



COIT20258 Software Engineering

Term 2 - 2020

Profile information current as at 02/05/2024 11:00 pm

All details in this unit profile for COIT20258 have been officially approved by CQUniversity and represent a learning partnership between the University and you (our student). The information will not be changed unless absolutely necessary and any change will be clearly indicated by an approved correction included in the profile.

General Information

Overview

This unit aims to teach you both theoretical and practical aspects of software engineering which involves specification, development, management and evolution of software systems. The theory will focus on software processes, requirements engineering, system models and modelling, robustness analysis, architectural design, object-oriented design, and software development methodology. The theoretical and practical aspects of software testing, software evolution, and quality management including quality assurance, and quality control are also discussed. You will also be introduced to the principles of software evolution and configuration management. This unit has a strong practical focus and the theoretical aspects are reinforced with practical work. The software engineering life cycle support functionality provided by modern software development environments is also addressed.

Details

Career Level: *Postgraduate*

Unit Level: *Level 9*

Credit Points: 6

Student Contribution Band: 8

Fraction of Full-Time Student Load: 0.125

Pre-requisites or Co-requisites

Pre-requisites: COIT20247 Database Design and Development, COIT20256 Data Structures and Algorithms, and

COIT20248 Information Systems Analysis and Design Anti-requisites: COIT20226 Software Design and Implementation

Important note: Students enrolled in a subsequent unit who failed their pre-requisite unit, should drop the subsequent unit before the census date or within 10 working days of Fail grade notification. Students who do not drop the unit in this timeframe cannot later drop the unit without academic and financial liability. See details in the [Assessment Policy and Procedure \(Higher Education Coursework\)](#).

Offerings For Term 2 - 2020

- Brisbane
- Melbourne
- Online
- Rockhampton
- Sydney

Attendance Requirements

All on-campus students are expected to attend scheduled classes – in some units, these classes are identified as a mandatory (pass/fail) component and attendance is compulsory. International students, on a student visa, must maintain a full time study load and meet both attendance and academic progress requirements in each study period (satisfactory attendance for International students is defined as maintaining at least an 80% attendance record).

Website

[This unit has a website, within the Moodle system, which is available two weeks before the start of term. It is important that you visit your Moodle site throughout the term. Please visit Moodle for more information.](#)

Class and Assessment Overview

Recommended Student Time Commitment

Each 6-credit Postgraduate unit at CQUniversity requires an overall time commitment of an average of 12.5 hours of study per week, making a total of 150 hours for the unit.

Class Timetable

[Regional Campuses](#)

Bundaberg, Cairns, Emerald, Gladstone, Mackay, Rockhampton, Townsville

[Metropolitan Campuses](#)

Adelaide, Brisbane, Melbourne, Perth, Sydney

Assessment Overview

1. **Practical Assessment**

Weighting: 20%

2. **Practical Assessment**

Weighting: 30%

3. **Online Test**

Weighting: 50%

Assessment Grading

This is a graded unit: your overall grade will be calculated from the marks or grades for each assessment task, based on the relative weightings shown in the table above. You must obtain an overall mark for the unit of at least 50%, or an overall grade of 'pass' in order to pass the unit. If any 'pass/fail' tasks are shown in the table above they must also be completed successfully ('pass' grade). You must also meet any minimum mark requirements specified for a particular assessment task, as detailed in the 'assessment task' section (note that in some instances, the minimum mark for a task may be greater than 50%). Consult the [University's Grades and Results Policy](#) for more details of interim results and final grades.

CQUniversity Policies

All University policies are available on the [CQUniversity Policy site](#).

You may wish to view these policies:

- Grades and Results Policy
- Assessment Policy and Procedure (Higher Education Coursework)
- Review of Grade Procedure
- Student Academic Integrity Policy and Procedure
- Monitoring Academic Progress (MAP) Policy and Procedure – Domestic Students
- Monitoring Academic Progress (MAP) Policy and Procedure – International Students
- Student Refund and Credit Balance Policy and Procedure
- Student Feedback – Compliments and Complaints Policy and Procedure
- Information and Communications Technology Acceptable Use Policy and Procedure

This list is not an exhaustive list of all University policies. The full list of University policies are available on the [CQUniversity Policy site](#).

