FIT2024
Software engineering practice

Unit Guide

Semester 1, 2011

Last updated: 27 Feb 2011

Table of Contents

FIT2024 Software engineering practice - Semester 1, 2011

This unit develops students understanding of and skills in professional Software Engineering practices at the personal level. Students experience work in a team environment and extend their programming skills by learning a new object oriented language and maintaining a system that is larger than their experience in prior units. Students develop skills in estimating, monitoring, reviewing and reporting on practical projects.

Mode of Delivery

- Clayton (Day)
- Sunway (Day)

Contact Hours

2 hrs lectures/wk, 2 hrs laboratories/wk

Workload

Workload commitments are:

- two-hour lecture and
- two-hour tutorial (or laboratory) (requiring advance preparation)
- 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 time for discussion groups.

Unit Relationships

Prohibitions

CSE2201, GCO3811, FIT3037

Prerequisites

FIT1007 or FIT1008 or FIT1015 or CSE1303 or CSE1203 or BUS2011 or FIT2034

Chief Examiner

David Squire

Campus Lecturer

Clayton

David Squire

Learning Objectives

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

- the Personal Software Process and its benefits, including the need for planning, estimation, recording time, product and defect metrics, reviews, and reflection;
- the importance of, and the relationship between, a quality process and a quality product;
- reinforceing and extending their knowledge of OO programming concepts by learning how they are implemented in another programming language;
- the Software Engineers role in software development and maintenance and working with large systems;
- the Team Software Process and how it relates to the Personal Software Process.

Developed attitudes that enable them to:

- develop a positive professional attitude;
- recognise the importance of adhering to software engineering principles in designing and implementing systems;

Developed the skills to:

- make personal estimates and work plans, produce work logs and diaries, produce product and defect metrics, and participate in technical review meetings;
- monitor, reflect upon, and improve their own productivity and effectiveness;
- use a new object oriented programming language to construct systems consisting of many interacting classes;
- analyse, debug and perform maintenance on large existing object-oriented programs.

Demonstrated the communication skills necessary to:

- be able to produce reports on their plans, progress, and reviews;
- be able to work effectively in small teams, and cooperatively with other teams.

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): 40%; In-semester assessment: 60%

Assessment Task	Value	Due Date
Individual Exercises	10%	Each exercise is due one week after the tutorial in which it is set.
Team Project	10 + 10 + 15 + 15 = 50%	See Assignment 2 description for due dates
Examination 1	40%	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
- Graded assignments with comments
- Interviews

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

You will need access to:

- **EiffelStudio** - Available for MS Windows, Linux, and Mac OS X
- **TortoiseSVN (for MS Windows only)** - or any other Subversion client
- A tool for creating UML diagrams, such as Visual Paradigm

Links to sites where these tools can be downloaded for free will be provided on the unit website. On-campus students may use this software which is installed in the computing labs. Information about computer use for students is available from the ITS Student Resource Guide in the Monash University Handbook.

Unit Schedule

Week	Date*	Activities	Assessment
0	21/02/11		No formal assessment or activities are undertaken in week 0
1	28/02/11	Introduction to Eiffel	
2	07/03/11	SE & Quality; Eiffel	Tutorials commence
3	14/03/11	SE Process, PSP; Eiffel	Exercise 1 due
4	21/03/11	Specification; Configuration Management	Exercise 2 due
5	28/03/11	Project & Quality Management; Testing;	Exercise 3 due; Hand out Assignment
6	04/04/11	Writing Test Plans and Test Reports	Exercise 4 and PIP due
7	11/04/11	Software Metrics	Stage 1-Planning due
8	18/04/11	Software Maintenance; Software Design	Stage 1-Implementation due
Mid semester break			
9	02/05/11	SQA & Reviews (PSP2)	
10	09/05/11	Software Reuse	Stage 2-Planning due
11	16/05/11	Software Performance	
12	23/05/11	SE Tools	Stage 2-Implementation due
	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

Students must attend at least nine of the eleven tutorials in order to pass the unit. Assessment will take place in almost all tutorials.

