MONASH University
Information Technology

FIT4004
System validation and verification, quality and standards

Unit Guide

Semester 1, 2011

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: 01 Mar 2011

Table of Contents

FIT4004 System validation and verification, quality and standards - Semester 1, 2011

This unit covers the products, processes, techniques and tools for system validation and verifications including acceptance tests. Commercial Testing Tools from Rational, HP and other 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)

Contact Hours

2 hrs lectures/wk

Workload

Estimated weekly commitment needed for the unit, including classes, reading, assessment, time needed for computer access, and other activities:

- two-hour lecture
- one-hour tutorial
- one-hour unsupervised lab/tute activity in the MUSE Lab to get familiarised with tools, work on assignments, self study, etc.
- a minimum of 2-3 hours of personal study per one hour of contact time in order to satisfy the reading and assignment expectations.

- You will need to allocate up to 5 hours per week in some weeks, for use of a computer, including reading research papers for an assignment and lab discussions with class members.

Unit Relationships

Prohibitions

CSE4431

Prerequisites

FIT2004, FIT2024, FIT3042, FIT3077 and one of FIT2002 or FIT3086
or students must be enrolled in FIT Masters program at Monash

Chief Examiner

Yuan-Fang Li

Campus Lecturer

Clayton

Yuan-Fang Li

Tutors

Clayton

Nabeel Mohammed

Learning Objectives

At the completion of this unit students will have -
A knowledge and understanding of:

- the role of validation and verification methods in the system life cycle;
- key issues in software testing, testing levels and testing activities;
- testing techniques - based on testers experience - adhoc testing, exploratory testing - specification-based - equivalence partioning, boundary-value analysis, finite-state machine based, random testing - code-based - control-flow and data-flow technique - fault-based - error seeding, mutation testing - usage-based - reliability measures, operational profile - based on type of apps - GUI, web based, OO, component testing, testing concurrent/distributed/real-time/embedded systems -selection and combination of techniques;
- test related measures - evaluation of software under test - fault density, types of faults - evaluation of tests done - criteria such as coverage, thoroughness; mutation score;
- empirical work, replication experiments vs case study.

Developed attitudes that enable them to:

- adhere to software quality engineering principles;
- recognise the importance of adhering to software engineering principles of Validation and Verification and standards in the design and development of test methods;
- have an understanding of inspection and debugging approaches, configuration management, performance, and quality standards issues.

Developed the skills to:

- use open source IDEs such as Eclipse and unit testing with JUnit, and coverage tools such as djUnit and Cobertura, commercial validation tools such as from IBM/Rational, and other similar products to help detect software system defects;
- conduct continuous integration process for the application at unit, integration & system testing level with access to SVN, Hudson Continuous Integration (CI) server etc;

- appreciate how assertion mechanisms impact reasoning;
- be able to analyse and control defects in complex systems.

Graduate Attributes

Monash prepares its graduates to be:

1. responsible and effective global citizens who:

a. engage in an internationalised world
b. exhibit cross-cultural competence
c. demonstrate ethical values

critical and creative scholars who:

a. produce innovative solutions to problems
b. apply research skills to a range of challenges
c. communicate perceptively and effectively

Assessment Summary

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

Assessment Task	Value	Due Date
Unit, Integration, System and Continuous testing - Phase 1	20%	Week 6
Unit, Integration, System and Continuous testing - Phase 2	15%	Week 9
Unit, Integration, System and Continuous testing - Phase 3	15%	Week 12
Examination 1	50%	To be advised

Teaching Approach

Lecture and tutorials or problem classes

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

Feedback

Our feedback to You

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

- Informal feedback on progress in labs/tutes
- Solutions to tutes, labs and assignments

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 SETU, Student Evaluation of Teacher and Unit. 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, and on student evaluations, see:
http://www.monash.edu.au/about/monash-directions/directions.html
http://www.policy.monash.edu/policy-bank/academic/education/quality/student-evaluation-policy.html

Previous Student Evaluations of this unit

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

Required Resources

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

The MUSE Lab in Bldg 26/G13 is the lab used for this unit. It has all the software available in standard student labs and is also equipped with:

- Tools for Software testing such as JUnit 4.x (latest vers)
- Java build management Apache Maven 2.x
- Tools for version control, continuous testing and integration such as Hudson and Subversion to run on Windows machines
- Open source Eclipse or Commercial Java IDE IntelliJ IDEA (free site licence available)
- Additional software may be installed in a particular year based on the assignment requirement - such as AspectJ in 2007

Software may be:

- Downloaded from:
    - http://www.eclipse.org/downloads/
    - http://www.eclipse.org/aspectj/
    - http://www.jetbrains.com/idea/download/
    - http://tortoisesvn.net/downloads.html
    - http://maven.apache.org/download.html
- Purchased at academic prices at good software retailers.

