FIT5171
System validation and verification, quality and standards

Unit guide

Semester 1, 2009

Table of Contents

FIT5171 System validation and verification, quality and standards - Semester 1, 2009

Unit leader :

Dr Sita Ramakrishnan

Lecturer(s) :

Clayton

- Sita Ramakrishnan

Introduction

Welcome to FIT5171, Semester 1 2009.

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 subject covers the products, processes, techniques and tools for system validation & verifications including acceptance tests. Commercial Testing Tools from Rational, HP (former tools such as QTP from Mercury Interactive) and others will be used to apply in practice knowledge learnt about software testing from a theoritical perspective. Inspection and testing methodologies, analysis of artifacts, robustness, performance analysis configuration management, quality assurance plan and standards including ISO9000/AS39000, compliance, assessment, certification issues are covered. It shows how to predict, analyse and control defects in complex software systems. It introduces verification methods such as inductive methods for safety properties. It covers operational semantics for sequential and concurrent programs based on Hoare logic - assertion mechanisms - precondition, postcondition and invariants with a view to systematic test planning and validation.

Learning outcomes

Knowledge and Understanding

Understand the role of validation & verification methods in the system life cycle.

Attitudes, Values and Beliefs

Gain experience in using commercial validation tools from vendors such as IBM/Rational and HP (former tools such as QTP from Mercury Interactive), and other similar products in Open Source community to help detect software system defects. Also gain experience in Performance and Load Testing with testing tools from these

vendors.

- 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.

Workload

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

two-hour lecture and one-hour unsupervised tute-like activity in the MUSE Lab to familiarise with tools, work on assignments, self study etca 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

Before attempting this unit you must have satisfactorily completed CSE2201, CSE2304, CSE3308, BUS2176 and CSE2391, CSE3391, CSE2395 or CSE3395, or equivalent. New codes for the above CSE units are:

CSE2201 - FIT2024

CSE2304 - FIT2004

CSE3308 - FIT3077

BUS2176 - FIT3086

CSE2391/3391 & CSE2395/3395 - FIT3042

CSE4213 - FIT3013

 You should have completed or be studying CSE4213 concurrently.

You should have knowledge of :

Programming in C, C++ and Java OOSE, Analysis, Design & Programming OO Method - UML notation, method and SE process Project Management Unix, Perl

Relationships

FIT5171 is a core unit in the Bachelor of Software Engineering degree program. FIT5171 may be taken as an elective by Masters students.

You should have completed or be studying CSE4213 concurrently. **Check with your Masters course director if you are a Masters student**. You should have knowledge of :

- Programming in C, C++ and Java

Learning outcomes   2

- OOSE, Analysis, Design & Programming
- OO Method - UML notation, method and SE process
- Project Management
- Unix, Perl

Continuous improvement

Monash is committed to 'Excellence in education' (Monash Directions 2025 - http://www.monash.edu.au/about/monash-directions/directions.html) and strives for the highest possible quality in teaching and learning.

To monitor how successful we are in providing quality teaching and learning Monash regularly seeks feedback from students, employers and staff. One of the key formal ways students have to provide feedback is through Unit Evaluation Surveys. The University's Unit Evaluation policy (http://www.policy.monash.edu/policy-bank/academic/education/quality/unit-evaluation-policy.html) requires that every unit offered is evaluated each year. Students are strongly encouraged to complete the surveys as they are an important avenue for students to "have their say". The feedback is anonymous and provides the Faculty with evidence of aspects that students are satisfied and areas for improvement.

Faculties have the option of administering the Unit Evaluation survey online through the my.monash portal or in class. Lecturers will inform students of the method being used for this unit towards the end of the semester.

Student Evaluations

If you wish to view how previous students rated this unit, please go to http://www.monash.edu.au/unit-evaluation-reports/

Improvements to this unit

Monquest evaluation is planned for this unit.

Unit staff - contact details

Unit leader

Dr Sita Ramakrishnan
Senior Lecturer
Phone +61 3 990 58689
Fax +61 3 99031777

Lecturer(s) :

Dr Sita Ramakrishnan
Senior Lecturer
Phone +61 3 990 58689
Fax +61 3 99031777

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

FIT5171 System validation and verification, quality and standards - Semester 1, 2009

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.

Teaching and learning method

Lectures as specified below from weeks 1 - 13

Unsupervised 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.

2 Practical hands-on assignments and a third as a Paper submission.

Timetable information

Lectures timetabled on wednesdays 10 - 12 noon, Bldg 11/H7

commencing 4th March 2009.

Last session on 3rd June 2009.

Easter Break 10th April - 17th April 2009

SWOT Vac 8 June - 10 June 2009.

Exam period - 11 June - 3 July 2009.

Exam Results published - 17 July 2009.

