metrics.

Weighting:
15%

Criteria for assessment:

♦ Reasonableness of test strategy
♦ Conformance of test suite to test strategy
♦ Performance of test suite
♦ delivery of continuous integration
♦ calculation of metrics
♦ quality of analysis of metrics

Due date:
Week 12

- **Assessment task 4**

Title:
Tutorial Participation

Description:
Tutorials run from Week 2 through to Week 12. As part of the tutorial process, students will be asked to present analysis of provided readings, and provide short answers to tutorial exercises. Presentations and some tutorial exercises will be marked.

Weighting:
10%

Criteria for assessment:

♦ Correctness of information presented

♦Quality of presentation
♦Correctness and comprehensiveness of marked exercises.

Due date:
Weekly (starts in Week 2)

Examinations

• **Examination 1**

Weighting:
50%
Length:
2 hours
Type (open/closed book):
Open book
Electronic devices allowed in the exam:
None

Learning resources

Reading list

Extensive written course notes will be provided through Moodle.

A large collection of readings will also be made available. Some will be compulsory and this will be indicated on Moodle.

Monash Library Unit Reading List (if applicable to the unit)
http://readinglists.lib.monash.edu/index.html

Feedback to you

Types of feedback you can expect to receive in this unit are:

• Informal feedback on progress in labs/tutes
• Graded assignments without comments
• Solutions to tutes, labs and assignments

Extensions and penalties

Submission must be made by the due date otherwise penalties will be enforced.

You must negotiate any extensions formally with your campus unit leader via the in-semester special consideration process: http://www.monash.edu.au/exams/special-consideration.html

Returning assignments

Students can expect assignments to be returned within two weeks of the submission date or after receipt, whichever is later.

Assignment submission

It is a University requirement (http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-academic-integrity-managing-pla for students to submit an assignment coversheet for each assessment item. Faculty Assignment coversheets can be found at http://www.infotech.monash.edu.au/resources/student/forms/. Please check with your Lecturer on the submission method for your assignment coversheet (e.g. attach a file to the online assignment submission, hand-in a hard copy, or use an electronic submission). Please note that it is your responsibility to retain copies of your assessments.

Online submission

If Electronic Submission has been approved for your unit, please submit your work via the learning system for this unit, which you can access via links in the my.monash portal.

Required Resources

Please check with your lecturer before purchasing any Required Resources. Limited copies of prescribed texts are available for you to borrow in the library, and prescribed software is available in student labs.

The MUSE Lab in Bldg 26/G13 is the lab used for this unit. However, students can install the required software on their personal computers to minimise interruptions to development and testing. The required software includes:

- A working Python installation.
- A git client.

Recommended Resources

Students are encouraged to use their own personal computers to work on project assignments. All required software can be downloaded and installed onto personal computers.

Other Information

Policies

Monash has educational policies, procedures and guidelines, which are designed to ensure that staff and students are aware of the University's academic standards, and to provide advice on how they might uphold them. You can find Monash's Education Policies at:
www.policy.monash.edu.au/policy-bank/academic/education/index.html

Faculty resources and policies

Important student resources including Faculty policies are located at
http://intranet.monash.edu.au/infotech/resources/students/

Graduate Attributes Policy

http://www.policy.monash.edu/policy-bank/academic/education/management/monash-graduate-attributes-policy.h

Student Charter

www.opq.monash.edu.au/ep/student-charter/monash-university-student-charter.html

Student services

The University provides many different kinds of support services for you. Contact your tutor if you need advice and see the range of services available at http://www.monash.edu.au/students. For Malaysia see http://www.monash.edu.my/Student-services, and for South Africa see http://www.monash.ac.za/current/.

Monash University Library

The Monash University Library provides a range of services, resources and programs that enable you to save time and be more effective in your learning and research. Go to www.lib.monash.edu.au or the library tab in my.monash portal for more information. At Malaysia, visit the Library and Learning Commons at http://www.lib.monash.edu.my/. At South Africa visit http://www.lib.monash.ac.za/.