Unit Schedule

Week	Date*	Activities	Assessment
0	21/02/11		No formal assessment or activities are undertaken in week 0
1	28/02/11	Overview, testing fundamentals	
2	07/03/11	Mathematics for software testing & quality: set theory, graph theory, etc.	
3	14/03/11	Black-box testing	
4	21/03/11	White-box testing I	
5	28/03/11	White-box testing II	
6	04/04/11	Component testing	Unit, Integration, System and Continuous testing - Phase 1 due Week 6
7	11/04/11	Software quality & metrics	
8	18/04/11	System testing	
Mid semester break			
9	02/05/11	Object-oriented testing	Unit, Integration, System and Continuous testing - Phase 2 due Week 9
10	09/05/11	Mutation testing	
11	16/05/11	Testing vs model checking vs theorem proving	
12	23/05/11	Revision	Unit, Integration, System and Continuous testing - Phase 3 due Week 12
	30/05/11	SWOT VAC	No formal assessment is undertaken SWOT VAC

*Please note that these dates may only apply to Australian campuses of Monash University. Off-shore students need to check the dates with their unit leader.

Assessment Policy

To pass a unit which includes an examination as part of the assessment a student must obtain:

- 40% or more in the unit's examination, and
- 40% or more in the unit's total non-examination assessment, and
- an overall unit mark of 50% or more.

If a student does not achieve 40% or more in the unit examination or the unit non-examination total assessment, and the total mark for the unit is greater than 50% then a mark of no greater than 49-N will be recorded for the unit

Assessment Tasks

Participation

- ### Assessment task 1

Title:

Unit, Integration, System and Continuous testing - Phase 1

Description:

Students work on a small project with the tools specified, produce a report of their findings, and submit the files and report for assessment online on Blackboard. They are also required to submit a hardcopy of the report to the school office collection area for assignments. This is prior to demonstrating the testing done for this assignment in the MUSE Lab. During the demonstration of their work, they explain their understanding and answer queries from the lecturer/tutor.

Weighting:

20%

Criteria for assessment:

Details will be provided.

Due date:

Week 6

Remarks:

Hurdles marks from the tutes are taken into account in the calculation of final marks for the unit – As per the University/Faculty rule, hurdles are linked to the learning outcomes of lecture material, associated reading & assignments, and students must pass all assessed components. Hurdle grades of “Pass”, “Fail” are allocated for weekly tutes. A graded assessment is done around week 10-12 in the tute which together with the hurdle grade must not be < 40% for a Pass in that component. (Read the Faculty/University policy regarding assessment rules).

- ### Assessment task 2

Title:

Unit, Integration, System and Continuous testing - Phase 2

Description:

No written or file submission is required for this assessment. It will be based only on a demo in the lab and answering queries during an interview.

During the assessment interview:

- ♦ Students are required to demonstrate the functionality of the specified testing tool.
- ♦ Students are required to use it on an existing system and focus on regression testing and GUI testing.
- ♦ Students should demonstrate their understanding of automating GUI testing. They should discuss the steps in the test method, the GUI, test cases, test results and exception reports.

Weighting:

15%

Criteria for assessment:

Details will be provided.

Due date:

Week 9

Remarks:

Hurdles marks from the tutes are taken into account in the calculation of final marks for the unit – As per the University/Faculty rule, hurdles are linked to the learning outcomes of lecture material, associated reading & assignments, and students must pass all assessed components. Hurdle grades of “Pass”, “Fail” are allocated for weekly tutes. A graded assessment is done around week 10-12 in the tute which together with the hurdle grade must not be < 40% for a Pass in that component. (Read the Faculty/University policy regarding assessment rules).

- **Assessment task 3**

Title:

Unit, Integration, System and Continuous testing - Phase 3

Description:

A paper which must include an abstract, an overview of the paper, motivation, literature review, students' contribution, related work, weakness of the techniques discussed, further work and a summary/conclusion.

