COMP434

Advanced Software Engineering

S1 Day 2017

Dept of Engineering

Contents

General Information

Unit convenor and teaching staff
Michael Johnson
michael.johnson@mq.edu.au

Credit points
3

Prerequisites
(60cp at 100 level or above) including (COMP333 and COMP255)

Corequisites

Co-badged status

Unit description
This unit integrates prior learning in a specialist area of engineering with problem solving, emerging technology and aspects of engineering application, technical reporting and self-management to prepare students to work at a professional capacity. The unit aims to address the application of fundamental principles and methods at an advanced level in the context of standards and practices, modelling, analysis, design and practical implementation. The unit also develops skills in the critical evaluation of information, software and sources of error, and experimental methods. Learning will be achieved using case studies, laboratories, presentations, group work and traditional lecture format. The specific topics will focus on current advances in the area such as CASE tools, process methodologies, testability, software-hardware co-design, and formal verification techniques.

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:

Demonstrate an understanding of the ACS-EA Joint Board on Software Engineering competencies, and position one's own skill-base with respect to those competencies, in addition to demonstrable detailed technical skills in each of the areas
Develop an understanding of the profession of software engineering in Australia, and in less detail internationally, including significant industrial stakeholders, the roles of professional societies, legislative support for professionals, and the nature of

professionalism

Understanding of selected advanced concepts in software engineering, including topical issues and current research (more specific outcomes are negotiated individually for chosen advanced concepts)

Demonstrate self-learning, time-management, and project management, individually and in a group setting.

General Assessment Information

Note that each student is expected to complete satisfactorily all three components of the assessment. This is a highly integrated unit and attempts to merely accumulate marks in a component without utilising material from the others would be destined to fail.

This unit, like the workplace you will soon be in, requires active involvement and, like the workplace, you will be being judged throughout the semester.

There will be regular feedback during the semester, so students should have an idea of how they are progressing (and if you don't have a clear idea, be sure to ask!).

But it's really important to realise that this unit, the capstone unit in software engineering, is not like many (probably any) of the units you have studied before. You need to attend all classes and be actively involved. You need to work between classes and prepare for the following week's classes. You need to plan and manage carefully your own individual projects. And you need to take all this seriously and complete it in a business like and conscientious manner.

It goes without saying, but I'll say it anyway, that there are no recorded lectures or web-based Powerpoint presentations for you to use if you miss things. The unit is **about you** and your participation is essential.

Assessment Tasks

Name	Weighting	Hurdle	Due
Student engagement	30%	No	Every week
Student presentations	20%	No	To be agreed with each student
Submissions	50%	No	June 5

Student engagement

Due: **Every week**
Weighting: **30%**

The unit is built around highly interactive class debates for which students need to prepare between classes, and then they need to demonstrate critical thinking and active engagement in the in-class debates. Assessment of each student will be openly discussed with frequent feedback in a small class environment (and with each student's agreement).

This mode of learning is so important for the unit, and requires so much work by the students, that it has initially been given a heavy weighting in the assessment.

Nevertheless, as part of the active involvement of students the weightings, and indeed the assessment items themselves, are subject to negotiated change, and the items shown here and their weightings are current at the time of publication of the Unit Guide.

On successful completion you will be able to:

- Demonstrate an understanding of the ACS-EA Joint Board on Software Engineering competencies, and position one's own skill-base with respect to those competencies, in addition to demonstrable detailed technical skills in each of the areas
- Develop an understanding of the profession of software engineering in Australia, and in less detail internationally, including significant industrial stakeholders, the roles of professional societies, legislative support for professionals, and the nature of professionalism
- Understanding of selected advanced concepts in software engineering, including topical issues and current research (more specific outcomes are negotiated individually for chosen advanced concepts)

Student presentations

Due: **To be agreed with each student**
Weighting: **20%**

During the unit students are individually assigned problems to analyse (both critically themselves, and through a review of relevant literature) and the results of their studies are made available to the class in formal presentations.

Assessment of presentations includes a substantial component based on content, but also a component for surface features including the quality and clarity of their communication.

As part of the active involvement of students in assessment the weightings and the assessment items themselves are subject to negotiated change, and the items shown here and their weightings are current at the time of publication of the Unit Guide.

