

Welcome & Introduction

COMP6204: Software Project Management and Secure Development

Dr A. Rezazadeh (Reza)

Email: ra3@ecs.soton.ac.uk or ar4k06@soton.ac.uk

October 21

Overview

- House Keeping
 - Lecture times
 - Syllabus
 - Assignments
 - Exams
 - Resources
 - Where to get help.
- What is software Engineering

Teaching team

Reza Rezazadeh -
ar4k06@soton.ac.uk
Module Leader

Joshua Curry -
jsc3g14@soton.ac.uk
Lecturer

Housekeeping - What, when, where?

- We have 3 lectures a week, these will be recorded and available on line.
- One Self-paced laboratories/Online session
 - Please, get to know your group and start working

Mon	Tue	Wed	Thu	Fri
2PM - 3PM COMP6204 L3 - Software Prjct Man & S 34 / 3001 (L/T) 1-11, 15		10AM - 11AM COMP6204 L1 - Software Prjct Man & S 35 / 1005 1-11, 15	5PM - 6PM COMP6204 C - Software Prjct Man & S Online Delivery 1-11, 15	2PM - 3PM COMP6204 L2 - Software Prjct Man & S 02 / 1039 (L/T K) 1-11, 15

- Coursework laboratories.
 - Use Discord for laboratories' discussions
 - Log into <https://discord.ecs.soton.ac.uk> and then follow the prompts, you should get automatic access to the COMP6204 area
Josh will also be monitoring the discussion too.

Syllabus

- <https://secure.ecs.soton.ac.uk/module/2122/COMP6204/32959/syllabus>
- This module is to prepare students for undertaking large software projects.
 - It introduces the students to the high-level strategies required for managing projects from their genesis to completion.
- The module also introduces the students to secure engineering of software systems.
 - The practical aspect will enable students to gain practical coding skills in secure software development for web-based applications.
- While no prior knowledge of a specific programming language is assumed, students should already be competent in at least one high-level language.

Syllabus

- Aims and objectives
 - To prepare students for undertaking large software projects.

- Knowledge and Understanding

Having successfully completed the module, you will be able to demonstrate knowledge and understanding of:

- A1. Formal management for software projects
- A2. Quality assurance practices for software projects

- Subject Specific Intellectual and Research Skills

Having successfully completed the module, you will be able to demonstrate knowledge and understanding of:

- B1. Describe a number of modern software development methods, including the life cycle for developing secure software systems.
- B2. Select appropriate modern software development methods for a variety of software projects, taking into account assessment of risk.

Syllabus

- Subject Specific Practical Skills

Having successfully completed the module, you will be able to demonstrate knowledge and understanding of:

- D1. Implement security measures in server-side and client-side code
- D2. Evaluate the outcome of implementing security measures in server-side and client-side code

Indicative Content

Managing the software development process:

- Estimating the effort in software projects
- Contracts, planning and monitoring
- Costing and budgeting
- Models of Software Projects

Quality assurance:

- Concepts in QA
- Capability Maturity Modelling
- ISO 9000 standards
- Metrics
- Testing strategies
- Risk management
- Risk Based Software testing

Security by design

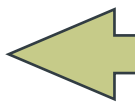
- Security models, and principles of secure computing
- Software Engineering methodology for secure systems
- Privacy and trust issues in software system design

Development methods:

- Iterative and incremental development
- Agile Development techniques
- Test-driven development
- Manual vs Automated Testing
- Refactoring
- Secure software design and development
- **Web-based Secure Coding**
 - General techniques for secure programming are covered using an example web development framework using e.g PHP, JavaScript or ASP.Net

Assessment:

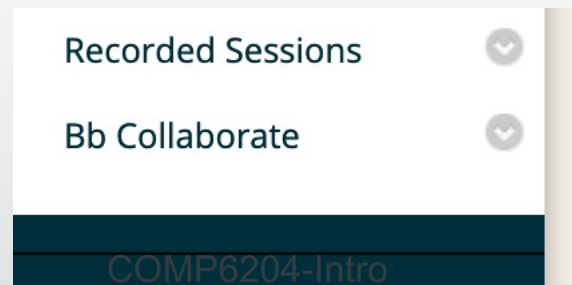
- Examination 70%
 - Closed book examination,
 - 2 hours, Section A and Section B
 - Section A - 11 short answer questions
 - Section B - Choose 2 questions from 3 questions
- Coursework 30%
 - Secure Laboratories using PHP
 - No marks for self-paced work, but everyone has to do it.



This was the case in previous years and it can be subject to change this year.