Communication, participation and feedback

Monash aims to provide a learning environment in which students receive a range of ongoing feedback throughout their studies. You will receive feedback on your work and progress in this unit. This may take the form of group feedback, individual feedback, peer feedback, self-comparison, verbal and written feedback, discussions (on line and in class) as well as more formal feedback related to assignment marks and grades. You are encouraged to draw on a variety of feedback to enhance your learning.

It is essential that you take action immediately if you realise that you have a problem that is affecting your study. Semesters are short, so we can help you best if you let us know as soon as problems arise. Regardless of whether the problem is related directly to your progress in the unit, if it is likely to interfere with your progress you should discuss it with your lecturer or a Community Service counsellor as soon as possible.

Unit Schedule

Week	Topic	Key dates
1	Basic Testing, Concepts, IEEE definition, Role of V& V in SDLC, Challenges in	Sem 1 commences

	Testing	2nd March 2009, 1st class for FIT5171 - wed 10-12 noon 4th March Bldg 11/H7
2	Faults, Failures and Errors, Test Oracle problem, Taxonomy of software testing	fri - last day to add a unit to sem 1 enrolment
3	Automated testing: JUnit testing, Automating JUnit, Run tests from Ant, Bugzilla, Rational tools, V Model	
4	Testing Techniques, Documenting tests, Test Adequacy and Test Coverage	
5	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
6	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 in Week 7 - exact schedule for demos will be allocated later.
	Mid semester break	
7	Test Oracles; Test Maturity Model; Software Standards	
8	Test Selection: Usage-based testing techniques; Code instrumentation; Effective test prioritization; Regression Testing	
9	Model Based and Model Driven Testing; Quality; Aspect Testing	Ass.2 20%
10	Mutation Testing	
11	Component Testing	
12	Experimental SE and Software Testing	Ass.3 40%
13	Revision/Presentations	Selected Ass.3 presentations by students

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 int he bookshop should you wish to purchase your own copies.

- 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)
- M Pezze and M Young (2007), Software Testing and Analysis, Wiley Publ.

Relevant journals and conference proceedings are used as reference material. Links for relevant articles are given in the weekly lecture notes.

- Hargrave library has copies of these books.

The Bookshop may have copies as well.

Recommended text(s) and readings

- 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)
- M Pezze and M Young (2007), Software Testing and Analysis, Wiley Publ.

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**
- 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.eclipse.org/a
- 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).

Library access

The Monash University Library site contains details about borrowing rights and catalogue searching. To learn more about the library and the various resources available, please go to http://www.lib.monash.edu.au.

The Educational Library and Media Resources (LMR) is also a very resourceful place to visit at http://www.education.monash.edu.au/library/

Monash University Studies Online (MUSO)

All unit and lecture materials are available through MUSO (Monash University Studies Online). Blackboard is the primary application used to deliver your unit resources. Some units will be piloted in Moodle. If your unit is piloted in Moodle, you will see a link from your Blackboard unit to Moodle (http://moodle.monash.edu.au) and can bookmark this link to access directly. In Moodle, from the Faculty of Information Technology category, click on the link for your unit.

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

Click on the Study and enrolment tab, then Blackboard under the MUSO learning systems.

In order for your Blackboard unit(s) to function correctly, your computer needs to be correctly configured.

For example:

- Blackboard supported browser
- Supported Java runtime environment

For more information, please visit: http://www.monash.edu.au/muso/support/students/downloadables-student.html

You can contact the MUSO Support by phone : (+61 3) 9903 1268

FIT5171 System validation and verification, quality and standards - Semester 1, 2009

For further contact information including operational hours, please visit:
http://www.monash.edu.au/muso/support/students/contact.html

Further information can be obtained from the MUSO support site:
http://www.monash.edu.au/muso/support/index.html

Assessment

Unit assessment policy

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 Task**

  Title : 1

  Description :

  Automated Testing and Analysis using JUnit, coverage & other related tools

  Weighting : 40%

  Criteria for assessment :

  Students work on a small project with the tools specified, produce a report of their findings and submit the files & 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.

  Due date : week 6 submission - week 7 - demo in the MUSE Lab
- **Assignment Task**

  Title : 2

  Description :

  Automated tests for the GUI aspects of software systems
  Weighting : 20%

  Criteria for assessment :

   No written or file submission is required for this assignment. Soley based on demoing in the lab and answering queries during the 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.

Due date : Week 9
- **Assignment Task**

Title : 3

Description :

Research Paper
Weighting : 40%

Criteria for assessment :

The paper 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.
Due date : Week 12

Assignment submission

Assignment 1- electronic submission through Blackboard, hardcopy of the report submitted to the school office and demonstrations / interviews with the lecturer/tutor in the lab.

Assignment 2 - demonstration of the work in the lab

Assignment 3 - electronic submission through Blackboard, hardcopy of the paper submitted to the school office and individual presentation to class from a selection of students.