More details will be provided.

Weighting:

15%

Criteria for assessment:

More details will be provided.

Due date:

Week 12

Remarks:

Hurdles marks from the tutes are taken into account in the calculation of final marks for the unit – As per the University/Faculty rule, hurdles are linked to the learning outcomes of lecture material, associated reading & assignments, and students must pass all assessed components. Hurdle grades of “Pass”, “Fail” are allocated for weekly tutes. A graded assessment is done around week 10-12 in the tute which together with the hurdle grade must not be < 40% for a Pass in that component. (Read the Faculty/University policy regarding assessment rules).

Examinations

- **Examination 1**

Weighting:

50%

Length:

2 hours

Type (open/closed book):

Closed book

Electronic devices allowed in the exam:

None

Assignment submission

Assignment coversheets are available via "Student Forms" on the Faculty website:
http://www.infotech.monash.edu.au/resources/student/forms/
You MUST submit a completed coversheet with all assignments, ensuring that the plagiarism declaration section is signed.

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.infotech.monash.edu.au/resources/student/equity/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

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: http://policy.monash.edu.au/policy-bank/academic/education/index.html

Key educational policies include:

- Plagiarism (http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-policy.html)
- Assessment (http://www.policy.monash.edu/policy-bank/academic/education/assessment/assessment-in-coursework-po
- Special Consideration (http://www.policy.monash.edu/policy-bank/academic/education/assessment/special-consideration-policy.h
- Grading Scale (http://www.policy.monash.edu/policy-bank/academic/education/assessment/grading-scale-policy.html)
- Discipline: Student Policy (http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-discipline-policy.html)
- Academic Calendar and Semesters (http://www.monash.edu.au/students/key-dates/);
- Orientation and Transition (http://www.infotech.monash.edu.au/resources/student/orientation/); and
- Academic and Administrative Complaints and Grievances Policy (http://www.policy.monash.edu/policy-bank/academic/education/management/complaints-grievance-policy.

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 www.monash.edu.au/students The Monash University Library provides a range of services and resources that enable you to save time and be more effective in your learning and research. Go to http://www.lib.monash.edu.au or the library tab in my.monash portal for more information. 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://adm.monash.edu/sss/equity-diversity/disability-liaison/index.html;
- Telephone: 03 9905 5704 to book an appointment with a DLO;
- Email: dlu@monash.edu

- Drop In: Equity and Diversity Centre, Level 1 Gallery Building (Building 55), Monash University, Clayton Campus.

Reading List

- Jorgensen, Paul C. (2008), Software Testing, A Craftsman's Approach, 3rd edition, Auerbach Publications.
- M Pezze and M Young (2007), Software Testing and Analysis, Wiley Publ.
- Apt, K.R and Olderog, E.R (1991) Verification of Sequential and Concurrent Programs, Springer-Verlag.
- Dahl, O-J (1992) Verifiable Programming, Prentice Hall.
- Deutsch, M.S (1982) Software Verification and Validation, Prentice Hall
- Dorfman, M and Thayer, R.H (eds) (1990) Standards, Guidelines and Examples on Systems and Software Requirement Engineering, IEEE Computer Soc. Press
- Ferdinand A.E (1993) Systems, Software, and Quality Engineering, Van Nostrand Reinhold. IEEE Standard for Software Quality Metrics Methodology, IEEE Publ. 1993
- Lewis, R.O (1992) Independent Verification and Validation - A Life Cycle Engineering Process for Quality Software, John Wiley & Sons
- Mazz, C.Et al. (1994) Software Engineering Standards, Prentice Hall
- J F Peters and W Pedrycz (2000) Software Engineering: An Engineering Approach, J Wiley Publ
- Robert V. Binder (1999) Testing Object-Oriented Systems: Models, Patterns, and Tools, Addison-Wesley
- David A Sykes John D McGregor (2001) Practical Guide to Testing Object-Oriented Software, Addison-Wesley
- Paul Jorgensen (Ed.) (2002), Software Testing: A Craftsman's Approach, Second Edition
- Daniel J.Mosley, Bruce A. Posey (2002) Just Enough Software Test Automation, Addison-Wesley
- Jerry Gao, H S Tsao and Ye Wu (2003), Testing and Quality Assurance for Component-based Software, Artech House (ISBN 1-58053-480-5)