COMP255

Software Engineering

S2 Day 2017

Dept of Computing

Contents

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
Convenor and Lecturer
Franck Cassez
franck.cassez@mq.edu.au
Contact via franck.cassez@mq.edu.au
E6A 313
Appointment

Lecturer
Mike Johnson
michael.johnson@mq.edu.au
Contact via michael.johnson@mq.edu.au
E6A383
Appointment

Credit points
3

Prerequisites
(18cp at 100 level or above) including COMP125

Corequisites

Co-badged status

Unit description
This unit introduces engineering principles and practices to all stages of the software development lifecycle to ensure a systematic, quality-focused and quantifiable approach to the management, design, development, maintenance, verification and validation of [large and complex] software products, projects and processes. Problem formulation and solving are emphasised. Topics covered include: requirements gathering and specification; object-oriented modelling using the Unified Modeling Language (UML); process management; and software design, testing and evolution.

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:

Ability to write properly documented code, unit tests, and to use best software engineering practices

Ability to use modern software development techniques and tools (Agile, Version Control)

Knowledge of some requirements elicitation techniques and the ability to apply these techniques to a some problem domains.

Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the development life cycle.

An understanding of Software Engineering including its principles, practices, ethics and responsibilities.

General Assessment Information

A major part of all software engineering projects is being able to report accurately and on time. For all your assignments, and for your professional life in the future, you are encouraged to

- set your personal deadline earlier than the actual one
- keep backups of all your important files
- make sure that no-one else picks up your printouts or has access to your files

Late work will not be accepted. Develop good working habits and manage your time well. If your contributions are seriously affected by illness or misadventure do your utmost to contact the lecturer **before** the due date. If the severity of the disruption warrants it, you should consider applying for special consideration.

Assessment Tasks

Name	Weighting	Hurdle	Due
Weekly problems	10%	No	Weekly
Class Participation	10%	No	Weekly
Assignment 1	20%	No	Friday 15th September, 5pm
Assignment 2	20%	No	Friday 27th October, 5pm
Final Exam	40%	No	Exam Period

Weekly problems

Due: **Weekly**
Weighting: **10%**

Feedback: In lectures

Each week from week 1 to week 12, some questions/problems will be posted in iLearn related to the lecture content for that week.

By the end of each week you are required to submit your responses in the manner specified in the questions for that week. Your answers will need to demonstrate appropriate depth of understanding.

The marks for this task will be computed using the **ten best submissions**. **No late submissions.**

On successful completion you will be able to:
- Ability to write properly documented code, unit tests, and to use best software engineering practices
- Knowledge of some requirements elicitation techniques and the ability to apply these techniques to a some problem domains.
- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the development life cycle.
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.

Class Participation

Due: **Weekly**
Weighting: **10%**

Feedback: In classes

You are expected to actively participate to the weekly worshop. Your active **participation will be marked**. A roll will be taken to record your attendance.

Workshops will involve a range of activities, some individual, some in pairs, some in groups. Be prepared to present your ideas.

On successful completion you will be able to:
- Ability to use modern software development techniques and tools (Agile, Version Control)
- Knowledge of some requirements elicitation techniques and the ability to apply these techniques to a some problem domains.
- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the development life cycle.

- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.

Assignment 1

Due: **Friday 15th September, 5pm**
Weighting: **20%**

Assignment 1 will allow you to demonstrate the development of your understanding and your ability to apply the things that you have learned in the first part (weeks 1 to 6) of the unit.

You will have to produce and submit your deliverables (including programs) via Version Control System (VCS) and BitBucket. How to use the VCS will be covered in the lectures and the workshops.

No late submission.

On successful completion you will be able to:

- Ability to write properly documented code, unit tests, and to use best software engineering practices
- Ability to use modern software development techniques and tools (Agile, Version Control)
- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the development life cycle.
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.

Assignment 2

Due: **Friday 27th October, 5pm**
Weighting: **20%**

Assignment 2 will be a written assignment that will allow you to demonstrate the development of your understanding and your ability to apply the things that you have learned in the second part (weeks 7 to 12) of the unit.

