



# COMP4130 Managing Software Quality and Process



By Zhenchang Xing

# CECS Class Representatives

---

## **Roles and Responsibilities**

- ✓ Be creative and proactive in gathering feedback from your class mates about the course.
- ✓ Act as the official liaison between your classmates and your lecturers in communicating feedback about the course and providing course-related updates to your classmates.
- ✓ Provide regular reports to the Associate Director (Education) on the feedback you've been gathering.

## **Benefits of Being a Class Rep**

- ✓ The opportunity to develop skills sought by employers – particularly interpersonal, dispute resolution, leadership and communication skills.
- ✓ Empowerment: Play a more active role in determining the direction of your education. Become more aware of issues influencing your University and current issues in higher education.

## **Nominations**

- ✓ Please contact CECS Student Services ([studentadmin.cecs@anu.edu.au](mailto:studentadmin.cecs@anu.edu.au)) with your name, Student ID and the course number (e.g. COMP4130) you are interested in becoming a Class Representative for.
- 



# Module 1

Introduction to Managing Software Quality and Process

# Module 1 Overview (Week 1)

---

- ▶ **Topic 1.1 – About the Teacher and the Course**
  - ▶ My self-introduction, assumptions, and expectations
  - ▶ Course overview – modules, schedule, assessment
- ▶ **Topic 1.2 - Foundations of Managing Software Quality and Process**
  - ▶ Project success and quality factors
  - ▶ Quality in the lifecycle – right product, done right, managed right
  - ▶ Monitoring, metrics and feedback
  - ▶ Why agile?
  - ▶ Career opportunities




# Learning Outcomes

---

- ▶ Identify quality factors of project success
  - ▶ Quality factors are neither objective nor subjective
  - ▶ FURPS+ and process quality
- ▶ Summarize the three viewpoints (goals) to build better software
  - ▶ Right product + done right + managed right
  - ▶ The importance of monitoring and feedback
- ▶ Reflect on how Agile practices will improve software product
  - ▶ Principles in Agile Manifesto





# Module 1 – Topic 1.1



About the Teacher and the Course

# Topic Outline

---

- ▶ **Meet Your Teacher and Tutor**
  - ▶ My self-introduction, assumptions and expectations
- ▶ **Course Overview**
  - ▶ Course modules and topics
  - ▶ Course schedule – lectures, workshops/laboratories, assignments
  - ▶ Assessment schema (we have **hurdle assessment**)



# My Contact Details

---

## ▶ Zhenchang Xing

- ▶ Web page: <https://cecs.anu.edu.au/people/zhenchang-xing>
- ▶ Email: [zhenchang.xing@anu.edu.au](mailto:zhenchang.xing@anu.edu.au)
- ▶ Office phone: 55692
- ▶ Office: CSIT N328
- ▶ Consultation: by appointment





# I am a World Traveller ☺

---



I joined the RCCS in  
October 2016!

---

# I am a Software Engineering Researcher

---

- ▶ Software analysis, e.g.,
  - ▶ Code clone analysis: <http://www.se.fudan.edu.cn/clonepedia/diff/>
  - ▶ Time-travel debugging:  
<http://linyun.info/microbat/index.html>
- ▶ Big data analytics, e.g.,
  - ▶ Technology landscape:  
<http://graphofknowledge.appspot.com/>
  - ▶ Similar tech recommendation:  
<http://graphofknowledge.appspot.com/similartech>
  - ▶ Dual-lingual question retrieval:  
<http://yisearch.org/>
- ▶ Human-Computer Interface, e.g.,
  - ▶ ActivitySpace:  
<http://baolingfeng.weebly.com/ase2015-demonstration.html>

Feel free to approach  
me if you are interested  
in such research topics!



# My Teaching History

---

- ▶ I taught a second-year undergraduate Introduction to Software Engineering course in Nanyang Technological University (NTU), Singapore
- ▶ The course introduces the students to the world of software engineering. It covers
  - ▶ Software process
  - ▶ Project planning
  - ▶ SDLC activities – specification, design, testing (this is the focus)
  - ▶ Software maintenance
- ▶ Best match in RCCS curriculum
  - ▶ COMP2130?



# My NTU Course versus COMP4130

---

## ▶ My NTU course

- ▶ The students know nothing about software engineering
- ▶ Introduce a holistic view of software engineering
- ▶ Cover a wide range of software engineering techniques at introductory level

## ▶ COMP4130

- ▶ You already learned much software engineering knowledge
- ▶ Help you connect dots into a knowledge network
- ▶ Cover mainly techniques related to software quality and process



# My Assumptions and Expectations

---

- ▶ You are here to learn. I create an environment in which you can learn, not just from me, but also from your peers
- ▶ A "memorising" approach is NOT a right approach for learning in software engineering course
- ▶ Active participation in lectures and workshops is a necessary part of the course



# Feedback is Important!

---

- ▶ This is my **first time** teaching in ANU. Please help me improve the course and my teaching.
- ▶ I'd love to hear your feedback
- ▶ Really
- ▶ In person, email, Wattle ...
  - ▶ Whichever way you feel like





# Course Overview

# Five Modules

---

- ▶ Introduction to Managing Software Quality and Process (Week 1)
- ▶ Software Process and Agile Practices (Week 2)
- ▶ Right Product – Client Needs and Software Requirements (Week 3 – Week 5)
- ▶ Done Right – Reviews, Testing, and Metrics for Software Improvement (Week 6 – Week 10)
- ▶ Managed Right – Project Planning and Monitoring (Week 10 – Week 11)

See Module Summary on:

<https://wattlecourses.anu.edu.au/mod/book/view.php?id=1065096>

---





# Each Module Consists of:

---

- ▶ A set of topics, which will be introduced in the lectures.
- ▶ A set of discussions, exercises and/or lab hand-ons, which will be conducted in the workshops and laboratories.
- ▶ 1 to 2 individual assignments, through which you will practice the concepts you have learned in the lectures and workshops.
- ▶ 1 to 3 readings, which provide supplementary materials to assist the learning in the lectures and workshops and to help complete the assignments.

