Inf2: Software Engineering and Professional Practice

Lecture 1: Overview

Adriana Sejfia

School of Informatics
University of Edinburgh

Why take this course?

- **Software engineering is fascinating**
  - a blend of human and technical challenges
  - fast-moving
  - important
- **It could help you get a job!**
  - For many of you, it's one of the most job-relevant courses you'll take in your Informatics degree.

Course aims

- Give you an overview of what software engineering is
- Show you industry standard tools and techniques of the trade
- Discuss different processes for developing software
- Raise awareness of professional considerations surrounding software engineering
- Allow you to practice engineering software hands-on
- Encourage you to interact with technical documentation
- Develop your planning, organization, reflection, reviewing and teamwork skills
- Show you the reality of being a software engineer
- **Overall, take you beyond programming to engineering software**

Learning outcomes

- On completion of this course, you will be able to:

  1. Explain the modern techniques used in the design and development of large-scale software systems

  2. Apply, evaluate and reflect on these techniques in a small-scale, but realistic scenario

  3. Analyse the professional and ethical implications of software engineering decisions and propose solutions

  4. Comfortably read and write technical documentation

  5. Constructively engage in interaction with peers

Components of the course

- Two parts of the course:
- The software engineering (SE) part - focused on activities, processes, tools and techniques for engineering software.
- The professional practice (ProP) part - focused on larger professional considerations surrounding software engineering

- They go hand-in-hand, with ProP meant to complement SE.

Meet the lecturers!

Dr Adriana Sejfia

Course organiser

Lecturer on Software Engineering part

Dr Cristina Adriana Alexandru

Lecturer on SE and Professional Practice part

Lectures

- **All Tuesdays, Thursdays and Fridays @ 14:10-15:00**
  - 2 of them on SE (Adriana or Cristina)
  - One on ProP (Cristina) OR
  - One guest lecture from experienced industry professionals
- In person and live recorded
- Most are relevant for the coursework (including guest lectures!)
- Slides provided on OpenCourse page at least 24 hours prior to the class
- Essential/recommended resources included on slides, and already available on OpenCourse page under 'Weekly Reading'
- Self-study of course topics and Java essential!
- Unassessed Wooclap quizzes used during lectures to check understanding and engagement

Lectures - Code of Conduct

- Please arrive on time, or if a bit late avoid disrupting others
- Please keep quiet while lecturers present their slides
- Feel free to interrupt with questions by raising your hand; a microphone will be used
- Lecturers will also take breaks to ask for questions
- Please bring a device (mobile phone, laptop) to be able to answer to quizzes

Tutorials

- In Week 2 for ProP
- In Weeks 3, 5, 6, 8, 10 for SE
- Task sheets provided on Wednesday the week before, solutions provided Friday of the week
- Opportunity to practice concepts needed for the coursework
- Preparation before tutorials is desirable and expected by tutors
- Attendance will be taken

Labs

- **Mondays @13:30-15:00, Tuesdays, Thursdays @16:00-17:30 starting wk 2**
- Drop in as often as you like
- Focused on working on coursework (SE and ProP) and asking demonstrators questions about it
- See them as regular times to meet with your coursework team!
- Coming prepared with questions is desirable
- Other activities in labs:
  - Lab 1 and first lab for each new CW: on team development
  - During CW1: demonstrator interviews
  - During CW3: code reviews

Q&A Sessions

- Run:
  - During Tuesday lab on SE part (Adriana)
  - During Thursday lab (or same time online) on SE and ProP part (Cristina)
- Expectation for you to formulate questions

- Separate Q&A sessions can be organised if motivated by amount of questions in labs and Piazza

Formative assessment

- ProP formative coursework, starting in week 2
  - Individual work
  - Writing an argumentative essay on professional issues
  - Receiving feedback from:
    - Your peers, given guidance for high quality feedback
    - The markers

- Peer review system TBA

- Opportunities for bonus marks (/100% for course) available:
  - 2 for submitting an essay and two good peer reviews

Summative assessment (marks that count!)

