Software Engineering COMP 201

Teaching staff
Sebastian Coope
Ashton Building, Room G.18
E-mail: coopes@liverpool.ac.uk

Mr Henry Forbes
H.J.Forbes@liverpool.ac.uk
Ashton Building, Room G.17

See Canvas for all notes

Lecture 1 – Module Introduction

This module

- Is NOT focused largely on programming or programming technology (APIs etc), we do however look at some coding issues…. (e.g. coupling, cohesion and code structure)
- Why focus away from coding…
  - This is covered in other modules
  - System failures are rarely about lack of knowledge of programming language or incorrect use of API or programming language
- Failures are very often due to
  - Wrong or missing requirements (missing or wrong features)
  - Poor software design (software is inflexible and fragile)
  - Problems with testing (bugs)
  - Weak risk analysis and/or project management (project overrun)
  - Problem with process (or lack of process) (overrun, cancelled)

What is SOFTWARE ENGINEERING

- ENGINEERING
  - Making stuff
  - But
    - In a structured and disciplined manner!
    - Using tried and tested approaches
- SOFTWARE
  - Code (instructions)
  - Data designs (data base schemas)
  - AI training models

Why Software Engineering?

- **Software development is hard** !
- **Important to distinguish :**
  - **"easy" systems** (*one developer, one user, experimental use only*)
  - **"hard" systems** (*multiple developers, multiple users, products*)
- **Experience with "easy" systems is misleading**
  - *Single person techniques do not scale up*
- **Analogy with bridge building:**
  - Over a stream = easy, one person job
  - Over River Severn … ? (*the techniques do not scale)*

Why Software Engineering ?

- The problem is ***complexity***
- There are many sources of complexity, but size is key:
  - The Linux kernel contains >13 million lines of code
  - Windows XP contains >40 million lines of code

Software engineering is about managing this complexity.

Why Software Engineering ?

- Software failure can be very serious
  - Software controls safety critical systems
  - Software protects sensitive data
  - Software is involved in systems which handle money
- Software Engineering has to
  - Produce software which has a very low chance of faulting
  - Be able to demonstrate/proof that software has very low chance of fault
    - Testing or program proving

Software engineering tasks

- Define requirements
  - What should it do?
- Design the product
  - Design how the product should look and be constructed
  - UI design, software module design, data design (what data?)
- Implement and test
  - **Coding**, testing and validation
- Managing the process
  - Software project management

Teaching Method

- Lectures 3 hours/week Streams also on Canvas
- 1 Hour/week seminar, tutorial
- Practical work **(3 Assignments)**

---------- **Course Assessment** ----------

- A two-hour examination: **60%**
- Coursework: **40%**

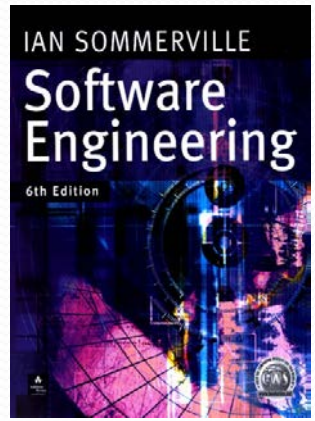
COMP201 Practical

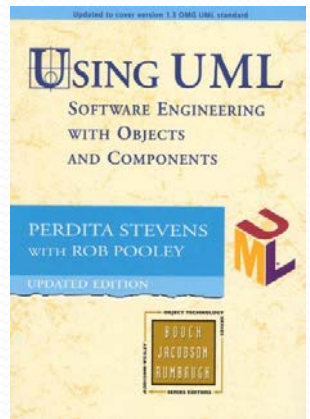
- Practical slots: (from WEEK THREE)
  - 1 hour/week
  - Use these sessions to do your coursework
- COMP 201 Assignments
  - Assignment 1 – Part 1 Requirements Engineering **(20%)**
    - Use case analysis, non-functional requirements
  - Assignment 2 – Part 1 Design and implementation **(5%)**
  - Assignment 2 – Part 2 Modelling with UML **(15%)**

Recommended Course Textbooks

- I. Sommerville (2001,2004, 2007)
  Software Engineering 6th ,7th or 8th Edition,
  Addison-Wesley, Harlow, Essex, UK

- P. Stevens with R. Pooley (2000),
  Using UML: Software Engineering with Objects and Components, 1st or 2nd Edition,
  Addison-Wesley, Harlow, Essex, UK





Outline Syllabus

- ***Introduction to Software Engineering***
- ***Software models***
- ***Software requirements***
- ***Software Design and Implementation***
  - ***UML (Unified Modeling Language)***
- ***Software verification, validation and testing***
- ***Management of Software Projects & Cost Estimation***

Software

Software Engineering

- The economies of ALL developed nations are dependent on software.
- More and more systems are software controlled
- Software engineering is concerned with theories, methods and tools for **professional software development**.
- Some software can be classified as **critical** (air traffic control, medical software, nuclear reactor control software..).

