

FIT4002
Software engineering industry experience studio project

Unit Guide

Full year, 2015

Copyright © Monash University 2014. All rights reserved. Except as provided in the Copyright Act 1968, this work may not be reproduced in any form without the written permission of the host Faculty and School/Department.

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: 25 Feb 2015

Table of Contents

<u>FIT4002 Software engineering industry experience studio project - Full year, 2015</u>	1
<u>Mode of Delivery</u>	1
<u>Workload Requirements</u>	1
<u>Additional workload requirements</u>	1
<u>Unit Relationships</u>	1
<u>Prohibitions</u>	1
<u>Co-requisites</u>	1
<u>Prerequisites</u>	2
<u>Chief Examiner</u>	2
<u>Campus Lecturer</u>	2
<u>Clayton</u>	2
<u>Your feedback to Us</u>	2
<u>Previous Student Evaluations of this Unit</u>	3
<u>Academic Overview</u>	4
<u>Learning Outcomes</u>	4
<u>Unit Schedule</u>	5
<u>Teaching Approach</u>	5
<u>Assessment Summary</u>	5
<u>Assessment Requirements</u>	7
<u>Assessment Policy</u>	7
<u>Assessment Tasks</u>	7
<u>Participation</u>	7
<u>Learning resources</u>	10
<u>Reading list</u>	10
<u>Feedback to you</u>	10
<u>Extensions and penalties</u>	10
<u>Returning assignments</u>	10
<u>Assignment submission</u>	10
<u>Online submission</u>	11
<u>Required Resources</u>	11
<u>Field trips</u>	11
<u>Other Information</u>	12
<u>Policies</u>	12
<u>Faculty resources and policies</u>	12
<u>Graduate Attributes Policy</u>	12
<u>Student Charter</u>	12
<u>Student services</u>	12
<u>Monash University Library</u>	12
<u>Disability Liaison Unit</u>	12
<u>Other</u>	13
<u>Engineers Australia Stage 1 competencies</u>	13
<u>Relationship between Unit Learning Outcomes and BSE Course Outcomes</u>	15
<u>Relationship between Unit Learning Outcomes and Assessments</u>	15

FIT4002 Software engineering industry experience studio project - Full year, 2015

Students will undertake a large project and work in groups on a software project for a client. The client may be internal to Monash or from the industry or research organisation. In general, projects involve all aspects of the system development lifecycle. Groups are responsible for their own project management, with guidance from a supervisor. Some projects will warrant students working in pairs or individually.

Mode of Delivery

Clayton (Day)

Workload Requirements

Minimum total expected workload equals 12 hours per week comprising:

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

- One 2-hour seminar
- One 2-hour laboratory

(b.) Additional requirements (all students):

- A minimum of 8 hours additional study per week including undertaking all stages of the software lifecycle for the project, preparation of project documentation, preparation for individual and group presentation, software walkthroughs and SWEBOK interviews. Students are also expected to attend fortnightly group meetings with project supervisor, hold regular meetings with client (may be off-campus) and attend regular meetings of the project group.

See also Unit timetable information

Additional workload requirements

For Software Engineering Studio unit, the workload commitments are for 2 semesters of study

Unit Relationships

Prohibitions

CSE4002

Co-requisites

FIT4004

Prerequisites

FIT2002 and FIT3077

Chief Examiner

Dr David Squire

Campus Lecturer

Clayton

David Squire (unit coordinator)

Consultation hours: Matters for unit coordinator may be raised during weekly seminar slot

Carlo Kopp (back-up unit coordinator and project supervisor)

Consultation hours: Matters may be raised during regular meetings

Lachlan Andrew (project supervisor)

Consultation hours: Matters may be raised during regular meetings

Robyn Mcnamara (project supervisor)

Consultation hours: Matters may be raised during regular meetings

Yuan-Fang Li (project supervisor)

Consultation hours: Matters may be raised during regular meetings

Robert Merkel (project supervisor)

Consultation hours: Matters may be raised during regular meetings

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

- A peer assessment tool will be used so that students can get regular and timely feedback on their contributions.

