



MONASH University
Information Technology

FIT5171
System validation and verification, quality and standards

Unit Guide

Semester 1, 2010

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 2010

Table of Contents

<u>FIT5171 System validation and verification, quality and standards - Semester 1, 2010</u>	1
<u>Chief Examiner:</u>	1
<u>Lecturer(s) / Leader(s):</u>	1
<u>Clayton</u>	1
<u>Additional communication information:</u>	1
<u>Introduction</u>	2
<u>Unit synopsis</u>	2
<u>Learning outcomes</u>	2
<u>Contact hours</u>	2
<u>Further unit information</u>	2
<u>Workload</u>	2
<u>Unit relationships</u>	3
<u>Prerequisites</u>	3
<u>Prohibitions</u>	3
<u>Teaching and learning method</u>	4
<u>Teaching approach</u>	4
<u>Timetable information</u>	4
<u>Tutorial allocation</u>	4
<u>Unit Schedule</u>	4
<u>Improvements to this unit</u>	5
<u>Unit Resources</u>	6
<u>Prescribed text(s) and readings</u>	6
<u>Recommended text(s) and readings</u>	6
<u>Required software and/or hardware</u>	6
<u>Equipment and consumables required or provided</u>	7
<u>Study resources</u>	7
<u>Assessment</u>	8
<u>Overview</u>	8
<u>Faculty assessment policy</u>	8
<u>Assignment tasks</u>	8
<u>Due dates and extensions</u>	9
<u>Late assignment</u>	10
<u>Return dates</u>	10
<u>Appendix</u>	11

FIT5171 System validation and verification, quality and standards - Semester 1, 2010

Chief Examiner:

Dr Sita Ramakrishnan

Senior Lecturer

Phone: +61 3 990 58689

Fax: +61 3 990 55159

Lecturer(s) / Leader(s):

Clayton

Dr Sita Ramakrishnan

Senior Lecturer

Phone: +61 3 990 58689

Fax: +61 3 990 55159

Additional communication information:

Dr Sita Ramakrishnan, Bldg 63, Room 119, Clayton campus

Available for consultation in 119/63 for 1 hour after FIT5171/5171 lecture

sitar@infotech.monash.edu.au

Mr Quazi Mamun, Bldg 63, level 1 p/grad room, Clayton campus

quazi.mamun@infotech.monash.edu.au

Tutor/assessor/marking role- shared between Sita and Mamun for FIT5171/5171 assignments and also for assessing assignments demonstrated in the MUSE lab.

Introduction

Welcome to FIT5171.

In this unit, we will examine system validation and verification, quality and standards. In particular, we will review the important phases of testing and emphasize the significance of each phase when testing different types of software. The course will commence with challenges and taxonomy of testing and examine concepts such as test adequacy, test coverage, data-flow testing, category-partition method, test automation, test generation, test oracles, software standards, test selection, regression testing, test prioritization, mutation testing and model based testing. By the end of the course, students should be able to apply state of the art in testing technology for O-O, component-based, GUI and web based systems. Students will also become familiar with open research issues in testing. They will gain hands-on testing and analysis experience via the assignments.

Unit synopsis

This unit covers the core software engineering disciplines concerned with managing and delivering quality software. Topics include processes, tools and techniques for system validation and verification, including major commercial tools used in industry. It shows how to predict, analyse and control defects in complex software systems. Inspection and testing methodologies, analysis of artefacts, robustness, performance analysis configuration management, quality assurance plan and standards including ISO9000/AS39000, compliance, assessment, certification issues are covered.

Learning outcomes

At the completion of this unit students will:

- have knowledge and understanding of the role of validation & verification methods in the system life cycle;
- have gained practical experience in using commercial validation tools to help detect software system defects;
- appreciate how assertion mechanisms impact reasoning;
- be able to analyse and control defects in complex systems;
- have an understanding of inspection & testing methods, configuration management, performance, and quality standards issues.

Contact hours

2 hrs lectures/wk

Further unit information

plus 1 hr tutorial/wk plus 1 hour unsupervised tute-like activity in the MUSE lab

Workload

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

two-hour lecture, one-hour tutorial and one-hour unsupervised lab/tute activity in the MUSE Lab to familiarise 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

Prerequisites

Recommended knowledge: programming in C, C++ and Java; OOSE, Analysis, Design and Programming; OO Method - UML notation, method and SE process; Project Management.

Prohibitions

CSE4431, FIT4004

Teaching and learning method

Teaching approach

Lectures as specified below from weeks 1 - 13

1 hour tute/week for reinforcing material covered in lectures through discussion & students' participation with preset hurdle exercises and marks allocated for a final assessment in the tute.

Unsupervised lab/tute like activity in the MUSE lab to encourage students to learn the material covered and exercise it in tools available in the lab, and get ready for assignments.

3 assignments.