Resources 1:

- Module website:
 - <https://secure.ecs.soton.ac.uk/notes/comp6204/>
- Also where you can see any notification or changes to the module (so please look regularly)
- Will have the slides on the [NotesWiki](#) page
- We will have lecture videos on the Blackboard website
 - [Software Project Management and Secure Development COMP6204-32959-21-22](#)
 - You have to go to:



Resources 2:

- You can probably find everything you need on the Web, but I suggest you consider buying a book or two.
- A Software Engineering text
 - Any good one will do: Pressman, Sommerville, ...
- Some kind of Management Text
 - Management for Engineers (Chapman, Cooper, Page)...
- For aspects of Software Development,
 - Software Components: Guidelines and Applications by Muthu Ramachandran
- Don't forget we have a library which is a good source for *reference*
- Please note we must follow British copyright laws

Resources 3- an additional remark

- I will make my slides available to you on the the [NotesWiki](#) website but...
 - They are merely the slides I use for lectures
 - They do not amount to comprehensive notes about the course material
- Use the library properly
 - It is not a substitute for *buying books you need*

A comment about MSc study in the UK

- Study in the UK **differs** from some other parts of the world
- Students are expected to do more than **remember** what they have been told
- Don't be surprised by questions that ask for an **opinion**
 - Be sure always to give **supporting argument**
- **Assessments** – marks awarded are often lower than is normal elsewhere

Online session

- We will have one **online** session each week.
- You will be asked how you got on with the self-paced laboratories.
- Any questions about the self-passed slides that is not clear.
 - So make sure you have read the slide and watched the videos.
 - These are the topics that are on the [NotesWiki](#) page
- I will also be asking questions from the slides, some multiple choice and so more descriptive.
 - Similar to the exam questions

SOFTWARE ENGINEERING

Example: Underground System

- Design a system to monitor & control the trains on the central line.
- Issues:
 - Specification:
 - Objects/data structures
 - Routines/functions
 - Cost
 - Effort
 - Management

What is Software Engineering?

- The establishment and use of **sound engineering principles** (methods) in order to obtain economically viable software that is reliable and works on real machines.

(Bauer, F. L. Software Engineering. Information Processing 71., 1972).

- Software engineering.
 - (1) The application of a **systematic, disciplined, quantifiable** approach to the **development, operation, and maintenance** of software; that is, the application of engineering to software.
 - (2) The study of approaches as in (1)
 - (*IEEE Std 610-1990*).

What is Software Engineering?

- Software engineering is the technological and managerial discipline concerned with systematic production and maintenance of software products that are developed and modified on time and within cost estimates (Fairley, R. Software Engineering Concepts. New York: McGraw-Hill, 1985).
- Software engineering is the computer science discipline concerned with developing large applications. Software engineering covers not only the technical aspects of building software systems, but also management issues, such as directing programming teams, scheduling, and budgeting ([WebReference Webopaedia](#)).

What is Software Engineering?

- SEI software engineering definition from 1990 *SEI Report on Undergraduate Software Engineering Education* ([CMU/SEI-90-TR-003](#)):
 - Engineering is the **systematic** application of **scientific** knowledge in **creating** and **building cost-effective** solutions to practical problems in the service of mankind.
 - Software engineering is that form of engineering that applies the **principles of computer science and mathematics** to achieving **cost-effective** solutions to software problems.

What is Software Engineering?

- Engineering deals with the understanding, design and implementation of large, complex systems.
- Software engineering is concerned with theories, methods and tools for professional software development.
- Its not an exact science, rather a collection of ideas, techniques and tools which assist in the software development process.

What is it Different?

- Many of the techniques of **general product management** are applicable to the software domain.
- However, there are **certain characteristics** that **distinguish** them from others. These are [Brooks 87]
 - **Invisibilty**- when a physical artefact is being manufactured or constructed the progress can actually be seen. However with software, progress is not always immediately visible.
 - **Complexity** – as a general statement pre pound spent, software products are more complex than other engineering artefacts.
 - **Flexibility** – A major advantage of software is the ease with which it can be changed. Therefore the **general perception is that it is easier to accommodate changes** in software than to change the physical or organisational systems in which the software interfaces with.
 - Hence software systems are likely to be subject to a high degree of change.

What is it Different?

- Pressman points out that
 - Software, as yet can not be manufactured in the classical sense, but is developed or engineered.
 - Software does not wear out (eventually fail) like hardware, but ideally continues until it is obsolete.
 - Although in practice, due to modifications it may still experience failure over a period of time.
 - In addition, each failure in software shows an error in design or the implementation process.

What is it about?

- Group development
 - formalise overall project to all team members.
- Measure progress
 - progress can be monitored and re-scheduled.
- Identify subtasks
 - easier to complete, test and integrate several small tasks rather than a single large one.
- Cost-effective software development
 - identify cost, time v. features, reliability, ease of use trade-offs.