Previous Student Feedback

Feedback, Recommendations and Responses

Every unit is reviewed for enhancement each year. At the most recent review, the following staff and student feedback items were identified and recommendations were made.

Feedback from University policy.

Feedback

Where appropriate, replace exams in non-foundational units with more practical, authentic assessments.

Recommendation

Replace the final exam with an applied project which is more appropriate for building software engineering skills.

Feedback from Have your Say: student evaluation.

Feedback

Include more interactive sessions to discuss theoretical aspects.

Recommendation

Introduce a workshop (reducing the lecture time) where the theoretical aspects introduced in the lecture can be discussed in smaller groups.

Unit Learning Outcomes

On successful completion of this unit, you will be able to:

1. Apply the principles of software design
2. Propose appropriate software management processes for a software engineering project
3. Employ Unified Modeling Language (UML) and design patterns in the design and documentation of software systems
4. Critique design alternatives and development methods available to a software developer
5. Design, develop, and test non-distributed, three layered software applications
6. Exploit the software engineering life cycle support functionality provided by modern software development environments.

The Australian Computer Society (ACS) recognises the Skills Framework for the Information Age (SFIA). SFIA is in use in over 100 countries and provides a widely used and consistent definition of ICT skills. SFIA is increasingly being used when developing job descriptions and role profiles.

ACS members can use the tool MySFIA to build a skills profile at

<https://www.acs.org.au/professionalrecognition/mysfia-b2c.html>

This unit contributes to the following workplace skills as defined by SFIA:

- Systems Design (DESN)
- System Integration (SINT)
- Programming/Software Development (PROG)
- Data Analysis (DTAN)
- Database/Repository Design (DBDS)
- Testing (TEST)
- Applications Support (ASUP)

Alignment of Learning Outcomes, Assessment and Graduate Attributes



Alignment of Assessment Tasks to Learning Outcomes

Assessment Tasks	Learning Outcomes					
	1	2	3	4	5	6
1 - Practical Assessment - 20%			•		•	•
2 - Practical Assessment - 30%	•		•		•	•
3 - Online Test - 50%	•	•		•		

Alignment of Graduate Attributes to Learning Outcomes

Graduate Attributes	Learning Outcomes					
	1	2	3	4	5	6
1 - Knowledge	○	○	○	○	○	○
2 - Communication	○	○		○	○	
3 - Cognitive, technical and creative skills	○	○	○	○	○	○
4 - Research						
5 - Self-management						
6 - Ethical and Professional Responsibility					○	
7 - Leadership						
8 - Aboriginal and Torres Strait Islander Cultures						

Alignment of Assessment Tasks to Graduate Attributes

Assessment Tasks	Graduate Attributes							
	1	2	3	4	5	6	7	8
1 - Practical Assessment - 20%	○	○	○		○	○		
2 - Practical Assessment - 30%	○	○	○		○	○		
3 - Online Test - 50%	○	○	○		○	○		

Textbooks and Resources

Textbooks

COIT20258

Prescribed

Software Engineering

Edition: 10th

Authors: Sommerville, I.

Pearson Higher Ed

ISBN: 1-292-09613-6

Binding: Paperback

Additional Textbook Information

If you prefer to study with a paper copy, they are available at the CQUni Bookshop here: <http://bookshop.cqu.edu.au> (search on the Unit code). eBooks are available at the publisher's website.

[View textbooks at the CQUniversity Bookshop](#)

IT Resources

You will need access to the following IT resources:

- CQUniversity Student Email
- Internet
- Unit Website (Moodle)
- Java 8
- NetBeans 8.2
- Apache Derby 10.14.2.0
- Scene Builder 8.5.0

Referencing Style

All submissions for this unit must use the referencing style: [Harvard \(author-date\)](#)
For further information, see the Assessment Tasks.