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. select and use appropriate tools, techniques and strategies for managing risks in the context of a software project;
2. select and use appropriate tools, techniques and strategies and managing project resources, including time and personnel;
3. choose and follow a software development methodology that is appropriate to the team, project and client, and justify this methodology;
4. design and implement a software system of a quality acceptable to an external client;
5. elicit requirements from a client and ensure that these are communicated to team members and other stakeholders in an appropriate form;
6. produce internal documentation of a sufficient quality to support project development activities (including specification, analysis, design, testing);
7. communicate effectively with other project stakeholders, including clients, end users, and supervisors;
8. verify systematically that project deliverables meet agreed quality standards;
9. produce external documentation of a sufficient quality to meet the needs of clients, end users, and client-site technical staff.

Unit Schedule

Week	Activities	Assessment
0	This unit covers two semesters, however this unit schedule lists only semester 1 activities (see Moodle site for full year schedule)	No formal assessment or activities are undertaken in week 0
1	Introduction to the unit, descriptions of available projects.	Project preferences must be submitted
2	Leadership and group dynamics	
3	Personality Typing in Software Engineering	Project Management Plan first draft (part of Assessment Task 5)
4	Guest lecture series	
5	Guest lecture series	Initial requirements document (part of Assessment Task 5)
6	Guest lecture series	
7	Student presentations	Individual seminar presentation
8	Student presentations	Individual seminar presentation
9	Student presentations. Design walkthroughs	Individual seminar presentation. Design walkthrough with supervisor
10	Design walkthroughs	Design walkthrough with supervisor.
11	Group presentations	Group presentation of project
12	Group presentations	Group presentation of project
	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

- **Studio teaching**

Studio teaching is a facilitated active, participatory, peer learning approach.

- **Seminars**

Students will listen to presentations from the unit-coordinator, guest speakers and fellow students on topics relevant to the studio project.

Assessment Summary

Assignments: 100%

Assessment Task

Value

Due Date

5%

In seminar slot in semester 1 (Weeks 7 to 9)

Unit Schedule

Individual seminar presentation

Design and Software walkthroughs	20%	One code walk through scheduled each semester (Weeks 9 and/or 10)
Group presentations	20%	One presentation each semester in the seminar slot (Weeks 11 and 12)
Individual SWEBOK interview	5%	Will be scheduled during the examination period (Semester 2)
Process and project documentation	20%	Different due dates throughout the project. Final versions of all documentation will be assessed at the end of the project (see unit schedule and more detailed breakdown on Moodle site).
Software product	30%	Client acceptance sign-off due Semester 2 (Week 11), final project website with all software artifacts due end of Semester 2 (Week 12)

Assessment Requirements

Assessment Policy

Faculty Policy - Unit Assessment Hurdles

(<http://intranet.monash.edu.au/infotech/resources/staff/edgov/policies/assessment-examinations/assessment-hurdles>)

Academic Integrity - Please see resources and tutorials at

<http://www.monash.edu/library/skills/resources/tutorials/academic-integrity/>

Assessment Tasks

Participation

Attendance at all seminar classes is expected.

Students are also expected to attend weekly or fortnightly meetings with project supervisors.

• Assessment task 1

Title:

Individual seminar presentation

Description:

Each student will be required to give a short seminar presentation to the class on a topic from the SWEBOK (topic to be prior approved by the unit coordinator).

Weighting:

5%

Criteria for assessment:

Assessment criteria will be:

- ◆ Amount of research and preparation
- ◆ Understanding of the topic
- ◆ Quality of the oral presentation

Due date:

In seminar slot in semester 1 (Weeks 7 to 9)

• Assessment task 2

Title:

Design and Software walkthroughs

Description:

In first semester, each project group will undertake a design walkthrough with the project supervisor. In second semester, each project group will undertake a software walkthrough with the project supervisor. The unit coordinator and/or the client may also attend.

Weighting:

20%

Criteria for assessment:

There will be one walkthrough each semester (10% each).

For each walkthrough, 5% will be a group mark, 5% an individual mark.

Assessment Requirements

For the group mark, the assessment criteria will be the overall quality of the project group's code as a whole.

For the individual mark, the assessment criteria will be their individual contribution to the project group code, as well as their demonstrated understanding of the code.

Due date:

One code walk through scheduled each semester (Weeks 9 and/or 10)

• Assessment task 3

Title:

Group presentations

Description:

Each project group will give a presentation each semester in the class seminar time slot. In semester 1, it will be presentation on the project and progress to date. In semester 2, the presentation will describe the project as a whole and the final software product.