Timetable information

For information on timetabling for on-campus classes please refer to MUTTS, <http://mutts.monash.edu.au/MUTTS/>

Tutorial allocation

On-campus students should register for tutorials/laboratories using the Allocate+ system: <http://allocate.its.monash.edu.au/>

Unit Schedule

Week	Date*	Topic	Key dates
1	01/03/10	Basic Testing, Concepts, IEEE definition, Role of V& V in SDLC, Challenges in Testing	Sem 1 2010 commences in March 2010
2	08/03/10	Faults, Failures and Errors, Test Oracle problem, Taxonomy of software testing	fri - last day to add a unit to sem 1 enrolment
3	15/03/10	Automated testing: JUnit testing, Automating JUnit, Run tests from Ant, Bugzilla, Rational tools, V Model	
4	22/03/10	Testing Techniques, Documenting tests, Test Adequacy and Test Coverage	
5	29/03/10	Testing Techniques: Data-flow testing, Category-Partition method; Test levels; Testing & Certification	Last day to discontinue sem 1 unit & receive fee refund of tuition fees
Mid semester break			
6	12/04/10	Test case Generation (contd): Data-flow Testing; OO test methods; automated GUI testing; Web testing	Ass 1 40% due Week 6, Students complete their assignment and Demo their work in the MUSE Lab -

			exact schedule for demos will be allocated later.
7	19/04/10	Test Oracles; Test Maturity Model; Software Standards	
8	26/04/10	Test Selection: Usage-based testing techniques; Code instrumentation; Effective test prioritization; Regression Testing	
9	03/05/10	Model Based and Model Driven Testing; Quality; Aspect Testing	Ass.2 20%
10	10/05/10	Mutation Testing	
11	17/05/10	Component Testing	
12	24/05/10	Experimental SE and Software Testing	Ass.3 40%
13	31/05/10	Revision/Presentations	Guest lecture - to be advised

*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.

Improvements to this unit

Monquest evaluation is planned for this unit.

Unit Resources

Prescribed text(s) and readings

There is no one set text for the unit. However students are expected to read widely from the recommended reading list. Recommended books are available in the Hargrave Library and may also be in the bookshop should you wish to purchase your own copies.

Recommended text(s) and readings

- 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)

Required software and/or hardware

MUSE Lab in Bldg 26/G13 is the lab used for this unit. Has all the software in the standard student labs plus is equipped with

- standard image as in other student labs **PLUS**
- Tools for Software testing such as JUnit 4.x (latest vers) and compatible Ant vers
- Tools for Continuous testing and integration such as Apache continuum, Hudson and Subversion to run on windows machines
- open source Eclipse from <http://www.eclipse.org/downloads/djunit> from <http://works.dgic.co.jp/djunit/>
- IBM's Websphere, Rational Architect, RationalTester from IBM under Uni agreement
- 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://works.dgic.co.jp/djunit/>;<http://www-306.ibm.com/software/awdtools/tester/functional/>;<http://www.eclips>
- purchased at academic price at good software retailers

Equipment and consumables required or provided

Bachelor of Software Engineering is offered only for On-campus students. You may use the facilities available in the computing labs. Information about computer use for students is available from the ITS Student Resource Guide in the Monash University Handbook. You will need to allocate up to 4hours per week for use of a computer in the lab for doing assignments.

Study resources

Study resources we will provide for your study are:

For more information for units including FIT5171:

You can access MUSO and Blackboard via the portal (<http://my.monash.edu.au>).

Assessment

Overview

Assignments: 80%; Written report and presentation: 20%

Faculty 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.

Students are required to pass each of the assignments to be assigned a Pass in the unit. There is no formal examination for this unit.

Assignment tasks

Assignment coversheets

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.

Assignment submission and return procedures, and assessment criteria will be specified with each assignment.

• Assignment task 1

Title:

Unit, Integration, System and Continuous testing -Phase 1

Description:

Weighting:

40%

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).

- **Assignment task 2**

Title:

Unit, Integration, System and Continuous testing -Phase 2

Description:

Weighting:

20%

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).

- **Assignment task 3**

Title:

Unit, Integration, System and Continuous testing -Phase 3

Description:

To be detailed later

Weighting:

40%

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 or 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

Due dates and extensions

Please make every effort to submit work by the due dates. It is your responsibility to structure your study program around assignment deadlines, family, work and other commitments. Factors such as normal work pressures, vacations, etc. are not regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an extension is a matter of course.

Students requesting an extension for any assessment during semester (eg. Assignments, tests or presentations) are required to submit a Special Consideration application form (in-semester exam/assessment task), along with original copies of supporting documentation, directly to their lecturer within two working days before the assessment submission deadline. Lecturers will provide specific outcomes directly to students via email within 2 working days. The lecturer reserves the right to refuse late applications.

A copy of the email or other written communication of an extension must be attached to the assignment submission.

Refer to the Faculty Special consideration webpage or further details and to access application forms:
<http://www.infotech.monash.edu.au/resources/student/equity/special-consideration.html>

Late assignment

Assignments received after the due date will be subject to a penalty of 5% per day, including weekends.

Assignments received later than one week after the due date will not normally be accepted.

Return dates

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

Appendix

Please visit the following URL: <http://www.infotech.monash.edu.au/units/appendix.html> for further information about:

- Continuous improvement
- Unit evaluations
- Communication, participation and feedback
- Library access
- Monash University Studies Online (MUSO)
- Plagiarism, cheating and collusion
- Register of counselling about plagiarism
- Non-discriminatory language
- Students with disability
- End of semester special consideration / deferred exams