FAQs about Software Engineering

- What is:
  - software?
  - a software process?
  - software engineering?
  - a software process model?

Software and machine learning

- For machine learning AI type models the software consists of
  - A network with internal state (typically a neural net)
- The network is trained
- The training as well as the net itself is the software
- Implications
  - The model is extremely complex and hard to understand explicitly
  - The determinations of the network are probabilistic and therefore can be wrong

Examples of Software products

- Programs
  - Games, servers, databases, languages, office applications
  - Crypto coin wallets
  - Apps, embedded systems
- Documents
  - User manuals
  - Content
  - Designs and specifications

What is Software Engineering?

Software engineering is an engineering discipline which is concerned with *all aspects* of software production

Software engineers should

- adopt a systematic and organised approach to their work
- use appropriate tools and techniques depending on
  - the problem to be solved,
  - the development constraints and
  - the resources available

Good Software Engineers

- Tend to
  - Not keep making the SAME mistakes again and again
    - (so document your mistakes)
  - Use tried and tested approaches
  - Communicate well with others
  - Work well in teams
  - Document their work
  - Spend a lot of time testing
  - Produce code which can be fixed/modified and understood easily by others
  - Can predict their own productivity
- Being a good software engineer
  - Is not very easy!

Major Software Failures

- **Colonial Pipeline's costly ransomware attack 2021**
  - **Shut down fuel pipe line to Eastern US**
- **Tesla recalls almost 12,000 vehicles November 2021**
- **Log4j software bug leaves millions of web servers vulnerable Dec 2021**
- **Therac-25 (1985-1987)** : six people overexposed during treatments for cancer
- **Taurus (1993)** : the planned automatic transaction settlement system for London Stock Exchange cancelled after five years of development
- **Ariane 5 (1996)** : rocket exploded soon after its launch due error conversion (16 floating point into 16-bit integer leading to an exception)
- **The Mars Climate Orbiter :** assumed to be lost by NASA officials (1999): different measurement systems (Imperial and metric)

Recent failure…

- Mon, 22nd Jul 2024 Microsoft blue screen of death
- Widespread outage caused by a faulty CrowdStrike update, Falcon malware check
- Azure cloud service failure
- We currently estimate that CrowdStrike's update affected 8.5 million Windows devices (< 1 percent)
- Many medical services in UK affected, TV stations, airlines
- Problems with
  - Single point of failure (monopoly of service), failure to test

What is a Software Process?

- A Software Process is a **set of activities** whose goal is the development or evolution of software
- Fundamental activities in all software processes are:
  - **Specification** - what the system should do and its development constraints
  - **Development** - production of the software system (design and implementation)
  - **Validation** - checking that the software is what the customer wants
  - **Evolution** - changing the software in response to changing demands

What are the Attributes of Good Software?

The software should deliver the required functionality and performance to the user and should be maintainable, dependable, efficient and usable.

- **Maintainability**
  - Software must (easily) evolvable to meet changing needs
- **Dependability**
  - Software must be trustworthy (work with all data)
- **Efficiency**
  - Software should not make wasteful use of system resources
- **Usability**
  - Software must be usable by the users for which it was designed

Professional and Ethical Responsibility

- Software engineering involves wider responsibilities than simply the application of technical skills.
- Software engineers must behave in an honest and ethically responsible way if they are to be respected as professionals.
- Ethical behaviour is more than simply upholding the law.

Issues of Professional Responsibility

- Confidentiality
  - Engineers should normally respect the confidentiality of their employers or clients even without a formal confidentiality agreement.
- Competence
  - Engineers should not misrepresent their level of competence. They should not knowingly accept work which is beyond their competence.

Issues of Professional Responsibility

- Intellectual property rights
  - Engineers should be careful to ensure that the intellectual property of employers and clients is protected and know the local laws governing IP.
- Computer misuse
  - Software engineers should not use their technical skills to misuse other people's computers.

Ethics and software engineering

Tesla crashed due to software issues

The Washington Post analyzed NHTSA's numbers and found that Autopilot was involved in 736 crashes since 2019, including 17 fatalities

Ethics for the 21st Century

- What if a robot you design the AI software for
  - Hurts (kills) someone, causes damage
- Who is responsible?
  - You?
  - The owner of the robot?
  - The robot?

  https://en.wikipedia.org/wiki/Self-driving_car_liability
- What if a robot you design and produce, designs and produces another robot? Etc. etc.

Lecture Key Points

- We have seen the reasons for requiring solid software engineering principles in modern systems
- Software engineering is an engineering discipline concerned with all aspects of software production.
- Software products consist of developed programs and their associated documentation with several essential product attributes such as maintainability, dependability, efficiency and acceptability.
- Software Engineers have responsibilities to the engineering profession and society and should not simply be concerned with technical issues.