**Reminder:** Review Topic Plan and complete Assignment 1.1 Background Survey (Due on 23:59:59 Monday 27/02/2017)

---



# Design of Course Modules

---

## ▶ Aims

- ▶ Help you **connect the dots** you have learned in the previous courses **into a knowledge network** from the perspective of software quality in the lifecycle
- ▶ Adds a set of new skills to your skill sets for the three viewpoints of building better software: **right product, done right, and managed right**

## ▶ Broadness versus depth

- ▶ We would like to achieve a good balance between topic broadness and topic depth
- ▶ In between broadness and depth, we lean more towards broadness



# Course Information

---

- ▶ All is on Wattle website

- <https://wattlecourses.anu.edu.au/course/view.php?id=19311>

- ▶ Module summary, topic plan, discussions-exercises-labhandsons, readings, assignments
  - ▶ Course schedule, workshop/laboratory manuals, lecture slides and notes
  - ▶ Will remain stable, but will be continuously updated

- ▶ We do not have textbooks for most parts of the course

- ▶ For software testing part, we will use **Software Testing: A Craftsman's Approach** by Paul C. Jorgensen.

- ▶ No need to buy it



# Teaching & Learning Activities

---

- ▶ 20 lectures (Week 1 – Week 11)
  - ▶ Monday & Wednesday 9:00am – 10:30am, Han 2.24
- ▶ 7 workshops and 2 laboratories
  - ▶ Workshops (Week 2/3/4/5/8/10/11) :Thursday 11:00am – 13:00pm, CSIT N101
  - ▶ Laboratories (Week 6/9) :Thursday 11:00am – 13:00pm CSIT N111
- ▶ 2 Q&A sessions (Week 7/12)
  - ▶ Thursday 11:00am – 13:00pm, CSIT N101
- ▶ 12 compulsory readings
  - ▶ Most are short articles and videos for workshop and laboratory preparation
- ▶ 7 individual assignments
  - ▶ 3 light assignments + 2 assignments on requirements + 2 assignments on testing

Encourage active participation and learning!



# Lectures

---

- ▶ **Interaction** is very important
- ▶ Feel free to **interject** the lecture for questions
- ▶ In class, we will use Socrative web-based tool to help me collect your answers to questions I pose



# Workshops and Laboratories

---

- ▶ Three types of activities (check [workshop/laboratory manual](#) and [activity details](#) on Wattle course website)
  - ▶ 7 discussions: about fundamental concepts; MCQs-driven or scenario-based discussion
  - ▶ 4 exercises: analysis and design tasks on requirements and testing
  - ▶ 2 lab hands-ons: pair programming and test-driven development tasks
- ▶ **10% of total course marks** for workshop and laboratory participation (not attendance). Evaluation will be based on:
  - ▶ Answers to MCQ questions
  - ▶ Engagement in the group discussion
  - ▶ Presentation of discussion and exercise solutions
  - ▶ Experience summary in laboratory hands-ons.



# Assignments (No Late Submissions will be Accepted!)

---

- ▶ 7 individual assignments (**50% of total course mark**) (Check [assessment details and assessment criteria](#) on Wattle course website).
    - ▶ 1%, Assignment 1.1 Background survey (due on 27/02/2017, Week2)
    - ▶ 1%, Assignment 1.2 Online investigation of software failures and causal factors (due on 27/02/2017, Week2)
    - ▶ 3%, Assignment 2.1 Waterfall model controversies and improvements (due on 06/03/2017, Week3)
    - ▶ 10%, Assignment 3.1 Wireframing and storyboard (Due on 27/03/2017, Week6)
    - ▶ 10%, Assignment 3.2 User stories and acceptance test (Due on 27/03/2017, Week6)
    - ▶ 15%, Assignment 4.1 Black box testing (Due on 15/05/2017, Week11)
    - ▶ 10%, Assignment 4.2 White box testing (Due on 15/05/2017, Week11)
  - ▶ Submissions will be through Wattle website. Due time is **23:59:59** Monday of the due week.
  - ▶ No late submissions will be accepted, unless extension is granted otherwise.
- 



# Assessment Schema

---

- **Hurdle assessment:** >50% overall & >50% for assignments and workshop/lab participation & >50% for final exam

Assessment Item	Assessment Summary	Percentage of total course marks
Assignments	<ul style="list-style-type: none"><li>• One background survey to familiarize course topics (1%)</li><li>• Two light assignments for workshop activities preparation (4%)</li><li>• Four major assignments for requirements and testing (45%)</li></ul>	50%
Workshop/Lab Participation	<ul style="list-style-type: none"><li>• Participation in the readiness questions, discussion and presentation of 7 workshops and 2 laboratories</li></ul>	10%
Final Exam	<ul style="list-style-type: none"><li>• One, three-hour written examination during the official examination period at the end of the course</li></ul>	40%





# Student Time Budget

---

- ▶ The design of this course assumes that you will commit **an average of ~10 hours per week** to the course

Activity	Hours
Lecture preparation	10
Lecture attendance	30
Workshop/lab preparation	9
Workshop/lab participation	18
Reflection	12
Readings	16
Assignments	31
<b>Total</b>	<b>126</b>