Teaching Contacts

Dennis Jarvis Unit Coordinator

d.jarvis@cqu.edu.au

Mary Tom Unit Coordinator

m.tom@cqu.edu.au

Steven Gordon Unit Coordinator

s.d.gordon@cqu.edu.au

Schedule

Week 1 - 13 Jul 2020

Module/Topic	Chapter	Events and Submissions/Topic
Introduction; Software Processes.	Chapter 1; Chapter 2.	

Week 2 - 20 Jul 2020

Module/Topic	Chapter	Events and Submissions/Topic
Agile Software Development; Requirements Engineering; Case Study 1 - AddressBook: Initial Implementation.	Chapter 3; Chapter 4; Online Material.	

Week 3 - 27 Jul 2020

Module/Topic	Chapter	Events and Submissions/Topic
System Modelling; Robustness Analysis; Case Study 1 - AddressBook: First Refactoring	Chapter 5; Online Material.	

Week 4 - 03 Aug 2020

Module/Topic	Chapter	Events and Submissions/Topic
Architectural Design; Design of Layered Systems.	Chapter 6; Online Material.	

Week 5 - 10 Aug 2020

Module/Topic	Chapter	Events and Submissions/Topic
Design and Implementation (Part 1); Design Documents.	Chapter 7; Online Material.	

Vacation Week - 17 Aug 2020

Module/Topic	Chapter	Events and Submissions/Topic
--------------	---------	------------------------------

Week 6 - 24 Aug 2020

Module/Topic	Chapter	Events and Submissions/Topic
OOD Principles; Case Study 1 - AddressBook: Second Refactoring.	Online Material.	

Week 7 - 31 Aug 2020

Module/Topic	Chapter	Events and Submissions/Topic
Design and Implementation (Part 2); Unit Testing.	Chapter 7; Online Material.	Assignment 1 Due: Week 7 Monday (31 Aug 2020) 11:55 pm AEST

Week 8 - 07 Sep 2020

Module/Topic	Chapter	Events and Submissions/Topic
Software Testing; Case Study 2 - Wumpus: Project Brief	Chapter 8; Online Material.	

Week 9 - 14 Sep 2020

Module/Topic	Chapter	Events and Submissions/Topic
Software Evolution; Refactoring; Case Study 1 - Wumpus: Design (Part 1).	Chapter 9; Online Material.	

Week 10 - 21 Sep 2020

Module/Topic	Chapter	Events and Submissions/Topic
Software Reuse; Component-Based Software Engineering.	Chapter 15; Chapter 16.	

Week 11 - 28 Sep 2020

Module/Topic	Chapter	Events and Submissions/Topic
Quality Management; Case Study 2 - Wumpus: Design (Part 2)	Chapter 24; Online Material.	Assignment 2 Due: Week 11 Monday (28 Sep 2020) 11:55 pm AEST

Week 12 - 05 Oct 2020

Module/Topic	Chapter	Events and Submissions/Topic
Configuration Management; Review	Chapter 25; Online Material.	

Review/Exam Week - 12 Oct 2020

Module/Topic	Chapter	Events and Submissions/Topic
--------------	---------	------------------------------

Exam Week - 19 Oct 2020

Module/Topic	Chapter	Events and Submissions/Topic
--------------	---------	------------------------------

Assessment Tasks

1 Assignment 1

Assessment Type

Practical Assessment

Task Description

Students will use Robustness Analysis in the design of a three-layered Java application. The requirements for the application will be provided on the unit web site.

Assessment Due Date

Week 7 Monday (31 Aug 2020) 11:55 pm AEST

Return Date to Students

Two weeks after the due date.

Weighting

20%

Assessment Criteria

Criteria	Marks Available
Use case specifications	1.5
Robustness Analysis	8
Screens (one per use case)	3
Data access class (specification)	1.5
Sequence Diagrams (one per use case)	3
Report	3
Total	20

Referencing Style

- [Harvard \(author-date\)](#)

Submission

Online

Learning Outcomes Assessed

- Employ Unified Modeling Language (UML) and design patterns in the design and documentation of software systems
- Design, develop, and test non-distributed, three layered software applications
- Exploit the software engineering life cycle support functionality provided by modern software development environments.