Weighting:

20%

Criteria for assessment:

There will be one group presentation each semester (10% each).

For each presentation, 5% will be a group mark, 5% an individual mark.

All students in a team will get the same group mark (5%) for the following assessment criteria:

1. Selection and organisation of content
2. Co-ordination of multiple speakers
3. Quality of visual aids
4. Timing

Each student will receive an individual mark (5%) for the following assessment criteria:

1. Quality of presentation (understandability, coherence)
2. Voice (audibility, intonation, variation)
3. Use of language (e.g., vocabulary, appropriate level, use of jargon)
4. Non-verbal communication (e.g., body language, eye contact)

Due date:

One presentation each semester in the seminar slot (Weeks 11 and 12)

• Assessment task 4

Title:

Individual SWEBOK interview

Description:

At the end of second semester, the student will be interviewed on their knowledge and understanding of SWEBOK, and how it relates to their project. The 15-30 minute interview is a formal exam will be with the unit coordinator, and other members of the BSE teaching group.

Weighting:

5%

Criteria for assessment:

Assessment criteria:

Assessment Requirements

- ◆ Knowledge and understanding of the fundamental areas of software engineering (SWEBOK)
- ◆ Ability to relate SWEBOK to their particular project

Due date:

Will be scheduled during the examination period (Semester 2)

• Assessment task 5

Title:

Process and project documentation

Description:

There will be a variety of process and project documentation to be produced during the project, which must be updated as required. See Moodle site for the unit for details.

Weighting:

20%

Criteria for assessment:

Each piece of documentation will be assessed on:

- ◆ Appropriateness of content
- ◆ Technical quality of content
- ◆ Quality of writing and presentation

While there will be a single overall mark out of 20 for this assessment component, the marks each individual team member receives may be adjusted to reflect their individual contribution to the project.

Due date:

Different due dates throughout the project. Final versions of all documentation will be assessed at the end of the project (see unit schedule and more detailed breakdown on Moodle site).

• Assessment task 6

Title:

Software product

Description:

The final software deliverable for the project.

Weighting:

30%

Criteria for assessment:

The overall software deliverable for the project will be assessed on:

- ◆ Scope of functionality
- ◆ Performance on acceptance testing
- ◆ Quality of software artifacts (e.g. reusability, maintainability)

While there will be a single overall mark out of 30 for this assessment component, the marks each individual team member receives may be adjusted to reflect their individual contribution to the project, and peer assessment.

Due date:

Client acceptance sign-off due Semester 2 (Week 11), final project website with all software artifacts due end of Semester 2 (Week 12)

Learning resources

Reading list

- Relevant Journal Articles and Conference Proceedings depending on the project chosen.
- Gilb T and Graham D, Software inspection, Addison-Wesley, 1993
- Stiller, Project-based Software Engineering, Prentice-Hall, 2001
- Humphrey W, Managing the software process, Addison-Wesley, 1989
- Pfleeger S.L., Software Engineering Theory and Practice, Prentice Hall 2001
- Somerville I.S., Software Engineering Addison Wesley 2001
- Sallis P, Tate G and MacDonell S, Software Engineering: Practice, Management, Improvement, Addison-Wesley, 1995
- Humphrey W, Introduction to the Personal Software Process, Addison Wesley 2000
- Pressman R.S., Software Engineering, A Practitioner's approach, Fifth Ed., McGraw Hill, 2001
- Maciaszek, Requirements Analysis and System Design: Developing Information Systems with UML 2001, Prentice-Hall, 2001

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
- Interviews
- Other: Verbal feedback on progress from supervisor in fortnightly meetings; written comments on drafts of project documentation; written feedback on first SWEBOK interview; marking guide on group presentations; verbal feedback from supervisor and client during software walkthrough

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

Assessment Requirements

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.

Customised Software Engineering laboratory (the MUSE lab) at Clayton with the standard lab image plus high end software engineering & testing tools from IBM/Rational, Websphere software from IBM, Testing tools from Compuware. Open source tools such as Eclipse, Junit & coverage testing tools.

Field trips

May require visit to project client.

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 Community 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 competency 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.

In order to be able to design and architect their software, students must have a deep understanding of the principles of modularity and dependency management that are fundamental to software engineering.