On successful completion you will be able to:

- Understanding of selected advanced concepts in software engineering, including topical issues and current research (more specific outcomes are negotiated individually for chosen advanced concepts)
- Demonstrate self-learning, time-management, and project management, individually and in a group setting.

Submissions

Due: **June 5**
Weighting: **50%**

There will be a sequence of agreed written deliverables (these can include problem solutions, software, reports and essays) required throughout the unit. The most important of these is the unit portfolio, and it is expected that the other deliverables will form a part of it too. Nevertheless, some deliverables need to be submitted during the unit, while the portfolio will only be finally submitted at the end of the unit (June 5).

The unit portfolio is a student's record of their significant achievements, including their reflections, from throughout the unit. Thus it is both a technical document, and a personal journal. The personal reflections are very important (refer again to the learning outcomes).

As part of the active involvement of students in assessment the weightings and the assessment items themselves are subject to negotiated change, and the items shown here and their weightings are current at the time of publication of the Unit Guide.

On successful completion you will be able to:

- Demonstrate an understanding of the ACS-EA Joint Board on Software Engineering competencies, and position one's own skill-base with respect to those competencies, in addition to demonstrable detailed technical skills in each of the areas
- Develop an understanding of the profession of software engineering in Australia, and in less detail internationally, including significant industrial stakeholders, the roles of professional societies, legislative support for professionals, and the nature of professionalism
- Understanding of selected advanced concepts in software engineering, including topical issues and current research (more specific outcomes are negotiated individually for chosen advanced concepts)
- Demonstrate self-learning, time-management, and project management, individually and in a group setting.

Delivery and Resources

This unit emphasises a dynamic combination of lecture, practical and laboratory work, group and individual project work, and a high degree of self-learning. The overall goal is to be a "finishing school" for software engineers, and the unit includes highly technical work as well as deep reflection on the nature of the discipline and its state in Australia and internationally.

Research "in the library" as to the state of the art in software engineering will be required as the student develops his or her own understanding in identified areas as part of the project work.

There are no set texts, but a wide range of sources will need to be consulted and reading lists

developed.

This unit is different each year because the content is tailored to the individual experiences of each of the students that arise from their particular choices of electives in the software engineering program. The exact nature of the change from year to year depends of course upon the exact nature of the individual experiences of the enrolled students in each year.

Unit Schedule

Week commencing	**Topic**
29 February	Intro and raison d'etre for 434
7 March	The nature of the discipline
14 March	The role of professional societies
21 March	Development of individual projects 1
28 March	Development of individual projects 2
4 April	Development of individual projects 3
11 April	Break from classes (work on projects)
18 April	Break from classes (work on projects)
25 April	Reports on individual projects
2 May	Software engineering disasters
9 May	Software assurance and operating systems
16 May	Software security and machine code
23 May	Systems security
30 May	Mobile systems
6 June	Software verification
13 June	Final Reporting / Examination if required

Please note that this is the initial plan for arranging unit content, but it is subject to change. Our goal is to best serve student learning and depending on the distribution of Joint Board

competencies already attained by the incoming students material may be reaarranged and relative timings reallocated (in consultation with the enrolled students).

Individual projects will cover many areas including CASE tools, process methodologies, testability and formal verification as well as project specific technical material. The precise combination will vary based on the nature of each individual project.

Learning and Teaching Activities

Lecture

Delivery of material not previously seen by the students or material which will be presented in a differenct context with regard to graduate capabilities. There may be some review material, but this is minimal

Laboratory

Develop skills based competencies in experimentation with overlap/application to theory and simulation

Projects

Students plan and execute a combination of group and individual work to execute a project of substance, possibly with real world application. This activity leads to assessments that may be of both a group and individual nature as well as formal reports and a presentation

Class debates

This activity is used widely in this unit to engage students and encourage deep learning. At this advanced level, as well as technical material there is a need for students to develop their own internalised understanding of matters such as ethics, the nature of the profession, their approach to professional development, and their career goals. There aren't simple answers, and there is a strong need for detailed student engagement with the issues.

