



MONASH University
Information Technology

FIT3077
Software engineering: architecture and design

Unit Guide

Semester 1, 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: 26 Feb 2015

Table of Contents

<u>FIT3077 Software engineering: architecture and design - Semester 1, 2015</u>	1
<u>Mode of Delivery</u>	1
<u>Workload Requirements</u>	1
<u>Unit Relationships</u>	1
<u>Prohibitions</u>	1
<u>Prerequisites</u>	1
<u>Chief Examiner</u>	1
<u>Campus Lecturer</u>	2
<u>Clayton</u>	2
<u>Malaysia</u>	2
<u>Tutors</u>	2
<u>Clayton</u>	2
<u>Your feedback to Us</u>	2
<u>Previous Student Evaluations of this Unit</u>	2
<u>Academic Overview</u>	3
<u>Learning Outcomes</u>	3
<u>Unit Schedule</u>	4
<u>Teaching Approach</u>	4
<u>Assessment Summary</u>	4
<u>Assessment Requirements</u>	6
<u>Assessment Policy</u>	6
<u>Assessment Tasks</u>	6
<u>Participation</u>	6
<u>Examinations</u>	7
<u>Examination 1</u>	7
<u>Learning resources</u>	7
<u>Feedback to you</u>	7
<u>Extensions and penalties</u>	7
<u>Returning assignments</u>	7
<u>Assignment submission</u>	8
<u>Online submission</u>	8
<u>Required Resources</u>	8
<u>Recommended text(s)</u>	8
<u>Other Information</u>	9
<u>Policies</u>	9
<u>Faculty resources and policies</u>	9
<u>Graduate Attributes Policy</u>	9
<u>Student Charter</u>	9
<u>Student services</u>	9
<u>Monash University Library</u>	9
<u>Disability Liaison Unit</u>	9
<u>Other</u>	10
<u>Engineers Australia Stage 1 competencies</u>	10
<u>Relationship between Unit Learning Outcomes and BSE Course Outcomes</u>	11
<u>Relationship between Unit Learning Outcomes and Assessments</u>	11

FIT3077 Software engineering: architecture and design - Semester 1, 2015

This unit builds on introductory units to analysis and design. It provides the professional software engineer with advanced knowledge and skills in high-level architectural design, its theoretical foundations, industrial best practice, and relevant application context. In the software life-cycle, software architecture sits between analysis/specification and design/implementation. The field of software architecture has come of age with a thriving research community and numerous high-level models, methods, tools and practices widely used in industry.

Mode of Delivery

- Clayton (Day)
- Malaysia (Day)

Workload Requirements

Minimum total expected workload equals 12 hours per week comprising:

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

- Two hours of lectures
- One 1-hour laboratory

(b.) Additional requirements (all students):

- A minimum of 3 hours of personal study per one hour of contact time in order to satisfy the reading and assignment expectations.

See also Unit timetable information

Unit Relationships

Prohibitions

CSE3308

Prerequisites

FIT2001 and (FIT2004 or FIT2024 or CSE2304)

Chief Examiner

Dr David Squire

Campus Lecturer

Clayton

David Squire

Consultation hours: Consultation hours will be announced on the Moodle website. Making an appointment via email is strongly advised.

Robyn Mcnamara

Malaysia

Nik Nailah Abdullah

Tutors

Clayton

Robyn Mcnamara

Andre Oboler

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

Based on student feedback:

More detailed instructions will be provided on how to generate and use code for accessing web services.

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

At the completion of this unit students should be able to:

1. model and design flexible software at the architectural level using various tools and techniques;
2. analyse and design software systems taking into consideration various quality attributes and requirements;
3. take requirements for simple systems and develop software architectures and designs at a high level;
4. use configuration management tools effectively;
5. apply a variety of design patterns, frameworks and architectures in designing software;
6. locate and use off-the-shelf components in the construction of software.

Unit Schedule

Week	Activities	Assessment
0		No formal assessment or activities are undertaken in week 0
1	Introduction to FIT3077; What is Software Architecture?; Object-Oriented Analysis using UML	
2	Object-Oriented Analysis using UML	Assignment 1 specification available
3	Principles of Object-Oriented Analysis and Design; Design Patterns	
4	Principles of Object-Oriented Design	Assignment 1 due Friday
5	Principles of Object-Oriented Design	Assignment 2 (Stage 1) specification available
6	Design Principles and Design Patterns	
7	Analysis Patterns; Refactoring	
8	Software Architecture; Architectural Structures	
9	Documenting Software Architectures; The Model-View-Controller Architectural Pattern	Assignment 2 (Stage 1) due Monday. Demonstrations of Stage 1 solutions in lab time
10	Architecture and Design with COTS components	
11	Software Product Lines: Re-using Architectural Assets	
12	Service Orientation; Service-Oriented Architecture	Assignment 2 (Stage 2) due Monday. Final interviews in lab time
	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

Lecture and tutorials or problem classes

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

Assessment Summary

Examination (3 hours): 40%; In-semester assessment: 60%

Assessment Task	Value	Due Date
Assignment 1 - UML Design Assignment	20%	Friday Week 4
	40%	

Unit Schedule

Assignment 2 - Team Software
Architecture Assignment

Stage 1 due Monday Week 9, demonstration in lab. Stage
2 due Monday Week 12, interview in lab

Examination 1

40% To be advised

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

- **Assessment task 1**

Title:

Assignment 1 - UML Design Assignment

Description:

Students will be provided with software requirements scenarios for which they must produce a conceptual design for an object-oriented solution. This design must be described using UML diagrams. The purpose of this assignment is to reinforce UML knowledge and ensure that students are ready for the next assignment.

