



COMP255

Software Engineering

S2 Day 2014

Computing

Contents

<u>General Information</u>	2
<u>Learning Outcomes</u>	2
<u>General Assessment Information</u>	3
<u>Assessment Tasks</u>	3
<u>Delivery and Resources</u>	6
<u>Unit Schedule</u>	8
<u>Learning and Teaching Activities</u>	9
<u>Policies and Procedures</u>	9
<u>Graduate Capabilities</u>	11
<u>Changes from Previous Offering</u>	15
<u>Grading</u>	15

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

Teaching Staff

Michael Johnson

michael.johnson@mq.edu.au

Contact via michael.johnson@mq.edu.au

E6A 383

Mondays and Wednesdays from 6pm, C5C T1

Teaching Staff

Steve Cassidy

steve.cassidy@mq.edu.au

Contact via steve.cassidy@mq.edu.au

E6A 377

To be determined

Credit points

3

Prerequisites

18cp including (COMP115(P) or COMP155(P))

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); project and 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:

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

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

Ability to analyse the system requirements, build a logical model of the problem and turn the logical model from the analysis phase into a design model from which a system can be built.

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

Ability to use appropriate software tools to support the requirements, analysis and design phases.

Ability to derive an architecture and design solution from requirements and analysis work products incorporating aspects of quality, testing, and other post-design workflow activities.

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; and
- 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	Due
<u>Weekly problems</u>	10%	Weekly
<u>Class Participation</u>	10%	Weekly
<u>Assignment 1</u>	15%	Friday 12th September, 5pm
<u>Assignment 2</u>	15%	Friday 31st October, 5pm

Name	Weighting	Due
Final Exam	50%	Exam Period

Weekly problems

Due: **Weekly**

Weighting: **10%**

Feedback: In tutorial classes

Each week 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.

On successful completion you will be able to:

- Awareness of the life cycle of system development and the impact of implementation issues on various phases of the development life cycle.
- Knowledge of requirements elicitation techniques and the ability to apply these techniques to a wide range of problem domains.
- Ability to analyse the system requirements, build a logical model of the problem and turn the logical model from the analysis phase into a design model from which a system can be built.
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.
- Ability to use appropriate software tools to support the requirements, analysis and design phases.
- Ability to derive an architecture and design solution from requirements and analysis work products incorporating aspects of quality, testing, and other post-design workflow activities.

Class Participation

Due: **Weekly**

Weighting: **10%**

Feedback: In classes

Your attendance at mixed tutorial/practical classes is expected and your active participation will be marked. A roll will be taken to record your attendance.

Mixed classes 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:

- Awareness of the life cycle of system development and the impact of implementation issues on various phases of the development life cycle.
- Knowledge of requirements elicitation techniques and the ability to apply these techniques to a wide range of problem domains.
- Ability to analyse the system requirements, build a logical model of the problem and turn the logical model from the analysis phase into a design model from which a system can be built.
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.
- Ability to use appropriate software tools to support the requirements, analysis and design phases.
- Ability to derive an architecture and design solution from requirements and analysis work products incorporating aspects of quality, testing, and other post-design workflow activities.

Assignment 1

Due: **Friday 12th September, 5pm**

Weighting: **15%**

Assignment 1 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 first part of the unit.

On successful completion you will be able to:

- Awareness of the life cycle of system development and the impact of implementation issues on various phases of the development life cycle.
- Knowledge of requirements elicitation techniques and the ability to apply these techniques to a wide range of problem domains.
- Ability to analyse the system requirements, build a logical model of the problem and turn the logical model from the analysis phase into a design model from which a system can be built.

Assignment 2

Due: **Friday 31st October, 5pm**

Weighting: **15%**

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 of the unit.

On successful completion you will be able to:

- Knowledge of requirements elicitation techniques and the ability to apply these techniques to a wide range of problem domains.
- Ability to analyse the system requirements, build a logical model of the problem and turn the logical model from the analysis phase into a design model from which a system can be built.
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.

Final Exam

Due: **Exam Period**

Weighting: **50%**

Final 3 hour exam

On successful completion you will be able to:

- Awareness of the life cycle of system development and the impact of implementation issues on various phases of the development life cycle.
- Knowledge of requirements elicitation techniques and the ability to apply these techniques to a wide range of problem domains.
- Ability to analyse the system requirements, build a logical model of the problem and turn the logical model from the analysis phase into a design model from which a system can be built.
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.
- Ability to derive an architecture and design solution from requirements and analysis work products incorporating aspects of quality, testing, and other post-design workflow activities.

Delivery and Resources

What is new?

Many many things. Comp255 is changing significantly this semester. While the goals of the unit are largely the same as in the past, the techniques that we will use to teach the unit, and the expectations that we have for student work, have changed significantly.

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 may be more appropriate to email the lecturer teaching during that week.

Teaching and Learning Strategies

COMP255 is taught via lectures in C5C T1 and mixed classes in the E6A laboratories.

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 sometimes 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.

Mixed classes are smaller group classes in the laboratories which give you the opportunity to interact with your peers and with a tutor who has sound knowledge of the subject. Mixed classes 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 practical application to problems set. 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 problems statements.