Graduate Attributes

- Knowledge
- Communication
- Cognitive, technical and creative skills
- Self-management
- Ethical and Professional Responsibility

2 Assignment 2

Assessment Type

Practical Assessment

Task Description

Students will implement a three-layered Java application that conforms to a specification that will be provided on the unit web site.

Assessment Due Date

Week 11 Monday (28 Sept 2020) 11:55 pm AEST

Return Date to Students

Two weeks after the due date.

Weighting

30%

Assessment Criteria

Criteria	Marks Available
Functionality	6
Conformance to specified layer interactions	9
Conformance to coding guidelines	5
Testing	5
Source code documentation	5
Total	30

Referencing Style

- [Harvard \(author-date\)](#)

Submission

Online

Learning Outcomes Assessed

- Apply the principles of software design
- Employ Unified Modeling Language (UML) and design patterns in the design and documentation of software systems
- Design, develop, and test non-distributed, three layered software applications
- Exploit the software engineering life cycle support functionality provided by modern software development environments.

Graduate Attributes

- Knowledge
- Communication
- Cognitive, technical and creative skills
- Self-management
- Ethical and Professional Responsibility

3 Online Test

Assessment Type

Online Test

Task Description

The online test will take the form of a take home exam

Assessment Due Date

The take home exam will be scheduled in the examination period.

Return Date to Students

Marks will be released on the certification date.

Weighting

50%

Assessment Criteria

No Assessment Criteria

Referencing Style

- [Harvard \(author-date\)](#)

Submission

No submission method provided.

Learning Outcomes Assessed

- Apply the principles of software design
- Propose appropriate software management processes for a software engineering project
- Critique design alternatives and development methods available to a software developer

Graduate Attributes

- Knowledge
- Communication
- Cognitive, technical and creative skills
- Self-management
- Ethical and Professional Responsibility

Academic Integrity Statement

As a CQUniversity student you are expected to act honestly in all aspects of your academic work.

Any assessable work undertaken or submitted for review or assessment must be your own work. Assessable work is any type of work you do to meet the assessment requirements in the unit, including draft work submitted for review and feedback and final work to be assessed.

When you use the ideas, words or data of others in your assessment, you must thoroughly and clearly acknowledge the source of this information by using the correct referencing style for your unit. Using others' work without proper acknowledgement may be considered a form of intellectual dishonesty.

Participating honestly, respectfully, responsibly, and fairly in your university study ensures the CQUniversity qualification you earn will be valued as a true indication of your individual academic achievement and will continue to receive the respect and recognition it deserves.

As a student, you are responsible for reading and following CQUniversity's policies, including the [Student Academic Integrity Policy and Procedure](#). This policy sets out CQUniversity's expectations of you to act with integrity, examples of academic integrity breaches to avoid, the processes used to address alleged breaches of academic integrity, and potential penalties.

What is a breach of academic integrity?

A breach of academic integrity includes but is not limited to plagiarism, self-plagiarism, collusion, cheating, contract cheating, and academic misconduct. The Student Academic Integrity Policy and Procedure defines what these terms mean and gives examples.

Why is academic integrity important?

A breach of academic integrity may result in one or more penalties, including suspension or even expulsion from the University. It can also have negative implications for student visas and future enrolment at CQUniversity or elsewhere. Students who engage in contract cheating also risk being blackmailed by contract cheating services.

Where can I get assistance?

For academic advice and guidance, the [Academic Learning Centre \(ALC\)](#) can support you in becoming confident in completing assessments with integrity and of high standard.

What can you do to act with integrity?

**Be Honest**

If your assessment task is done by someone else, it would be dishonest of you to claim it as your own

**Seek Help**

If you are not sure about how to cite or reference in essays, reports etc, then seek help from your lecturer, the library or the Academic Learning Centre (ALC)

**Produce Original Work**

Originality comes from your ability to read widely, think critically, and apply your gained knowledge to address a question or problem