No late submission.

On successful completion you will be able to:

- Ability to write properly documented code, unit tests, and to use best software engineering practices
- Ability to use modern software development techniques and tools (Agile, Version Control)
- Knowledge of some requirements elicitation techniques and the ability to apply these

techniques to a some problem domains.

- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the development life cycle.
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.

Final Exam

Due: **Exam Period**
Weighting: **40%**

The final examination will be a three-hour examination (closed book) held during the usual University examination period and will cover all topics.

On successful completion you will be able to:

- Ability to write properly documented code, unit tests, and to use best software engineering practices
- Knowledge of some requirements elicitation techniques and the ability to apply these techniques to a some problem domains.
- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the development life cycle.
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.

Delivery and Resources

Location

COMP255 is taught via lectures in C5C T1 Theatre (Wednesday 10-11am) and W5A P.G. Price Theatre (Friday 1-3pm). Workshops (mixed classes) happen in various tutorial rooms (E6A119, E6A121).

Textbook

There is no textbook for COMP255. The unit content will be presented in lectures and practical classes. There will also be some assigned readings, and there are many books that can be consulted for reference material.

Lecture Recordings

Digital recordings of parts of lectures are expected to be available. Note that some of the lecture activities are interactive, and because there is no video recording it is unlikely that you will be able to understand what is happening if you were not present for the original lecture. In addition,

some confidential case-studies will be presented in lectures but will not be recorded.

If you miss a lecture you should obtain a copy of the lecture notes from another student, find out whatever else you can about the lecture to catch up, think carefully about the material, and then talk to the lecturer at the next lecture to discuss anything that is still unclear.

Website

Extra course materials, some online discussions, and the submission of assessable work, will be carried out via iLearn.

The unit will make use of discussion boards hosted within iLearn. Feel free to post questions there -- the discussion boards will be monitored by the staff teaching the unit. Everyone is encouraged to engage with the discussion and provide questions and/or answers. Such discussions will generally be about the material treated in comp255, but other aspects of Software Engineering may be discussed too. We are a "community of interest" and we are here to learn about software engineering (including the software engineering taught in comp255, but not only the software engineering taught in comp255!).

If you have a personal question it is usually more appropriate to email the lecturer teaching during that week.

Teaching and Learning Strategies

The feedback that you receive plays a crucial role in your learning. You have many opportunities to seek and receive feedback. During lectures, you are encouraged to ask the lecturer questions to clarify anything you might not be sure of. To ensure you don't miss feedback make sure that you 1) review your marked assignments, 2) read unit emails and information at the unit website and 3) attend lectures which are often used to point out what was expected in assessable work and to provide examples of good solutions and/or examples of common errors.

Lectures are used to motivate engagement with and reinforcement of the unit's subject matter. While lectures will be recorded via iLecture, some activities such as role plays and dramas conducted in the lecture theatre will not be captured properly in the iLecture recording. Volunteers to participate in role plays will be requested in the lecture, please consider whether you can contribute in this way. Lectures will of course include a significant amount of learning material, but even more importantly they include contextual material and learning activities that "make meaning" of the subject matter.

Workshops are smaller group classes which give you the opportunity to interact with your peers and with a tutor who has sound knowledge of the subject. Workshops will require working in small groups and sometimes involve reporting back to the class. The classes will focus on reinforcing understanding of the concepts and their practical applications to problems. It is important that you participate in the activities and make some notes from them to assist you with revision of the material.

Assignments will play a key role in providing formative evaluation so that students and the teachers can gauge levels of understanding. Assignments will be related to the lecture, practical and tutorial material and require students to bring together what they have been learning.

Exams will be designed to test your understanding of the course content and your application of the concepts to a number of scenarios or problem statements.

Each week you should:

- **Attend lectures**, take notes, ask questions
- **Complete the weekly questions** and submit them via ilearn
- **Attend your workshop** and seek feedback from your tutor on your work
- **Read assigned reading material**, add to your notes and prepare questions for your lecturer or tutor
- **Start working on** any **assignments** immediately after they have been released.