- A software development project worth 100%, in 3 parts:
- CW1: Requirements engineering (SE part 1, worth 15%)
- CW2: Design (SE part 2, worth 22%)
- CW3: Construction, testing (63%: SE part 3 38%, ProP 25%)

- **SE parts done in your 4-student teams; practical work, justification, reflection/self-assessment.**

- **ProP part individual, consisting of the writing of an essay on professional issues surrounding the project.**

- Marking a mix of additive and criteria-based, with high-level marking scheme provided in advance.

Teamwork and marking fairness

- **Teamwork skills very sought after by the IT industry**

- We aim to help you develop them through:
  - Realistic situation: quasi-randomly assigned team for SE parts
  - Written guide with resources, guest lecture on teamwork
  - Team development activities in some of the labs
  - Realistic amount of support for team difficulties
  - Requirement to reflect on teamwork in CW1-3
- For marking fairness: declaration of work contributed by each team member in CW1-3, and marks adjusted accordingly

Sources of support

- **Labs, tutorials**
- **Piazza online discussion forum, handled Mondays, Wednesdays, Fridays @ 15:00-17:00. There is a confidential channel for lecturers.**
- **Email (asejfia@ed.ac.uk for SE questions, Cristina.Alexandru@ed.ac.uk for ProP questions)**

Resources for SE part

- Essential books (all provided for free electronically by library):

- **Sommerville, Software Engineering (10th Ed)**
  - Large, classic. Comprehensive on SE, but limited on UML and Java.
- **Sommerville, Engineering Software Products: An Introduction to Modern Software Engineering**
  - Excellent on agile processes, product engineering, modern architectures
- **Stevens with Pooley, Using UML (2nd Ed)**
  - Covers basic SE, does UML thoroughly.

- Other resources will be provided as external links, or recommended

Resources for ProP part

Key Professional Issues Texts (4)

BOOK **Professional issues in information technology**

Bott, Frank, British Computer Society,, Second edition.., Swindon, British Computer Society, 2014

Available at Main Library Main Library (HUB SHORT LOAN) - Ground floor : T58.5 Bot. and more locations Check availability >

BOOK **A Rulebook for Arguments.**

Weston, Anthony., 5th ed.., Cambridge, Hackett Publishing Company, Incorporated, 2018

Check availability >

WWW

WEBSITE **ACM Code of Ethics and Professional Conduct**

ACM, ACM, 22 June 2018

Check availability >

WWW

WEBSITE **BCS Code of Conduct**

BCS, BCS

Check availability >

Getting started in the course

- Check the course webpage on Learn for general course information, later assessment instructions and marks
- Check the OpenCourse webpage for course schedule, materials, readings, resources, tutorial solutions
- Go onto the Piazza forum for the course (linked from both websites), start interacting
- Check your allocated tutorial slot and fill in the time change form if its time does not work for you
- Check who your coursework teammates are and get in touch with them
- Start reading

Why is SE still hard?

- **Easy (or at least routine) projects**
  - small systems (up to c. 100k LOC),
  - without hard timescales or budgets,
  - without requirement for very high reliability
  - without complex interfaces or legacy requirements [...]

- **Hard projects**
  - everything else. Projects with all the above challenges, and more.

Statistics

- From The Standish Group CHAOS reports on software development projects of medium-large organisations.
- Surveying on average 50k projects each year.
- Projects categorised 3 ways:
  - **Succeeded: on time, on budget, with a satisfactory result**
  - **Challenged: delivered something but maybe late, over budget, not satisfactory to the customer or user**
  - **Failed: cancelled without delivering anything**

Standish CHAOS trends

TRADITIONAL RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	39%	37%	41%	36%	36%
CHALLENGED	39%	46%	40%	47%	45%
FAILED	22%	17%	19%	17%	19%

The Traditional resolution of all software projects from FY2011–2015 within the new CHAOS database.

- Taken from https://cdn1-public.infotech.com/agile/CHAOSReport2015-Final.pdf
- 2011-2015 latest report freely available, but similar in references to last report (2020): 31% successful, 50% challenged, 19% failed.

Reading

- **Recommended: Google CHAOS Standish reports:**
  - Standish Group: ENDLESS MODERNIZATION: How Infinite Flow Keeps Software Fresh
  - Standish Group: CHAOS Report 2015