Disability Liaison Unit

Students who have a disability or medical condition are welcome to contact the Disability Liaison Unit to discuss academic support services. Disability Liaison Officers (DLOs) visit all Victorian campuses on a regular basis.

- Website: http://www.monash.edu/equity-diversity/disability/index.html
- Telephone: 03 9905 5704 to book an appointment with a DLO; or contact the Student Advisor, Student Commuity Services at 03 55146018 at Malaysia
- Email: dlu@monash.edu
- Drop In: Equity and Diversity Centre, Level 1, Building 55, Clayton Campus, or Student Community Services Department, Level 2, Building 2, Monash University, Malaysia Campus

Other

Engineers Australia Stage 1 competencies

This unit is a core unit in the Bachelor of Software Engineering accredited by Engineers Australia. Engineers Australia Accreditation Policy of Professional Engineering Programs requires that programs demonstrate how engineering graduates are prepared for entry to the profession and achieve Stage 1 competencies. The following information describes how this unit contributes to the development of these competencies for the Bachelor of Software Engineering. (Note: not all competencies may be emphasised in this unit).

Stage 1 competency	How the compency is developed in this unit
1. Knowledge and Skills base	
1.1. **Comprehension, theory based understanding** of the underpinning natural and physical sciences and the engineering fundamentals applicable to the engineering discipline.	Theoretical lecture materials, prescribed texts and recommended reading, tutorials.
1.2. **Conceptual understanding** of the mathematics, numerical analysis, statistics, and computer and information sciences, which underpin the engineering discipline.	Lectures and tutorial exercises on discrete mathematics relevant to software testing.
1.3. **In-depth understanding** of specialist bodies of knowledge within the engineering discipline.	Lectures on testing techniques and methodologies, tutorials, and projects.
1.4. **Discernment** of knowledge development and research directions within th engineering discipline.	Lectures on testing techniques and methodologies, tutorials, and projects.
1.5. **Knowledge** of engineering design practice and contextual factors impacting the engineering discipline.	Lectures, tutorials, and projects involve the consideration of techniques in the contexts of SE project types.
1.6. **Understanding** of the scope, principles, norms, accountabilities and bounds of sustainable engineering practice in the specific discipline.	Limited discussion in lectures of responsibilities of testers.
2. Engineering application ability	
2.1. **Application** of established engineering methods to complex engineering problem solving.	Lectures, labs and tutorials teach standard software testing tecniques and processes.
2.2 **Fluent application** of engineering techniques, tools and resources.	As well as discussed in 2.1, projects involve the use of standard testing and CI tools.
2.3. **Application** of systematic engineering synthesis and design processes.	Design is not a focus of the unit.
2.4. **Application** of systematic approaches to the conduct and management of engineering projects.	Management is not a focus of the unit.
3. Professional and personal attributes	
3.1. **Ethical** conduct and professional accountability.	Responsibilities of testers are discussed briefly in lectures, but not a major focus of the unit.
3.2. **Effective** oral and written communication in professional and lay domains.	Testability of requirements and reporting is covered in

		assignments and lectures but is not a major focus of the unit.
3.3.	**Creative**, innovative and proactive demeanour.	Not covered in this unit.
3.4.	**Professional** use and management of information.	Not relevant to this unit.
3.5.	**Orderly** management of self, and professional conduct.	Not covered in this unit.
3.6.	**Effective** team membership and team leadership.	Not covered in this unit.

Relationship between Unit Learning Outcomes and BSE Course Outcomes

No.	CO 1	CO 2	CO 3	CO 4	CO 5	CO 6	CO 7	C0 8	CO 9	CO 10	CO 11	CO 12	CO 13
1		X				X	X	X					
2			X	X									
3			X	X		X							
4	X		X	X	X								
5	X		X	X	X								
6	X		X	X	X								
7	X		X	X									

Relationship between Unit Learning Outcomes and Assessments

No.	Assignments	Tests	Practical Exercises	Exam
1	X		X	
2	X		X	X
3	X		X	X
4	X		X	X
5	X		X	X
6	X		X	X
7	X		X	