Unit Schedule

The topic titles are indicative, and may be subject to change.

Week	Topic	Lecturer
1	Introduction to Software Systems	Franck
2	Software Development Methods (Agile)	Franck
3	Version Control	Franck
4	Good Software Engineering Practices	Franck
5	Software Quality	Franck
6	Embedded and Real-time Systems	Franck
7	Software Architecture and Design	Mike
	Semester break	
8	Distributed Systems	Mike
9	Requirements Elicitation	Mike
10	Modelling Domains and Modelling Systems	Mike
11	Software Maintenance	Mike
12	Introduction to Security	Mike
13	Revision	Mike & Franck

Learning and Teaching Activities

Lectures

Each week you should attend three hours of lectures.

Practical Class

A weekly two-hour class (with tutorial and practical aspects combined in a single session called a workshop).

Reading, reflection and written submissions

Each week you will be required to write and submit a short document responding to set problems that will require some careful analysis and sometimes some reading and careful thought.

Policies and Procedures

Macquarie University policies and procedures are accessible from Policy Central. Students should be aware of the following policies in particular with regard to Learning and Teaching:

Academic Honesty Policy http://mq.edu.au/policy/docs/academic_honesty/policy.html

Assessment Policy http://mq.edu.au/policy/docs/assessment/policy_2016.html

Grade Appeal Policy http://mq.edu.au/policy/docs/gradeappeal/policy.html

Complaint Management Procedure for Students and Members of the Public http://www.mq.edu.au/policy/docs/complaint_management/procedure.html

Disruption to Studies Policy (in effect until Dec 4th, 2017): http://www.mq.edu.au/policy/docs/disruption_studies/policy.html

Special Consideration Policy (in effect from Dec 4th, 2017): https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policies/special-consideration

In addition, a number of other policies can be found in the Learning and Teaching Category of Policy Central.

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: https://students.mq.edu.au/support/student_conduct/

Results

Results shown in *iLearn*, 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.

Student Support

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

dents.mq.edu.au/support/

Learning Skills

Learning Skills (mq.edu.au/learningskills) provides academic writing resources and study strategies to improve your marks and take control of your study.

- Workshops
- StudyWise
- Academic Integrity Module for Students
- Ask a Learning Adviser

Student Services and Support

Students with a disability are encouraged to contact the Disability Service who can provide appropriate help with any issues that arise during their studies.

Student Enquiries

For all student enquiries, visit Student Connect at ask.mq.edu.au

IT Help

For help with University computer systems and technology, visit 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. The policy applies to all who connect to the MQ network including students.

Graduate Capabilities

Creative and Innovative

Our graduates will also be capable of creative thinking and of creating knowledge. They will be imaginative and open to experience and capable of innovation at work and in the community. We want them to be engaged in applying their critical, creative thinking.

This graduate capability is supported by:

Learning outcomes

- Ability to write properly documented code, unit tests, and to use best software engineering practices
- Ability to use modern software development techniques and tools (Agile, Version Control)
- Knowledge of some requirements elicitation techniques and the ability to apply these techniques to a some problem domains.

Assessment tasks

- Class Participation
- Assignment 1
- Assignment 2

Capable of Professional and Personal Judgement and Initiative

We want our graduates to have emotional intelligence and sound interpersonal skills and to demonstrate discernment and common sense in their professional and personal judgement. They will exercise initiative as needed. They will be capable of risk assessment, and be able to handle ambiguity and complexity, enabling them to be adaptable in diverse and changing environments.

This graduate capability is supported by:

Learning outcomes

- Ability to use modern software development techniques and tools (Agile, Version Control)
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.