1.2. **Conceptual understanding** of the mathematics, numerical analysis, statistics, and computer and information sciences, which underpin the engineering discipline.

Students must devise their own software solutions based on client requirements, which is likely to require strong skills in algorithmic reasoning. This is a core computer science skill. Proper risk management will also require students to have a working knowledge of probability.

1.3. **In-depth understanding** of specialist bodies of knowledge within the engineering discipline.

The studio project requires deep understanding of Software Engineering, at a minimum, and supporting the needs of some clients may involve exposure to other bodies of knowledge. Students are also interviewed by a panel of supervisors and required to explain how particular facets of their project and its development relate to the Software Engineering Body of Knowledge.

1.4. **Discernment** of knowledge development and research directions within the engineering discipline.

Students are expected to select their own technology stack, and this is usually done with little or no staff involvement. It is the students' responsibility to keep abreast of emerging technologies and products that can support their project, and to evaluate their suitability.

1.5. **Knowledge** of engineering design practice and contextual factors impacting the engineering discipline.

Projects must be conducted in a manner acceptable to the client, including interactions with a variety of different stakeholders at the client site. Thus, students must adapt their processes to the context of development. Moreover, the students must take the domain of the project into account in order to interpret client requirements.

1.6. **Understanding** of the scope, principles, norms, accountabilities and bounds of sustainable engineering practice in the specific discipline.

Students must negotiate their responsibilities with both their client and their supervisor, and are ultimately accountable to both. They must provide an acceptable product to the client, and must also satisfy their supervisor that they have applied good software engineering and project management practices by regularly reporting progress and by providing internal documentation.

2. Engineering application ability

2.1. **Application** of established engineering methods to complex

Supervisors monitor the analysis, design, and development practices of the students to ensure that good software

Other Information

engineering problem solving.

2.2 **Fluent application** of engineering techniques, tools and resources.

2.3. **Application** of systematic engineering synthesis and design processes.

2.4. **Application** of systematic approaches to the conduct and management of engineering projects.

3. Professional and personal attributes

3.1. **Ethical** conduct and professional accountability.

3.2. **Effective** oral and written communication in professional and lay domains.

3.3. **Creative**, innovative and proactive demeanour.

3.4. **Professional** use and management of information.

3.5. **Orderly** management of self, and professional conduct.

3.6. **Effective** team membership and team leadership.

engineering methods are being followed.

Students may choose their own toolset, within the parameters set by their client, but will need to be able to use it fluently in order to get the product built, tested, and deployed in time.

Students are expected to apply the analytical and design principles, techniques that they have learned during their undergraduate courses.

Students manage their own projects, including task allocation, time and resource tracking, progress monitoring, and reporting. They are expected to do this in accordance with principles instilled in earlier units and to a level sufficient to satisfy their satisfy.

Students are accountable to their team leaders, clients, and supervisors, all of whom should be monitoring their behaviour to ensure professionalism and ethical conduct.

Teams must communicate orally on a weekly basis with their clients, supervisors, and teammates, and must also carry out several presentations to their other classmates over the course of the academic year. These are assessed for their content and quality. Teams must also provide their clients with any manuals or documentation required, and must deliver several pieces of internal documentation to their supervisors. These artifacts are used to monitor students' progress and to ensure that proper software engineering practices are being applied, including documentation practices.

Projects are selected on the basis that they will not be trivial to implement. Therefore, students will need to apply creativity to solve novel problems. Supervisors do not play a strong role in task allocation and monitoring, as this is the responsibility of the teams themselves, and usually of the team leader. Therefore, students who are not proactive in seeking tasks to perform are unlikely to achieve a successful outcome.

Most projects will require teams to source domain-specific information from the client or elsewhere. This information needs to be managed in accordance with any relevant legislation and ethical principles as well as being made available efficiently to team members who require access in order to carry out their duties.

Students are expected to conduct themselves in a professional manner at all times when working with teammates, supervisors, or clients. They are responsible for managing their own time and resources.

All studio projects are conducted in self-managing teams. Projects are of sufficient size and complexity that it will be necessary for team members to work together effectively in order to achieve successful completion. One member of each team is appointed to act as team leader (with a minimum term of one semester).

Relationship between Unit Learning Outcomes and BSE Course Outcomes

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

Relationship between Unit Learning Outcomes and Assessments

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