Each week you should:

- Attend lectures, take notes, ask questions
- Attend your mixed classes 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

Assumed knowledge

This unit requires an understanding of fundamental programming concepts and knowledge of simple data structures (obtained in COMP115). Additional programming experience will be an advantage.

Week	Topic - (subject to change)	Lecturer
1	System architecture and design	Mike
2	Requirements Engineering and Use Cases	Mike
3	Object Oriented Modelling - Concepts, class diagrams	Mike
4	Object Oriented Modelling - interaction, state and package diagrams, model consistency checking	Mike
5	Agile Software Development	Steve

6	Software Testing	Steve
7	Software Delivery and Deployment	Steve
8 (Public holiday Monday)	Design and Implementation	Mike
9	Software Quality	Mike
10	Software Maintenance	Mike
11	Software Security Engineering	Mike
12	Embedded Systems, Distributed Systems, and Mobile Applications	Mike
13	Fundamental Reflections on Software Engineering	Steve and Mike

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).

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.html>

Grading Policy <http://mq.edu.au/policy/docs/grading/policy.html>

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

Grievance Management Policy http://mq.edu.au/policy/docs/grievance_management/policy.html

Disruption to Studies Policy http://www.mq.edu.au/policy/docs/disruption_studies/policy.html *The Disruption to Studies Policy is effective from March 3 2014 and replaces the Special Consideration Policy.*

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/

Special Consideration

Special Consideration is intended for a student who is prevented by serious and unavoidable disruption from completing unit requirements in accordance with their ability. Go to this [site](#) to find the disruption to studies form that needs to be completed and submitted along with appropriate evidence to support your case. Depending on the circumstances presented, the convenor may choose to give you an extra assessment task, a supplementary examination, etc. If a Supplementary Examination is granted as a result of the Special Consideration process the examination will be scheduled after the conclusion of the official examination period. For details of the Special Consideration policy specific to the Department of Computing, see the Department's [policy page](#).

Grade Appeal

In case of problems arising with your final grade, the first step is to organise a review. The Department recommends that you request an appointment with the convenor of the unit in order to review your grade. If the review does not solve the problem, a formal Grade Appeal can be lodged. See the [grade appeal policy](#).

Academic Honesty and Plagiarism

Plagiarism involves using the work of another person and presenting it as one's own. The Department, in line with [University policy](#), treats all cases seriously. In particular, the Department keeps a record of all plagiarism cases. This record is referred to so that an [appropriate penalty](#) can be applied to each case.

Student Support

Macquarie University provides a range of support services for students. For details, visit <http://students.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](#)

Staff-Student Liaison Committee

The Department has established a Staff-Student Liaison Committee at each level (100, 200, 300) to provide all students studying a Computing unit the opportunity to discuss related issues or problems with both students and staff. The committee meets three times during the semester. For each meeting, an agenda is issued and minutes are taken. These are posted on the web. If you have exhausted all other avenues, then you should consult the Director of Teaching (Assoc Professor Steve Cassidy) or the Head of Department (Dr. Christophe Doche). You are entitled to have your concerns raised, discussed and resolved.

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.

If you have a serious disability please talk to the lecturer as early in the semester as possible. We are eager to help. Our focus is on learning, and we want to facilitate your individual learning.

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://informatics.mq.edu.au/help/>.

When using the University's IT, you must adhere to the [Acceptable Use Policy](#). The policy applies to all who connect to the MQ network including students.

Graduate Capabilities

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 outcome

- 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

- Awareness of the life cycle of system development and the impact of implementation issues on various phases of the development life cycle.
- Ability to analyse the system requirements, build a logical model of the problem and turn the logical model from the analysis phase into a design model from which a system can be built.
- An understanding of Software Engineering including its principles, practices, ethics and responsibilities.
- Ability to use appropriate software tools to support the requirements, analysis and design phases.
- Ability to derive an architecture and design solution from requirements and analysis work products incorporating aspects of quality, testing, and other post-design workflow activities.

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 analyse the system requirements, build a logical model of the problem and turn the logical model from the analysis phase into a design model from which a system can be built.
- Ability to derive an architecture and design solution from requirements and analysis work products incorporating aspects of quality, testing, and other post-design workflow activities.

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

- Knowledge of requirements elicitation techniques and the ability to apply these techniques to a wide range of problem domains.

Assessment tasks

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

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

- Knowledge of requirements elicitation techniques and the ability to apply these techniques to a wide range of problem domains.
- Ability to analyse the system requirements, build a logical model of the problem and turn the logical model from the analysis phase into a design model from which a system can be built.
- Ability to derive an architecture and design solution from requirements and analysis work products incorporating aspects of quality, testing, and other post-design workflow activities.

Assessment tasks

- Class Participation
- Assignment 2

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 outcome

- Ability to analyse the system requirements, build a logical model of the problem and turn the logical model from the analysis phase into a design model from which a system can be built.

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.

Grading

Grading

Comp255 is a graded unit. You will be awarded a grade (Pass, Credit, etc) as defined in the University's Grading Policy.

As well as depending upon your overall work in the unit, your final grade will depend on your performance in each part separately. In particular, in order to pass this unit

- You must perform at a satisfactory level in the examination.
- You must properly engage with the practical material in the unit as demonstrated by the submission of satisfactory attempts at assignment and practical work and the submission of weekly problem solutions.