DEPARTMENT OF COMPUTER & INFORMATION SCIENCES

University of Strathclyde Science

Software Engineering

Lecture 1: Course Introduction and the Software Lifecycle

Billy Wallace
w.wallace@strath.ac.uk

CS993

Lecture Outline

- Course overview
  - How do you pass this course?
- What is software engineering?
- The software lifecycle

Course Overview

This will provide you with in-depth understanding of software development lifecycles, and development methods, tools and techniques, with particular attention to in-demand Agile methods. Upon completion you will:

- understand the key component involved in designing, building and testing a software system
- gain experience of the software development processes typically used within industry
- be familiar with the management tools and technologies used to support the development and management of software systems
- understand how to work as part of a team to design, build, test and deploy a software system

Timetable Info

- Tuesday at 1 – lectures (JA / 506)
- Wednesday 3-4pm – Invited speakers every 2nd week
- Thursday lab 1-3pm – credit bearing labs (LT / 1320)

University of Strathclyde Science

Lectures

- Lecture 1 - Introduction & the software lifecycle
- Lecture 2 - Project Management
- Lecture 3 - Configuration Management
- Lecture 4 - Requirements & Systems Analysis
- Lecture 5 - Design
- Lecture 6 - Development
- Lecture 7 - Testing
- Lecture 8 - Maintenance
- Lecture 9 - Models & Methods
- Lecture 10 - Quality

SWEBOK & SE2014

University of Strathclyde Science

- The Software Engineering Body of Knowledge defines the knowledge that a practitioner should have after four years of practice.
- This is reflected by the Software Engineering 2014 Curriculum Guidelines that defines the knowledge that an undergraduate software engineering student should possess upon graduation (after four years of study).
- This class tries to cover all of this in a single semester!
- Actually we use this as a framework to fit the various pieces together and give guidance to students as to where to go next.

SWEBOK Chapters

- Chapter 1 – Software Requirements
- Chapter 2 – Software Design
- Chapter 3 – Software Construction
- Chapter 4 – Software Testing
- Chapter 5 – Software Maintenance
- Chapter 6 – Software Configuration Management
- Chapter 7 – Software Engineering Management
- Chapter 8 – Software Engineering Process
- Chapter 9 – Software Engineering Models & Methods
- Chapter 10 – Software Quality

SWEBOK - Not Covered

- Chapter 11 – Software Engineering Professional Practice
- Chapter 12 – Software Engineering Economics
- Chapter 13 – Computing Foundations
- Chapter 14 – Mathematical Foundations
- Chapter 15 – Engineering Foundations

Invited Speakers

- Week 2 – John McGuire, Founder of Pulsion Technology.
- Week 4 – Allan Beck & Marnie McCormack, JP Morgan.
- Week 6 – Nadia McKay, Testing Services Director at EdgeTesting.
- Week 8 – TBA
- Week 10 – TBA

Assessment

Lab exercises (10 x 2%), group project report (30%) and individual project report (50%). Continuous assessment based upon a combination of written work, presentation work and project conduct. Assessment considers both the achievement of the group and the contribution of each individual within that group. The final delivery and presentation of the project takes place at the end of semester two.

Assessment is based upon the following key elements:

- Group Project
  - Report
  - Oral Presentation & Demonstration
  - Group Conduct & Progress
  - Peer Review and Individual Assessment
- Individual Project Report

Lab Exercises

- The purpose of these is generally to learn new tools that will be useful for you.
- You will receive credits for the exercises. You will blog about the exercise and send me a link to the blog post so that I can see that you completed it.
- I need to see the blog entries before midday the following Wednesday.

Group Project

- Software Engineering is all about working in groups, so this course would be pretty pointless if we didn't do that.
- Next week, John McGuire is going to give us the requirements for the group project.
- Before this, the groups should meet and talk about questions they want to ask.
- A general specification of the project is available.
- We don't cover requirements until week 4, so it might be worth having a scan at the slides for that before we meet John.

University of Strathclyde Science

Individual Project

- Write a report similar to the group project report talking about a software development project.
- Your individual project report should be written about your dissertation project. This should give you a head-start on that project.

Recommended Reading

- Software Engineering Body of Knowledge (SWEBOK)
  - https://www.computer.org/web/swebok/v3
- Beginning software engineering - Rod Stephens
  - Electronic version in library
- Software Engineering, Sommerville
  - Copies of 9th edition in library (outdated)
- Software Engineering, A Practitioner's Approach, Pressman
  - Copies of 6th edition in library (outdated)

Recommended Watching

- Selected videos developed for our Software Development Graduate Apprenticeship.
- Edx.org (https://www.edx.org/)
  - Software Engineering: Introduction
  - Software Engineering Essentials

What is software engineering?

University of Strathclyde Science

- According to Wikipedia, software engineering is the application of engineering to the development of software in a systematic method.
- Software systems are big, complex, expensive, important, etc. How do we actually build them?
- Programming is what one person does, software engineering is what teams do.
- Software engineering is a social activity, so it requires management.

