COMP1050
Introduction to the Study of Software Engineering
Session 1, In person-scheduled-weekday, North Ryde 2024
School of Computing

Contents
General Information 2
Learning Outcomes 2
General Assessment Information 3
Assessment Tasks 3
Delivery and Resources 6
Unit Schedule 7
Policies and Procedures 8
Changes from Previous Offering 10

Disclaimer
Macquarie University has taken all reasonable measures to ensure the information in this publication is accurate and up-to-date. However, the information may change or become out-dated as a result of change in University policies, procedures or rules. The University reserves the right to make changes to any information in this publication without notice. Users of this publication are advised to check the website version of this publication [or the relevant faculty or department] before acting on any information in this publication.
General Information

Unit convenor and teaching staff
Convener and lecturer
Kate Stefanov
kate.stefanov@mq.edu.au

Convener and lecturer
Ansgar Fehnker
ansgar.fehnker@mq.edu.au

Credit points
10

Prerequisites
Admission to BEng

Corequisites

Co-badged status

Unit description
This unit covers the fundamentals of software engineering, including understanding system requirements, finding appropriate engineering compromises, learning software engineering culture, forming camaraderie, understanding basic methods of design, coding, and testing, team software development, and the application of engineering tools.

Important Academic Dates
Information about important academic dates including deadlines for withdrawing from units are available at https://www.mq.edu.au/study/calendar-of-dates

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

ULO1: Work with version control, configuration management, unit/regression testing, issue tracking, and debugging tools.
ULO2: Create a project plan.
ULO3: Create and analyse design models.
ULO4: Make engineering tradeoffs.
ULO5: Demonstrate an understanding of software engineering culture and form camaraderie.
General Assessment Information

The assessment tasks include individual and group written submissions, module exams, workshop and project participation. The project participation is a hurdle assessment.

Requirements to Pass

To pass the unit, you will have to achieve:

- a total mark equal to or greater than 50%, and
- pass the hurdle.

Hurdle Assessment

Working in teams on a substantial software engineering project mostly takes place in the weekly workshops and is a hurdle. You are required to complete no fewer than 8 out of the 12 weekly workshop tasks to pass the hurdle requirement. If you have failed the hurdle, but have made a genuine effort on four weekly tasks, it is possible to remedy that. See section Delivery and Resources for details.

Late Assessment Submission Penalty

Assignment 1, Assignment 2, the interim and final report are to be uploaded by the due date. Late submissions are allowed. Unless a Special Consideration request has been submitted and approved, a 5% penalty (of the total possible mark of the task) will be applied for each day a written report or presentation assessment is not submitted, up until the 7th day (including weekends). After the 7th day, a grade of '0' will be awarded even if the assessment is submitted. The submission time for all uploaded assessments is 11:55 pm. A 1-hour grace period will be provided to students who experience technical issues.

For any late submission of time-sensitive tasks, such as scheduled tests/exams, performance assessments/presentations, and/or scheduled practical assessments/labs, please apply for Special Consideration.

Special Consideration

The Special Consideration Policy aims to support students who have been impacted by short-term circumstances or events that are serious, unavoidable and significantly disruptive, and which may affect their performance in assessment. If you experience circumstances or events that affect your ability to complete the assessments in this unit on time, please inform the convenor and submit a Special Consideration request through ask.mq.edu.au.

Assessment Tasks

<table>
<thead>
<tr>
<th>Name</th>
<th>Weighting</th>
<th>Hurdle</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Exams</td>
<td>20%</td>
<td>No</td>
<td>Week 4 (M1), Week 9 (M2),</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Exam period (M3)</td>
</tr>
<tr>
<td>Name</td>
<td>Weighting</td>
<td>Hurdle</td>
<td>Due</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>-----------</td>
<td>--------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>Working in teams on a substantial software engineering project</td>
<td>40%</td>
<td>Yes</td>
<td>Weekly workshop contribution, and 2 reports (week 6 and 12)</td>
</tr>
<tr>
<td>Assignment 1</td>
<td>20%</td>
<td>No</td>
<td>Week 7</td>
</tr>
<tr>
<td>Assignment 2</td>
<td>20%</td>
<td>No</td>
<td>Week 13</td>
</tr>
</tbody>
</table>

**Module Exams**

Assessment Type ¹: Examination  
Indicative Time on Task ²: 19 hours  
Due: Week 4 (M1), Week 9 (M2), Exam period (M3)  
Weighting: 20%

A number of invigilated examinations of the unit's content, spread through the semester. Students will have the opportunity to repair exam marks below passing mark.

On successful completion you will be able to:

• Work with version control, configuration management, unit/regression testing, issue tracking, and debugging tools.
• Create a project plan.
• Create and analyse design models.
• Make engineering tradeoffs.
• Demonstrate an understanding of software engineering culture and form camaraderie.