Types of Software

- **Generic products** (highest expenditure)
 - Stand-alone systems which are produced by a development organisation and sold on the open market to any customer.
 - Low risk, high expenditure, e.g. Microsoft Office.
- **Bespoke (customised) products**
 - Systems which are commissioned by a specific customer and developed specially by some contractor.
 - Highest effort and risk, e.g. Underground system.

Software Product Attributes

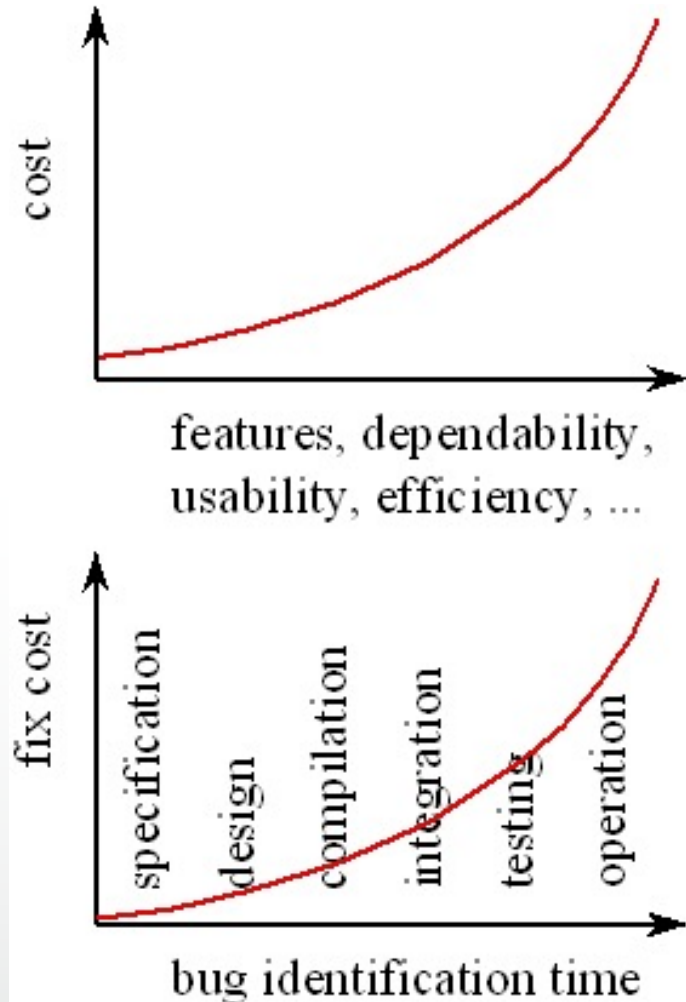
- Maintainability
 - to allow for evolution to meet changing specs.
- Dependability
 - failure should not result in physical or economic damage.
- Efficiency
 - system resources should be used efficiently.
 - Usability
 - Appropriate user interface and documentation should exist.

The Software Process

- Activities involved in the software process:
 - Specification.
 - Design (and coding).
 - Validation.
 - Documentation.
 - Evolution (and coding).
- Activities vary depending on the organisation and the type of system being developed.
 - Must be explicitly modelled if it is to be managed.

Development Costs

- Cost is a central theme of any engineering process.
- Determined by the software's complexity, reliability, ...
- Determined by “bugs” in the development process.



Why Secure software development

- The **target of attacks** has changed
 - Attackers traditionally, focus on operating system and network
 - Now the focus shifted to web applications, web browsers, mobile devices, embedded software
- The **attackers' nature** has changed
 - Traditionally, hackers are amateurs motivated by fun
 - Increasingly, hackers are professional organized crime and state-sponsored attackers

Why Secure software development

- Many attacks starts by **exploiting** a **vulnerability**
 - A security-relevant software **defect** that can be exploited to produce an **undesired behavior**
 - A software **defect** is present when the software **behaves incorrectly**
- Defects can be present in the software *design* and in its *implementation*
 - A **flaw** is a defect in the design
 - A **bug** is a defect in the implementation

Why Secure software development

- Examples of Software Flaws
- **Authentication** flaws
 - Authenticate an entity based on IP address or MAC address
 - Not automatic log out from a session
 - Not storing password encrypted or hashed
 - No limited life-time for authentication credentials
- Authorization flaws
 - Fail to revoke access/permission
- Incorrect use of cryptography
 - Rolling your own cryptographic algorithms or implementations
 - Misuse of libraries and algorithms
 - Poor key management
 - Randomness that is not random
- Examples of Software Bugs
 - Buffer overflow
 - SQL injection
 - Cross-site scripting
 - Cross-Site Request Forgery (CSRF)
 -

Summary

- Software engineer is the disciplined process of producing software
 - It helps communicates the design to members of the team
 - It helps manage the process through the life cycle: from concept to delivery and through to decommissioning

Questions -3 Mark questions

- What is Software Engineering
- How does it differ from other forms of engineering
- Some people say Software engineering is outdated, do you agree? Explain your answers