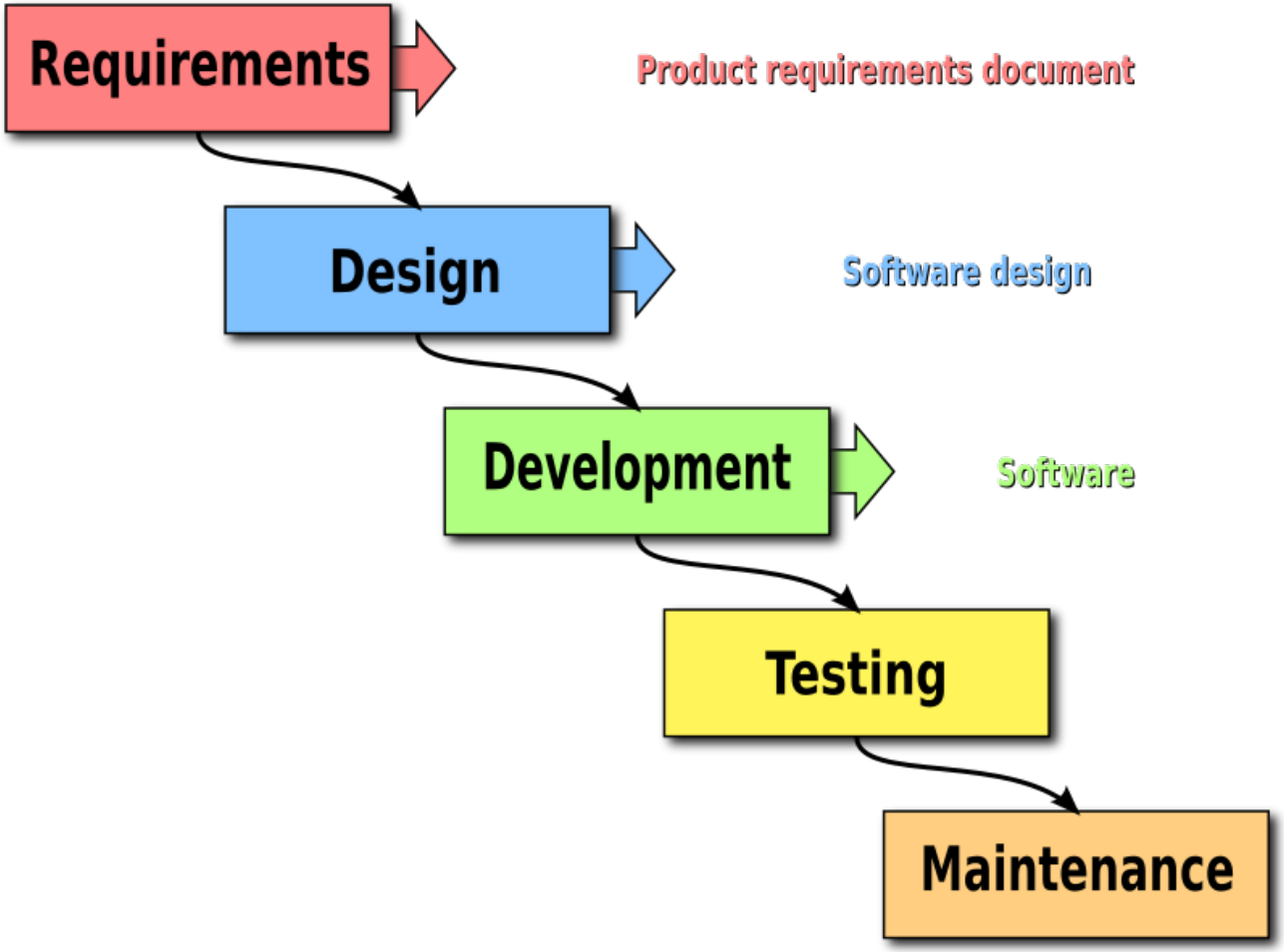
What is Software?

- Desktop application
- Mobile application
- Web application
- Mobile app with services in the cloud
- Application for sale to third parties
- Ditto including consulting to deploy
- Application for use in-house
- Library, kernel module, device driver, etc.
- Embedded software (IoT)

Life Cycles

- Software Development Life Cycle (SDLC)
  - Requirements
  - Design
  - Development
  - Testing
- Software Product Life Cycle (SPLC)
  - PLUS deployment, maintenance, support, etc.
  - As software engineers we need to be aware of all of the stages.

The Waterfall Model



University of Strathclyde Science

The Waterfall Model …

- … isn’t a model.
- It’s only here to help explain things.
- Don’t do it.
- It won’t work.