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://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

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

- Develop an understanding of the profession of software engineering in Australia, and in less detail internationally, including significant industrial stakeholders, the roles of professional societies, legislative support for professionals, and the nature of professionalism
- Understanding of selected advanced concepts in software engineering, including topical issues and current research (more specific outcomes are negotiated individually for chosen advanced concepts)

Assessment tasks

- Student engagement
- Submissions

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

- Demonstrate self-learning, time-management, and project management, individually and in a group setting.

Assessment task

- Submissions

Commitment to Continuous Learning

Our graduates will have enquiring minds and a literate curiosity which will lead them to pursue knowledge for its own sake. They will continue to pursue learning in their careers and as they participate in the world. They will be capable of reflecting on their experiences and relationships

with others and the environment, learning from them, and growing - personally, professionally and socially.

This graduate capability is supported by:

Learning outcomes

- Demonstrate an understanding of the ACS-EA Joint Board on Software Engineering competencies, and position one's own skill-base with respect to those competencies, in addition to demonstrable detailed technical skills in each of the areas
- Understanding of selected advanced concepts in software engineering, including topical issues and current research (more specific outcomes are negotiated individually for chosen advanced concepts)
- Demonstrate self-learning, time-management, and project management, individually and in a group setting.

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

- Demonstrate an understanding of the ACS-EA Joint Board on Software Engineering competencies, and position one's own skill-base with respect to those competencies, in addition to demonstrable detailed technical skills in each of the areas
- Develop an understanding of the profession of software engineering in Australia, and in less detail internationally, including significant industrial stakeholders, the roles of professional societies, legislative support for professionals, and the nature of professionalism
- Understanding of selected advanced concepts in software engineering, including topical issues and current research (more specific outcomes are negotiated individually for chosen advanced concepts)

Assessment tasks

- Student presentations
- Submissions

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

- Develop an understanding of the profession of software engineering in Australia, and in less detail internationally, including significant industrial stakeholders, the roles of professional societies, legislative support for professionals, and the nature of professionalism
- Understanding of selected advanced concepts in software engineering, including topical issues and current research (more specific outcomes are negotiated individually for chosen advanced concepts)

Assessment tasks

- Student engagement
- Student presentations
- Submissions

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 outcomes

- Demonstrate an understanding of the ACS-EA Joint Board on Software Engineering competencies, and position one's own skill-base with respect to those competencies, in addition to demonstrable detailed technical skills in each of the areas
- Understanding of selected advanced concepts in software engineering, including topical issues and current research (more specific outcomes are negotiated individually for chosen advanced concepts)

Assessment tasks

- Student presentations
- Submissions

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

- Demonstrate self-learning, time-management, and project management, individually and in a group setting.

Assessment tasks

- Student engagement
- Student presentations
- Submissions

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 outcomes

- Develop an understanding of the profession of software engineering in Australia, and in less detail internationally, including significant industrial stakeholders, the roles of professional societies, legislative support for professionals, and the nature of professionalism
- Demonstrate self-learning, time-management, and project management, individually and in a group setting.

Assessment task

- Student engagement

Socially and Environmentally Active and Responsible

We want our graduates to be aware of and have respect for self and others; to be able to work with others as a leader and a team player; to have a sense of connectedness with others and country; and to have a sense of mutual obligation. Our graduates should be informed and active participants in moving society towards sustainability.

This graduate capability is supported by:

Learning outcomes

- Develop an understanding of the profession of software engineering in Australia, and in less detail internationally, including significant industrial stakeholders, the roles of professional societies, legislative support for professionals, and the nature of professionalism
- Demonstrate self-learning, time-management, and project management, individually and in a group setting.

Unit goals

As we've said, comp434 is a very different unit from the other kinds of units you will have studied in your software engineering (SE) program. So, we want to be explicit about the unit's goals. They are:

1. To fill the gaps in ICT knowledge that can arise from a curriculum for SE including a limited number of required specialist IT units. Each student has different gaps, as they depend on the students' choices of electives and on his or her goals for future employment in software engineering (there are many different kinds of software engineers).
2. To provide a capstone experience in which students reflect upon their entire degree and contextualise it with respect to other software engineering degrees, SE curricula, and their own understanding of SE, as well as developing their own near-graduation perspective on SE as a field, on professionalism and on professional practice.

These are the goals that lie behind the learning outcomes.