



COMP430

Software Engineering Practices

S2 Day 2018

Dept of Computing

Contents

<u>General Information</u>	2
<u>Learning Outcomes</u>	2
<u>General Assessment Information</u>	3
<u>Assessment Tasks</u>	4
<u>Delivery and Resources</u>	6
<u>Policies and Procedures</u>	6
<u>Graduate Capabilities</u>	7
<u>Changes from Previous Offering</u>	11

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, Lecturer, Tutor

Anthony Sloane

anthony.sloane@mq.edu.au

Contact via Email

9WW315

By appointment

Credit points

3

Prerequisites

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

Corequisites

Co-badged status

Unit description

This unit provides an opportunity for students to practice and demonstrate their software engineering skills within the context of a team. Students will work together to understand, design, modify, test and deliver non-trivial software using practices that are in use in the information technology industry.

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:

Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.

Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.

Effectively use modern software development tools such as version control systems and issue trackers.

Effectively communicate progress and results of the software development process.

General Assessment Information

COMP430 applies an agile method to the engineering of features in a non-trivial software project. Students work as a team on a project that uses technology with which they are already familiar or that has a relatively short learning curve. Normally the project will not be developed from scratch. Rather, COMP430 approximates the normal situation in real-world software development where a team is required to understand, enhance and fix existing software.

The three main assessment items are focused on the software artefact(s) being developed by the class as a team. The items will be assessed in a continuous manner with feedback being provided frequently by the unit convenor to each student.

Students will be asked to document their contributions and associated evidence in each of the three assessment areas of Planning and Estimation, Development and Review as the semester unfolds. E.g., documentation of a development activity might include a link to the associated commits and pull requests in the source code repository. Or documentation of planning and estimation contributions might cite the role a student has played in the discussion that took place in a class session.

Based on the supplied evidence and the convenor's own observations, these assessment items will be assessed according to the following standards:

- * Distinction/High Distinction: An extremely valuable team member who makes many key contributions to many different aspects of the software that is developed by the team.
- * Credit: A team member who makes useful contributions to multiple aspects of the team's software.
- * Pass: A team member who makes a small number of contributions to the team's software or focuses on a single area of contribution.

The team contribution assessment item captures contribution to the functioning of the team rather than specific contribution to development of the software artefact (which is captured by the other three assessment items). E.g., a high team contribution mark might be earned by a student who often comes up with useful suggestions in planning discussions for features proposed by other team members or who volunteers to present some useful technology to the team so everyone can learn about it. Students are encouraged to think of ways in which they would like to contribute but the unit convenor will also suggest opportunities. Students should plan to attend all classes since they are vital meetings of the software team.

Participation will be assessed according to the following standards:

- * Distinction/High Distinction: An extremely valuable team member who contributes strongly in many, varied ways to the smooth and efficient functioning of the team.
- * Credit: A team member who makes multiple varied contributions to the team's operations, not just in a single particular way.
- * Pass: A team member who makes a small number of contributions to the way the team operates or whose contributions have a single focus.

Assessment Tasks

Name	Weighting	Hurdle	Due
<u>Planning and Estimation</u>	30%	No	Throughout
<u>Development</u>	30%	No	Throughout
<u>Review</u>	30%	No	Throughout
<u>Team contribution</u>	10%	No	Throughout

Planning and Estimation

Due: **Throughout**

Weighting: **30%**

This assessment item measures student contributions to team planning and estimation activities. Included are activities such as soliciting and developing client input, proposing new features or fixes, estimating how much functionality can be achieved in a development period, including taking into account review of experiences in previous periods.

On successful completion you will be able to:

- Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.
- Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.
- Effectively use modern software development tools such as version control systems and issue trackers.
- Effectively communicate progress and results of the software development process.

Development

Due: **Throughout**

Weighting: **30%**

This assessment measures student contribution to the team's software development activities, including development of tests for a proposed feature, coding the feature, debugging, ensuring that tests pass and submission to the team's version control repository.

On successful completion you will be able to:

- Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.

- Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.
- Effectively use modern software development tools such as version control systems and issue trackers.
- Effectively communicate progress and results of the software development process.

Review

Due: **Throughout**

Weighting: **30%**

This assessment item measures student contribution to team review activities. Included are activities such as code and test review of features developed by other team members, merging features into the baseline version of the system, and reflection on achievements in a development period.

On successful completion you will be able to:

- Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.
- Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.
- Effectively use modern software development tools such as version control systems and issue trackers.
- Effectively communicate progress and results of the software development process.

Team contribution

Due: **Throughout**

Weighting: **10%**

The participation assessment item captures overall contribution to the functioning of the team rather than specific contributions to the software artefact (which is captured by the other three assessment items).

On successful completion you will be able to:

- Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.
- Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.
- Effectively use modern software development tools such as version control systems and issue trackers.

- Effectively communicate progress and results of the software development process.

Delivery and Resources

The focus of the unit is on development periods that iterations of a few weeks duration where milestones such as new features, bug fixes and the like are proposed, planned, delivered and reviewed. Students will develop code by themselves or in pairs, but students should aim to participate in each aspect of the project rather than focusing one or two aspects. In particular, students will be asked to review each other's code and participate in reviews of each software milestone.

Tools such as distributed version control and milestone planning software will be used throughout to coordinate activities.

In each week there is a one hour lecture during which agile principles, useful technologies and the problem domain will be discussed and the team will meet to discuss plans and progress. The two hour practical session will be used to continue these discussions and for sub-groups to work in a more focused manner on current activities.

Policies and Procedures

Macquarie University policies and procedures are accessible from [Policy Central \(https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-central\)](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 \(https://students.mq.edu.au/support/study/student-policy-gateway\)](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 \(https://staff.mq.edu.au/work/strategy-planning-and-governance/university-policies-and-procedures/policy-central\)](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://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

- Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.
- Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.

Assessment tasks

- Planning and Estimation
- Development
- Review

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

- Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.
- Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.

Assessment tasks

- Planning and Estimation
- Development
- Team contribution

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

- Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.
- Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.
- Effectively use modern software development tools such as version control systems and issue trackers.
- Effectively communicate progress and results of the software development process.

Assessment tasks

- Planning and Estimation
- Development
- Review
- Team contribution

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

- Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.
- Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.
- Effectively use modern software development tools such as version control systems and issue trackers.
- Effectively communicate progress and results of the software development process.

Assessment tasks

- Planning and Estimation
- Development
- Review

- Team contribution

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

- Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.
- Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.
- Effectively communicate progress and results of the software development process.

Assessment tasks

- Planning and Estimation
- Development
- Review

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

- Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.
- Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.
- Effectively use modern software development tools such as version control systems and issue trackers.

Assessment tasks

- Planning and Estimation

- Development

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

- Understand and perform the stages of a modern software development process to achieve non-trivial outcomes.
- Understand and make use of principles of software project management, particularly relating to teamwork, roles and responsibilities.
- Effectively use modern software development tools such as version control systems and issue trackers.
- Effectively communicate progress and results of the software development process.

Assessment tasks

- Planning and Estimation
- Review
- Team contribution

Changes from Previous Offering

This is a new unit in 2018.