**Working in teams on a substantial software engineering project**

Assessment Type ¹: Project  
Indicative Time on Task ²: 0 hours  
Due: Weekly workshop contribution, and 2 reports (week 6 and 12)  
Weighting: 40%  
This is a hurdle assessment task (see assessment policy for more information on hurdle assessment tasks)

Students work in groups to develop week-by-week a software engineering project, both learning the principles and practices of software engineering, and gaining an overview of a wide range of software engineering areas (that are studied in much greater depth in individual units later in their program).
On successful completion you will be able to:

- Work with version control, configuration management, unit/regression testing, issue tracking, and debugging tools.
- Create a project plan.
- Create and analyse design models.
- Make engineering tradeoffs.
- Demonstrate an understanding of software engineering culture and form camaraderie.

Assignment 1

Assessment Type: Problem set
Indicative Time on Task: 20 hours
Due: Week 7
Weighting: 20%

An opportunity to demonstrate the learning achieved in the first half of the unit's lectures

On successful completion you will be able to:

- Work with version control, configuration management, unit/regression testing, issue tracking, and debugging tools.
- Create a project plan.

Assignment 2

Assessment Type: Problem set
Indicative Time on Task: 20 hours
Due: Week 13
Weighting: 20%

An opportunity to demonstrate the learning achieved in the second half of the unit's lectures

On successful completion you will be able to:

- Create and analyse design models.
- Make engineering tradeoffs.
If you need help with your assignment, please contact:

- the academic teaching staff in your unit for guidance in understanding or completing this type of assessment
- the Writing Centre for academic skills support.

Indicative time-on-task is an estimate of the time required for completion of the assessment task and is subject to individual variation.

**Delivery and Resources**

**Week 1 Classes**
Both lectures and workshops start in Week 1.

**Weekly Workshops (includes a hurdle task)**
There are weekly workshops with set tasks and time allocated to working on the assessment task *Working in teams on a substantial software engineering project*. You will need to conscientiously attend the workshops and engage with the work, with your tutor and fellow students. Contribution to the workshop and the group project is a hurdle task meaning that you cannot complete COMP1050 satisfactorily without completing the hurdle task satisfactorily. We expect you to be involved every week, but we know, of course, that things happen, you might be ill, for example. You need to be present and actively engaged in at least eight of the twelve sessions to be eligible to meet the hurdle, and if illness or anything else leads you to miss more than four sessions, you should speak to the convenor.

Contribution is assessed as follows:

- **Weekly contribution to the workshop**: This entails participation and completion of weekly tasks. It contributes for (10%) to your final grade. Your best 10 out of 12 weekly tasks will be used for your grade. **IMPORTANT. This is a hurdle.** If you complete fewer than 8 out of 12 weekly tasks, you will not pass the hurdle requirement. However, if you made a genuine effort on four weekly tasks but still failed to pass the hurdle, you have the option to take a viva examination in the week 13. The viva will cover the exercises completed during the workshops and the software engineering project.
- **Interim project report**: Due in week 6. It contributes for (10%) to your final grade. It describes the first weeks of the software engineering project. This is a collaborative document.
- **Final project report**: Due in week 12. It contributes for (20%) to your final grade. It builds on the interim report and describes the entire software engineering project. This is a collaborative document.

The "0 hours" estimated time required for that task is because the task is to be completed during
your scheduled workshop class (and class hours are recorded separately).

Lectures
This course includes 2-hour weekly lectures. Attendance at these lectures is highly recommended as they introduce the concepts that will be used in the workshops and will be covered in the assessments for Assignment 1, Assignment 2, and the module exams.

Assignments
Assignments play a crucial role in evaluating student understanding. They are based on the lecture material, workshop activities and weekly tutorial material, and require students to integrate what they have learned, think critically and creatively.

Module exams
There are three module exams. All module exams are designed to test student understanding of the course content and the application of the concepts to a number of scenarios or problem statements.

Two of the module exams are held in workshops during the semester, one in Week 4 and another one in Week 9. The third module exam is held during the exam period.

Unit Resources
Although there is no required textbook, we suggest that students read:


This is a valuable resource for understanding the day-to-day tasks of software engineers. The book is available in the library in both online and ebook formats.

The workshop exercises and the software engineering project are outlined in the workshop manual. Additionally, lecture notes and recordings will be made available after each lecture.

Methods of Communication
We will communicate with you via your university email, through announcements on iLearn and in lectures. Queries to convenors can either be placed on the iLearn discussion board or sent to the unit convenor via the contact email on iLearn.