Police Scotland/Accenture

University of Strathclyde Science

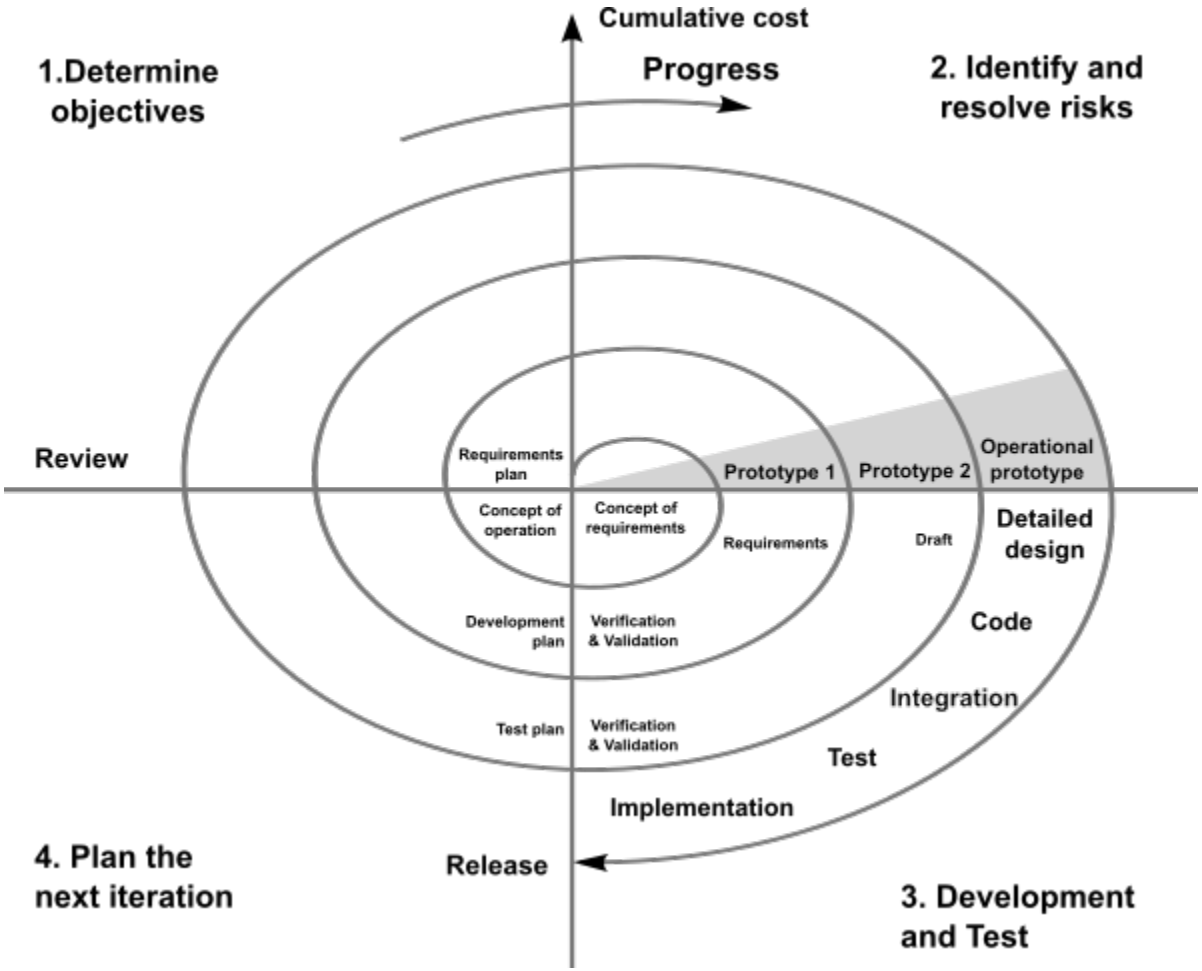
There was “no single reason” why the Police Scotland i6 computer system project failed, a report from Audit Scotland has concluded.
The report finds that good practice was followed in the planning and procurement stages of the ambitious project, which was intended to replace 130 electronic and paper-based systems covering 80 per cent of police processes for recording crime and missing persons.
Multinational professional services company Accenture was awarded a ten-year, £46.11m contract to provide the i6 computer system to Police Scotland by the Scottish Police Authority (SPA) in 2013, with projected savings of £200m to the authority and Police Scotland.
However, the “complex and ambitious programme” suffered early on as Police Scotland and the contractor disagreed over the terms of the contract, leading to “damaged relationships and trust between the two organisations from a very early stage.”
There was also an issue with the **waterfall approach** used to produce the software – common at the time the project was commissioned **???** – which meant that each stage was dependent on the previous one and the system could not be tested by police until it was almost complete. -- ***www.holyrood.com***

University of Strathclyde Science

Iterative Models

- What happens if we find a bug while testing?
- What if the bug is that it's too slow and a data structure chosen at design time doesn't scale?
- What if a user testing the UI says "this isn't what we need after all"?

Spiral Model (Boehm, 1988)



University of Strathclyde Science

Summary

- We need a “system” that will support the entire software development lifecycle in a flexible manner.
- Software development is naturally iterative and we should build that into the system.
- There’s no one right answer, and Agile lets us accommodate that by saying that it’s okay to try new things to improve processes.
- In my opinion, one of the strengths of Agile methodologies is the opportunity to build a culture where it’s easier for developers to make progress and do good work.

University of Strathclyde Science

Moving on from here …

- There's no invited speaker this week.
- The practical this week is us being incremental. We'll have a quick look at some useful tools that we'll cover in more depth later.
- We start to play with Agile methods and Scrum next week.

University of Strathclyde Glasgow