Weighting:

20%

Criteria for assessment:

Designs will be assessed for completeness with respect to the specification provided, correct use of UML notation, and design quality.

Due date:

Friday Week 4

- **Assessment task 2**

Title:

Assignment 2 - Team Software Architecture Assignment

Description:

This is a team assignment. Students will work on it in pairs. This assignment will have two stages, each requiring students to analyse a problem, design a solution, and implement it in an object-oriented programming language. The second stage will build on the first, and the ease with which the students' stage one design can be extended for stage two will depend on appropriate use of architecture and design knowledge presented in this unit.

The main focus of the assignment is on the application of the design patterns and principles presented in lectures. Simply meeting the external functional requirements for the system is not sufficient to pass this assignment.

Weighting:

40%

Criteria for assessment:

Teams will be assessed on the basis of their submitted design documentation and code, as well as an interview. Both team members will be required to demonstrate knowledge and understanding of all parts of their design.

Assessment Requirements

Designs and code will be assessed for quality, including extensibility and appropriate use of design patterns. Completeness of code functionality with respect to the problem requirements will also be assessed.

Individual contributions to the group assignment will be assessed during the interview.

Due date:

Stage 1 due Monday Week 9, demonstration in lab. Stage 2 due Monday Week 12, interview in lab

Examinations

- **Examination 1**

Weighting:

40%

Length:

3 hours

Type (open/closed book):

Closed book

Electronic devices allowed in the exam:

None

Learning resources

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
- Graded assignments with comments
- Interviews

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 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.

You will need access to:

- A tool for creating UML diagrams, such as Rational Rose, Visual Paradigm, etc.
- An object-oriented programming language, with the capacity to create a graphical user interface.

On-campus students may use the software that 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.

Recommended text(s)

Fowler M.. (2003). *UML Distilled*. (3rd Edition) Addison-Wesley.

Gamma E., Helm R., Johnson R., Vlissides J. M.. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. (1st Edition) Addison-Wesley.

Fowler M.. (1996). *Analysis Patterns: Reusable Object Models*. (1st Edition) Addison-Wesley.

Fowler M., Beck K., Brant J., Opdyke W., Roberts D.. (1999). *Refactoring: Improving the Design of Existing Code*. (1st Edition) Addison-Wesley.

Bass L., Clements P. and Kazman R.. (2003). *Software Architecture in Practice*. () Addison-Wesley.

Allen, P and Schlamann, H.. (2006). *Service Orientation: Winning Strategies and Best Practices*. () Cambridge University Press.

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.	N/A
1.2. Conceptual understanding of the mathematics, numerical analysis, statistics, and computer and information sciences, which underpin the engineering discipline.	N/A
1.3. In-depth understanding of specialist bodies of knowledge within the engineering discipline.	Application of design patterns and design principles in a major software project
1.4. Discernment of knowledge development and research directions within the engineering discipline.	N/A
1.5. Knowledge of engineering design practice and contextual factors impacting the engineering discipline.	Application of design patterns and design principles in a major software project, including location and use of off-the-shelf software components
1.6. Understanding of the scope, principles, norms, accountabilities and bounds of sustainable engineering practice in the specific discipline.	N/A
2. Engineering application ability	
2.1. Application of established engineering methods to complex engineering problem solving.	Application of established design patterns, design principles, and architectural patterns in a major software project
2.2 Fluent application of engineering techniques, tools and resources.	Application of OO design patterns and principles, use of integrated development environment, use of version control systems, use of off-the-shelf components, all in a major software project.
2.3. Application of systematic engineering synthesis and design processes.	Application of established design patterns, design principles, and architectural patterns in a major software project
2.4. Application of systematic approaches to the conduct and management of engineering projects.	Iterative development of a large software project with new requirements added

Other Information

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.

part-way through.

N/A

Communication of design in graphical models and interviews

Opportunity to gain bonus marks through innovative extensions to the system developed for the major assignment.

Participation in a team for the major assignment.

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	X	X	X								
2		X	X										
3		X	X	X	X		X		X	X	X		X
4			X							X			
5	X	X	X	X	X								X
6			X		X		X						X

Relationship between Unit Learning Outcomes and Assessments

No.	Assignments	Quizzes	Practical Exercises	Exam
1	X		X	X
2	X			X
3	X			
4	X			
5	X			X
6	X			