Unit Schedule
Unit guide COMP1050 Introduction to the Study of Software Engineering

<table>
<thead>
<tr>
<th></th>
<th>Degree Introductions + What is Software Engineering?</th>
<th>Forming a team, Discussion and group work</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>SDLC and Software Processes</td>
<td>GitHub project, Markdown</td>
</tr>
<tr>
<td>3</td>
<td>Requirements and epics, Issue Tracking</td>
<td>Team formation, Project assignment</td>
</tr>
<tr>
<td>4</td>
<td>Version Control Systems and Debugging</td>
<td>Git, Processing</td>
</tr>
<tr>
<td>5</td>
<td>Software Testing, Code Review</td>
<td>GitHub, Testing and Review</td>
</tr>
<tr>
<td>6</td>
<td>Software Testing (Unit and regression)</td>
<td>More testing, Coverage</td>
</tr>
<tr>
<td>7</td>
<td>Testing in the SDLC, Project Management</td>
<td>Definition of Done, Project Management</td>
</tr>
<tr>
<td>8</td>
<td>Software Architecture and Design Models</td>
<td>Composition, Modelling</td>
</tr>
<tr>
<td>9</td>
<td>Software modelling and Software Quality</td>
<td>Analysing Processing code, Coupling and Cohesion</td>
</tr>
<tr>
<td>10</td>
<td>High Level Software Architecture</td>
<td>Beta and acceptance testing of project</td>
</tr>
<tr>
<td>11</td>
<td>Software Traceability and Configuration Management</td>
<td>Requirements dependency, Project review</td>
</tr>
<tr>
<td>12</td>
<td>Professionalism and Ethics</td>
<td>Tutorial project presentation, Final report time</td>
</tr>
<tr>
<td>13</td>
<td>Review and project final presentation</td>
<td>Q&amp;A</td>
</tr>
<tr>
<td></td>
<td>Review and project final presentation</td>
<td>Q&amp;A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The University will advise on how these will proceed as we move forward in the Semester.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>More information will be provided as it becomes available</td>
</tr>
</tbody>
</table>

Topics are tentative, subject to adjustments. Workshops start in Week 1.

**Policies and Procedures**

Macquarie University policies and procedures are accessible from Policy Central (https://policies.mq.edu.au). Students should be aware of the following policies in particular with regard to Learning and Teaching:

- Academic Appeals Policy
- Academic Integrity Policy
- Academic Progression Policy
Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: https://students.mq.edu.au/admin/other-resources/student-conduct

Results

Results published on platform other than eStudent, (eg. iLearn, Coursera etc.) or released directly by your Unit Convenor, are not confirmed as they are subject to final approval by the University. Once approved, final results will be sent to your student email address and will be made available in eStudent. For more information visit ask.mq.edu.au or if you are a Global MBA student contact globalmba.support@mq.edu.au

Academic Integrity

At Macquarie, we believe academic integrity – honesty, respect, trust, responsibility, fairness and courage – is at the core of learning, teaching and research. We recognise that meeting the expectations required to complete your assessments can be challenging. So, we offer you a range of resources and services to help you reach your potential, including free online writing and maths support, academic skills development and wellbeing consultations.

Student Support

Macquarie University provides a range of support services for students. For details, visit http://students.mq.edu.au/support/

The Writing Centre

The Writing Centre provides resources to develop your English language proficiency, academic writing, and communication skills.

- Workshops
- Chat with a WriteWISE peer writing leader
- Access StudyWISE
- Upload an assignment to Studioity
- Complete the Academic Integrity Module
The Library provides online and face to face support to help you find and use relevant information resources.

- **Subject and Research Guides**
- **Ask a Librarian**

**Student Services and Support**

Macquarie University offers a range of **Student Support Services** including:

- **IT Support**
- **Accessibility and disability support** with study
- **Mental health support**
- **Safety support** to respond to bullying, harassment, sexual harassment and sexual assault
- **Social support including information about finances, tenancy and legal issues**
- **Student Advocacy** provides independent advice on MQ policies, procedures, and processes

**Student Enquiries**

Got a question? Ask us via AskMQ, or contact Service Connect.

**IT Help**

For help with University computer systems and technology, visit [http://www.mq.edu.au/about_us/offices_and_units/information_technology/help/](http://www.mq.edu.au/about_us/offices_and_units/information_technology/help/).

When using the University’s IT, you must adhere to the **Acceptable Use of IT Resources Policy**.

**Changes from Previous Offering**

**Module exams instead of a final exam**

In response to student feedback the final exam has been replaced by module exams.

**The workshops and the software project.**

While the software engineering content is the same as, the order in which topics are covered, the exercises, and the nature of the project may change.

**Lectures**

While the topics of the lectures remain the same, the order in which they are covered, and the application examples may change.
Repeating Students

Students who retake this unit are expected to fully participate in all learning activities, just as first-time students do. Previous participation in the unit does not exempt you from any requirements. Completing homework or assessments in a previous iteration of the unit will not exempt you from completing the assignments again this time around.

Unit information based on version 2024.02 of the Handbook