Assignment coversheets

Must use the standard coversheet from the Faculty of IT for assignment submission - Refer to:

 http://www.infotech.monash.edu.au/resources/student/assignments/

University and Faculty policy on assessment

Due dates and extensions

The due dates for the submission of assignments are given in the previous section. 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 seldom regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an extension is a matter of course.

 The due dates for the submission of assignments are given in the previous section. 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 seldom regarded as appropriate reasons for granting extensions. Students are advised to NOT assume that granting of an

extension is a matter of course.

Requests for extensions must be made to the unit lecturer at your campus at least two days before the due date. You will be asked to forward original medical certificates in cases of illness, and may be asked to provide other forms of documentation where necessary. A copy of the email or other written communication of an extension must be attached to the assignment submission.

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.

Assessment for the unit as a whole is in accordance with the provisions of the Monash University Education Policy at http://www.policy.monash.edu/policy-bank/academic/education/assessment/

We will aim to have assignment results made available to you within two weeks after assignment receipt.

Plagiarism, cheating and collusion

Plagiarism and cheating are regarded as very serious offences. In cases where cheating has been confirmed, students have been severely penalised, from losing all marks for an assignment, to facing disciplinary action at the Faculty level. While we would wish that all our students adhere to sound ethical conduct and honesty, I will ask you to acquaint yourself with Student Rights and Responsibilities (http://www.infotech.monash.edu.au/about/committees-groups/facboard/policies/studrights.html) and the Faculty regulations that apply to students detected cheating as these will be applied in all detected cases.

In this University, cheating means seeking to obtain an unfair advantage in any examination or any other written or practical work to be submitted or completed by a student for assessment. It includes the use, or attempted use, of any means to gain an unfair advantage for any assessable work in the unit, where the means is contrary to the instructions for such work.

When you submit an individual assessment item, such as a program, a report, an essay, assignment or other piece of work, under your name you are understood to be stating that this is your own work. If a submission is identical with, or similar to, someone else's work, an assumption of cheating may arise. If you are planning on working with another student, it is acceptable to undertake research together, and discuss problems, but it is not acceptable to jointly develop or share solutions unless this is specified by your lecturer.

Intentionally providing students with your solutions to assignments is classified as "assisting to cheat" and students who do this may be subject to disciplinary action. You should take reasonable care that your solution is not accidentally or deliberately obtained by other students. For example, do not leave copies of your work in progress on the hard drives of shared computers, and do not show your work to other students. If you believe this may have happened, please be sure to contact your lecturer as soon as possible.

Cheating also includes taking into an examination any material contrary to the regulations, including any bilingual dictionary, whether or not with the intention of using it to obtain an advantage.

Plagiarism involves the false representation of another person's ideas, or findings, as your own by either copying material or paraphrasing without citing sources. It is both professional and ethical to reference clearly the ideas and information that you have used from another writer. If the source is not identified, then you have plagiarised work of the other author. Plagiarism is a form of dishonesty that is insulting to the reader and grossly unfair to your student colleagues.

Register of counselling about plagiarism

The university requires faculties to keep a simple and confidential register to record counselling to students about plagiarism (e.g. warnings). The register is accessible to Associate Deans Teaching (or nominees) and, where requested, students concerned have access to their own details in the register. The register is to serve as a record of counselling about the nature of plagiarism, not as a record of allegations; and no provision of appeals in relation to the register is necessary or applicable.

Non-discriminatory language

The Faculty of Information Technology is committed to the use of non-discriminatory language in all forms of communication. Discriminatory language is that which refers in abusive terms to gender, race, age, sexual orientation, citizenship or nationality, ethnic or language background, physical or mental ability, or political or religious views, or which stereotypes groups in an adverse manner. This is not meant to preclude or inhibit legitimate academic debate on any issue; however, the language used in such debate should be non-discriminatory and sensitive to these matters. It is important to avoid the use of discriminatory language in your communications and written work. The most common form of discriminatory language in academic work tends to be in the area of gender inclusiveness. You are, therefore, requested to check for this and to ensure your work and communications are non-discriminatory in all respects.

Students with disabilities

Students with disabilities that may disadvantage them in assessment should seek advice from one of the following before completing assessment tasks and examinations:

- Faculty of Information Technology Student Service staff, and / or
- your Unit Coordinator, or
- Disabilities Liaison Unit

Deferred assessment and special consideration

Deferred assessment (not to be confused with an extension for submission of an assignment) may be granted in cases of extenuating personal circumstances such as serious personal illness or bereavement. Information and forms for Special Consideration and deferred assessment applications are available at http://www.monash.edu.au/exams/special-consideration.html. Contact the Faculty's Student Services staff at your campus for further information and advice.