COMP255

Software Engineering

S2 Day 2018

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
Lecturer and Convenor
Michael Johnson
michael.johnson@mq.edu.au
Contact via michael.johnson@mq.edu.au
9WW 383
After each lecture

Lecturer
James Zheng
james.zheng@mq.edu.au
9WW 384
By 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 good 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 those techniques to a range of problem domains

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

If you receive special consideration for the final exam, a supplementary exam will be scheduled in the week of December 17-21 2018. By making a special consideration application for the final exam you are declaring yourself available for a resit during the supplementary examination period and will not be eligible for a second special consideration approval based on pre-existing commitments. Please ensure you are familiar with the policy prior to submitting an application. Approved applicants will receive an individual notification one week prior to the exam with the exact date and time of their supplementary examination.

Finally a remark about **hurdle** assessments:

Students are required to perform satisfactorily to meet the hurdle requirement. If a student makes a serious attempt, but fails to meet the hurdle requirement for a particular item of assessment, then he or she will be offered one further opportunity to meet the hurdle requirement. In this unit we expect students to reach a pass level (50%) in these assessment items, but, recognising that they are designed as an aid to learning (they are formative), the hurdle requirement will be merely 40%. Students obtaining between 30% and 40% on an assessment item will be given a further assessment task to allow them another opportunity to demonstrate the performance required to meet the hurdle.

In other words, in this unit we are define 'satisfactory' as at least 40%, and 'serious attempt' as at

least 30%.

Assessment Tasks

Name	Weighting	Hurdle	Due
Weekly problems	10%	Yes	Weekly
Class Participation	10%	Yes	Weekly
Assignment 1	15%	Yes	Thursday 13th September, 5pm
Assignment 2	15%	Yes	Thursday 25th October, 5pm
Final Exam	50%	No	Exam Period

Weekly problems

Due: **Weekly**

Weighting: **10%**

This is a hurdle assessment task (see assessment policy for more information on hurdle assessment tasks)

Feedback: In lectures

Each week from week 1 to week 12, some questions/problems will be posted in iLearn, usually 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.

No late submissions.

Students are expected to complete all 12 submissions satisfactorily, but must complete at least 8 of 12 to meet the hurdle requirement.

On successful completion you will be able to:

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

responsibilities.

Class Participation

Due: **Weekly**
Weighting: **10%**
This is a hurdle assessment task (see assessment policy for more information on hurdle assessment tasks)

Feedback: In classes

You are expected to actively participate in the weekly workshop. Your active **participation will be marked**. A roll will be taken to record your attendance. You must attend at your assigned weekly workshop time.

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

Students are expected to participate in all 12 workshops satisfactorily, but must participate satisfactorily in at least 8 to meet the hurdle requirement.

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

Assignment 1

Due: **Thursday 13th September, 5pm**
Weighting: **15%**
This is a hurdle assessment task (see assessment policy for more information on hurdle assessment tasks)

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.

No late submission.

On successful completion you will be able to:

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

Assignment 2

Due: **Thursday 25th October, 5pm**
Weighting: **15%**
This is a hurdle assessment task (see assessment policy for more information on hurdle assessment tasks)

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

Final Exam

Due: **Exam Period**
Weighting: **50%**

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 good 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 those techniques to a range of problem domains
- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the life cycle
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.

Delivery and Resources

Location

COMP255 is taught via lectures on Monday from noon to 2pm in Mason Theatre (formerly E7BT1), and on Tuesday from 9am to 10am in 29WWT1 (formerly X5BT1). Workshops (mixed classes) are two hours long and take place in various locations generally on level 1 of 9WW (formerly E6A) at a time assigned to each student individually.

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 most lectures are planned to be recorded, some activities such as role plays and dramas conducted in the lecture theatre will not be captured properly in the 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	Software Architecture and Distributed Systems	Mike
2	Requirements Definition and Requirements Specification	Mike
3	Modelling Domains and Modelling Systems	Mike
4	Software Development Methods, especially Agile	James
5	Version Control	James
6	Good Software Engineering Practices	James
7	Software Quality	James
	Semester break	
8	Embedded Systems	James
9	Real Time Applications	James
10	Design, Implementation and Deployment	Mike
11	Software Maintenance	Mike
12	Software Security Engineering	Mike
13	Revision	Mike & James

Learning and Teaching Activities

Lectures

Each week you need to 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). The workshops are compulsory and some of what you will learn there is not taught elsewhere in the unit (but is examinable).

Reading, reflection and written submissions

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

Assignments

The assignments in this unit exercise your abilities in real-world analysis and design. They are a vitally important part of the unit and involve team work as well as significant amounts of individual work. Each student is required to submit his or her own individual solutions, and they are individually assessed.

Policies and Procedures

Macquarie University policies and procedures are accessible from Policy Central (https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-central). 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
- Assessment Policy
- Fitness to Practice Procedure
- Grade Appeal Policy
- Complaint Management Procedure for Students and Members of the Public
- Special Consideration Policy *(**Note:** The Special Consideration Policy is effective from 4 December 2017 and replaces the Disruption to Studies Policy.)*

Undergraduate students seeking more policy resources can visit the Student Policy Gateway (https://students.mq.edu.au/support/study/student-policy-gateway). It is your one-stop-shop for the key policies you need to know about throughout your undergraduate student journey.

If you would like to see all the policies relevant to Learning and Teaching visit Policy Central (https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/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/study/getting-started/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 good 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 those techniques to a range of 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 good 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 those techniques to a range of problem domains
- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the 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 good 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 those techniques to a range of problem domains
- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the 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 good 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 good 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 those techniques to a range of problem domains
- Awareness of the life cycle of software systems development and the impact of implementation issues on various phases of the 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, 2018):

- version control systems will be included in workshop material
- assignment 1 and assignment 2 are co-dependent -- students will need to participate in assignment 1 in order to complete assignment 2
- hurdle requirements have been reintroduced to help (some) students manage their learning better
- there will be lecture material emphasising the purpose and importance of the unit for students' likely future careers

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.

Changes since First Published

Date	Description
25/07/2018	Additional information about a supplementary examination has been added, including information about its timing.