- **Assessment task 1**

    Title:
    Individual Exercises
    Description:
    Four (4) programming exercises will be set in weeks 2 through 5. Each of these four exercises is due at the following week's tutorial, where it will be briefly assessed. It will not be possible to obtain marks for work presented after this time. The exercises are cumulative, so it is vital that students keep up with the schedule.

    Students must attend tutorials to submit your work and receive feedback. The exercises are designed to improve object-oriented programming and testing skills, and knowledge of Eiffel, prior to the commencement of the major assignment work. The Personal Software Process (PSP) will be introduced as part of these exercises, and a Process Improvement Proposal (PIP) will be due with the final exercise, which is assessed in week 6.
    Weighting:
    10%
    Criteria for assessment:
    Each program will be assessed for correctness with respect to its functional specification, as well as for good programming style and object-oriented design. Each programming exercise is worth 2%. The PIP is also worth 2%.
    Due date:
    Each exercise is due one week after the tutorial in which it is set.

- **Assessment task 2**

    Title:
    Team Project
    Description:
    Teams will be provided with an existing system written in Eiffel. The project has two (2) stages. In the first stage, they will be required to debug the existing code, and extend it to implement some new functionality. In the second stage further requirements will be added, and students will have to design and implement changes to the system to meet

these new requirements. All work must be planned, measured and tested according to the Personal Software Process (PSP). Each stage consists of a software engineering planning and design component, and an implementation component. These components are submitted and marked separately.

♦ Stage 1 - Planning and Design: 10%, Friday, week 7
♦ Stage 1 - Implementation and Testing: 10%, Friday, week 8
♦ Stage 2 - Planning and Design: 15%, Friday, week 10
♦ Stage 2 - Implementation and Testing: 15%, Friday, week 12

Weighting:
10 + 10 + 15 + 15 = 50%

Criteria for assessment:
Plans will be assessed for completeness with respect to the requirements and the PSP. Designs and code will be assessed for completeness with respect to the requirements, quality of object-oriented design, and coding style. All PSP metrics and test documentation will be assessed for completeness, accuracy and conformance to PSP guidelines.

Due date:
See Assignment 2 description for due dates

Examinations

- **Examination 1**

Weighting:
40%
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

The most highly recommended texts for this unit are:

- Pressman, R. S., *Software Engineering: A Practitioner's Approach*, 6th Edition, McGraw-Hill, 2005. Text books are available from the Monash University Book Shops, and slso available in the

Monash library. A later edition may also be used, but will not necessarily match the chapter references given in the lecture slides.
- Meyer, B., Object-Oriented Software Construction, 2nd Edition, Prentice-Hall, 1997.

Other texts you might find useful include:

- Pfleeger, S. L., Software Engineering Theory and Practice, 2nd Edition, Prentice Hall, 2001.
- Sommerville, I., Software Engineering 7th edition, Addison-Wesley, 2004.
- Humphrey, W.S., Introduction to the Personal Software Process, Addison Wesley 1997.
- Humphrey, W.S., A Discipline for Software Engineering, Addison Wesley 1995.
- Humphrey, W.S., Managing the Software Process, Addison Wesley 1990.
- Humphrey, W.S., Lovelace, M., Hoppes, R., Introduction to the Team Software Process, Addison Wesley, 1998.
- Peters, J. S., Pedrycz, W., Software Engineering, An Engineering Approach, J Wiley, 2000.
- Wiener R.S., Introduction to Computer Science with Eiffel, Prentice Hall 1996.

Further Reading:

- Bennett S, Skelton and Lunn, Schaum's Outlines UML, McGraw-Hill, 2001.
- Larman C, Applying UML and patterns, Prentice-Hall, 1999.
- Jacobson I, Booch G and Rumbaug J, The Unified Software Development Process , Addison-Wesley.
- Covey, S.R., The 7 Habits of Highly Effective People, Pocket Books, 1999.
- Brooks, F., The Mythical Man Month, Addison-Wesley, 1995.