Industry Mentor Scheme

Orientation

Suzanne M. Embury, Robert Haines
Duncan Hull, Caroline Jay, Markel Vigo

School of Computer Science
The University of Manchester
Manchester, UK
A bit out us

Suzanne Embury, Senior Lecturer (course leader)

Caroline Jay, Lecturer

Markel Vigo, Lecturer

Duncan Hull, Lecturer

Rob Haines, Software engineer
Our students

• ~200 in each academic year studying Computer Science

• Typical A-level offer is AAA (including Maths with significant Pure Maths component)

• Not necessarily any programming experience on arrival in year one
# Software Engineering at UG

<table>
<thead>
<tr>
<th>Year</th>
<th>Semester 1</th>
<th>Semester 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year 1</td>
<td>…</td>
<td>Java programming</td>
</tr>
<tr>
<td></td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td></td>
<td>…</td>
<td>Team project (web + DB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year 2</td>
<td></td>
<td>Year-long compulsory course unit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(software engineering)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year 3</td>
<td></td>
<td>3 elective course units</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(agile s/w engineering; user experience;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s/w engineering in a connected world)</td>
</tr>
</tbody>
</table>
Drivers for Change

- Industry Club input
- Current students’ feedback
- Industrial Placement Skills Audit
- Survey of Competitor Institutions
- Changes in industry practice
- Changes in tool sets

Software Engineering at UG
Skills Audit from Placement Students

Number of Students Reporting Using this Skill on Placement

- Software testing: 21
- Adding functionality to existing code base: 19
- Knowledge transfer/systant role: 18
- Working in distributed/overseas team: 17
- Maintaining an existing code base/support role: 16
- Database application development/SQL: 13
- Working to strict deadline/on mission critical task: 11
- C#: 11
- JavaScript: 10
- Writing technical documentation: 10
- Migrating/porting legacy code: 8
- Version/source control: 8
- Debugging code (written by self and others): 8
- Requirements gathering/interviewing: 8
- Building new tools/applications from scratch: 8
- Presenting ideas/demos to others, persuading people: 7
- Java: 7
- Data warehousing/ETL/data feeds: 7
- Autosys/scheduling of batch work: 7
- Test automation: 7
- Costing and pricing of work/estimating effort: 6
- Technical design/collaborative design: 6
- Reengineering/optimisation of code or processes or workflows: 6
- ASP/JSP: 6
- User acceptance testing/stress testing: 6
- Agile methods: 6
- Writing shell scripts: 6
Survey of Competitor Institutions

• Often not much public information

• Some examples of innovative modern practice
  – e.g. Sheffield, Royal Holloway, UCL

• Lots of UML
• Focus on OO programming/design

• Lack of coverage of modern technical practices
  – And some not so modern ones…
Software Engineering in the Lab

Typically:

• Small systems (few hundred lines)
• Fixed requirements
• Mandatory requirements
• Code built from scratch…
• … on top of well-behaved black-box components
• Testing considered optional

• Will never actually be used
Software Engineering in the Wild

• Typically
  – Very large systems
  – Uncertain and changing requirements
  – Need to make value judgments about what to deliver
  – Adding functionality to an existing code base
  – Using legacy components/data/technologies
  – Testing essential

– Must deliver value
– Must be maintainable in the long term
This Semester

Based around a large open source system

- Java
- Multi-user
- Multithreaded
- Client-server architecture
- TCP oriented network protocol
- mySQL and H2 persistence engines
- Open systems architecture designed for modification
- Complex business logic/business rules
Stendhal Multi-Player Adventure Game

https://stendhalgame.org/ Massively multiplayer online role-playing game (MMPORG)
Our Students Must

• Work in teams of 6 to:
  – Fix bugs
  – Add smallish new features
  – Re-architect a part of the system to add maintainability

• Use code quality and test coverage tools
• Use CI server
• Use a simple Git workflow
• Use code review techniques
• Choose a subset of the requirements
Teaching Methods

• One 2-hours workshop per week
  – Hands on

• 2 hours of scheduled “group working” time

• No face-to-face lectures
  – Selected lecture material provided as vodcasts for students to watch in their own time

• Each team has an industry mentor working with them through the year.

• (semester 2 only for now, semester 1 AND 2 later)
Why a mentoring scheme?

• Increase the industrial relevance of the course
• Provide students with an opportunity to discuss
  – The realities of being a software engineer
  – Typical tools and techniques used by software teams
  – The current jobs market
  – The range of career options available
  – How they might continue to succeed in their careers
  – The mentor’s own personal experiences
  – Anything you think we’ve missed!
• Tap into all your experience
What’s in it for you?

- Provide input into training the next generation
- Gain exposure to a cohort of potential recruits
  - Build relationships with students
  - See how teams and individuals work
  - Improve the quality of the entire cohort
  - Help embed good software engineering practices early
  - Advertise your company/jobs/graduate scheme
Dos and don’ts

• Please do
  – Talk to students about how they are approaching the lab coursework
  – Talk about how you might approach the coursework in industry
  – Talk about general processes and tools
  – Talk about your experiences and areas of interest

• Please don’t
  – Get hijacked into actually working on their lab coursework
  – Worry about “shy” students – we have ice breakers
  – Worry about the syllabus
How it will work

- [http://cs-mentoring.eventbrite.co.uk](http://cs-mentoring.eventbrite.co.uk)
  - ~40 mentors drop out?
  - first: week 4: greeted by ambassador
  - second: week 8: either here or on-site
  - third: (optional) week 11 showcase

- Mentoring guide
Who has signed up so far?

In no particular order:

- BBC, IBM, ARM Holdings
- Imagination Technology
- NCC Group, Avecto, Autotrader
- Data Centred, Barclays, RentalCars
- AppSense, CDL, Sage
- LateRooms, Web Applications UK, On The Beach
- (everyone here today)

- Northern Powerhouse / Southern Powerhouse?
Over to you

• Three break-out groups

• What you want from the mentoring scheme
  – Strengths
  – Weaknesses

• Any other issues we haven’t thought of or covered