Software Engineering (IT): Course Introduction

Gül Calikli and Blair Archibald

Overview

- This course is all about *how* we build software
- This semester: the *processes* behind building software
  - Are we building the right thing?
  - How to manage a project
  - Agile development

Overview

- This course is all about *how* we build software
- This semester: the *processes* behind building software
  - Are we building the right thing?
  - How to manage a project
  - Agile development
- Next semester: how we *write* the software
  - What does good design look like?
  - What tools do we have for design/architecture?
  - How do we cope with complex systems?
  - What common patterns/architectures exist?

Lecturers

handangul.calikli@glasgow.ac.uk

- Empirical Software Engineering
- Human Aspects of Software Engineering
- Data Analytics for Software Engineering

blair.archibald@glasgow.ac.uk

- Formal Methods
- Graph Transformation
- Parallelism/High Performance Computing

Caveats

- There are **a lot** of ways to build software
- We will give an overview of techniques/tools found in practice
  - Some are more common than others
  - You should be aware of a range of techniques
    - To better integrate to different teams in future
  - Techniques may not necessarily *better* or more *correct* than another style
  - Tools to put a rover on Mars are going to be different from those to build a small app

Good Software Engineers are **flexible**

Course Structure

- 15 Credit Course
  - 1.5 times the size of what you are used to
- Taught over 2 semesters:
  - Semester 1 is **online** (*Live* Thursdays 9:00-11:00)
    - Software life-cycle/management/requirements capture
  - Semester 2 is **in-person**
    - Software design/clean code/architecture patterns

Course Structure

- 2h lecture each week
  - Online (Live) in first semester
- Materials/Recordings will be posted after the lecture
  - Please do try to attend live:
    - Having a fixed study structure is really useful!
    - You can ask questions
    - It's really boring for us to speak to an empty room!
- Labs starting in semester 2

Assessment

- Exam at the end of the course (70%)
- Assessed Exercises
  - *Team* based (like almost all software development)
  - AE1: Semester 1 (10%)
    - Produce a high level design for a system
  - AE2: Semester 2 (20%)
    - Implement (part of) your design (using good practice!)

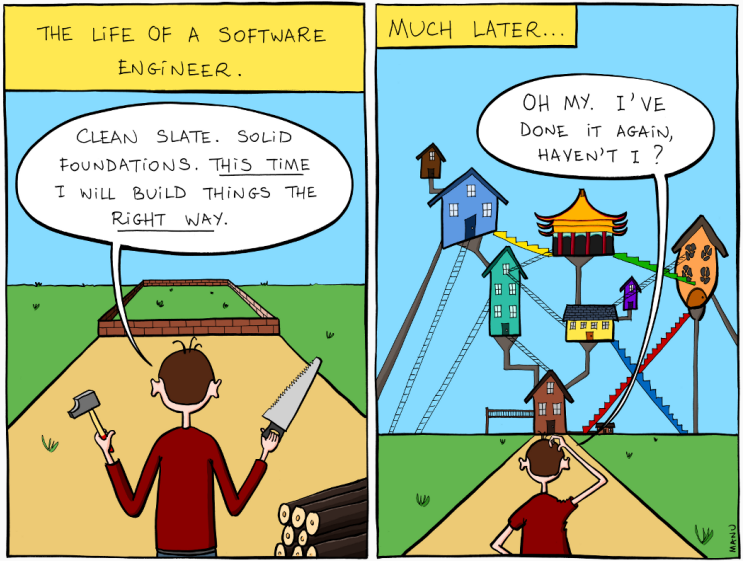
Tips to Do Well

- Attend the lectures and labs
  - Attendance and final course mark are **correlated**
- Attempt questions in lectures and labs
- Spend a *reasonable* time on the assessed exercises
  - Only 30% of the course, which itself is a smaller percentage of your degree
  - Don't spend ages at the detriment of other work

Tips to Do Well

- Seek out additional material for areas you are stuck
  - *Lots* of SE books (some suggestions:)
    - *Agile Project Management* - J Carroll & D Morris
    - *Head First Object-Oriented Analysis & Design* - B McLaughlin, G Pollice & D West.
    - *Head First Design Patterns* - E Freeman & E Robson
  - *Lots* of info on SE online
  - Speak with your peers
  - Ask for help if you need it
- Make sure to do well in your programming courses
  - We are going to see a lot of code in semester 2!

Lets Get Started



Q&A from Lecture

- *How many people per team*: 6/7
- *How do we choose teams*: choose own teams, or allocated if not possible
- *When does AE1 Begin*: exact date TBC, but probably week 3
  - Completed and marked this semester
- *When does AE2 Begin*: Second semester, exact date TBC
- *What programming language will we use*: Java
- *When is the exam*: April/May exam diet