Assessment tasks

- Weekly problems
- Assignment 1
- Assignment 2

Discipline Specific Knowledge and Skills

Our graduates will take with them the intellectual development, depth and breadth of knowledge, scholarly understanding, and specific subject content in their chosen fields to make them competent and confident in their subject or profession. They will be able to demonstrate, where relevant, professional technical competence and meet professional standards. They will be able to articulate the structure of knowledge of their discipline, be able to adapt discipline-specific knowledge to novel situations, and be able to contribute from their discipline to inter-disciplinary solutions to problems.

This graduate capability is supported by:

Learning outcomes

- Ability to write properly documented code, unit tests, and to use best software engineering practices
- Ability to use modern software development techniques and tools (Agile, Version Control)

- Knowledge of some requirements elicitation techniques and the ability to apply these techniques to a some problem domains.
- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the development life cycle.
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.

Assessment tasks

- Weekly problems
- Class Participation
- Assignment 1
- Assignment 2
- Final Exam

Critical, Analytical and Integrative Thinking

We want our graduates to be capable of reasoning, questioning and analysing, and to integrate and synthesise learning and knowledge from a range of sources and environments; to be able to critique constraints, assumptions and limitations; to be able to think independently and systemically in relation to scholarly activity, in the workplace, and in the world. We want them to have a level of scientific and information technology literacy.

This graduate capability is supported by:

Learning outcomes

- Ability to write properly documented code, unit tests, and to use best software engineering practices
- Ability to use modern software development techniques and tools (Agile, Version Control)
- Knowledge of some requirements elicitation techniques and the ability to apply these techniques to a some problem domains.
- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the development life cycle.

Assessment tasks

- Weekly problems
- Assignment 1
- Assignment 2
- Final Exam

Problem Solving and Research Capability

Our graduates should be capable of researching; of analysing, and interpreting and assessing data and information in various forms; of drawing connections across fields of knowledge; and they should be able to relate their knowledge to complex situations at work or in the world, in order to diagnose and solve problems. We want them to have the confidence to take the initiative in doing so, within an awareness of their own limitations.

This graduate capability is supported by:

Learning outcome

- Ability to write properly documented code, unit tests, and to use best software engineering practices

Assessment tasks

- Weekly problems
- Assignment 1
- Assignment 2
- Final Exam

Effective Communication

We want to develop in our students the ability to communicate and convey their views in forms effective with different audiences. We want our graduates to take with them the capability to read, listen, question, gather and evaluate information resources in a variety of formats, assess, write clearly, speak effectively, and to use visual communication and communication technologies as appropriate.

This graduate capability is supported by:

Learning outcomes

- Ability to write properly documented code, unit tests, and to use best software engineering practices
- Ability to use modern software development techniques and tools (Agile, Version Control)
- Knowledge of some requirements elicitation techniques and the ability to apply these techniques to a some problem domains.
- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the development life cycle.

Assessment tasks

- Weekly problems
- Class Participation

- Assignment 1
- Assignment 2
- Final Exam

Engaged and Ethical Local and Global citizens

As local citizens our graduates will be aware of indigenous perspectives and of the nation's historical context. They will be engaged with the challenges of contemporary society and with knowledge and ideas. We want our graduates to have respect for diversity, to be open-minded, sensitive to others and inclusive, and to be open to other cultures and perspectives: they should have a level of cultural literacy. Our graduates should be aware of disadvantage and social justice, and be willing to participate to help create a wiser and better society.

This graduate capability is supported by:

Learning outcome

- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.

Changes from Previous Offering

The unit comp255 is changing substantially as the software engineering programs at Macquarie are further developed and as we take account of student feedback on previous offerings. While the intended outcomes remain substantially the same, students should not assume that any particular aspects of comp255 this year will be in any particular ways similar to those aspects of previous years' offerings.

Changes for this year (Semester 2, 2017):

- the two parts of the unit are delivered respectively from weeks 1 to 6 and weeks 7 to 12.
- assignments 1 and assignment 2 are independent
- no hurdle requirements

Grading

At the end of the session, you will receive a mark that reflects your achievement in the unit.

The final mark for the unit will be calculated by combining the marks for all assessment tasks according to the percentage weightings shown in the assessment summary.

 No late submissions for the weekly problems and the assignments.