Making a Year-long Software Engineering Project Agile

Leon Sterling
Alexi Lopez-Lorca
University of Melbourne, Australia

Acknowledgments

- (ex-) Faculty of ICT, Swinburne University of Technology
- Computing and Information Systems, University of Melbourne
- Software Engineering Project Teaching teams
  - – 2016: Leon, Alexi, Liz Haywood (5 projects)
  - – 2017: Leon, Alexi, Gil Tidhar, Doc Wallace (7 projects)
  - – 2018: Leon, Alexi, Gil, Liz, Eduardo Oliveira (9 projects)
- Craig Anslow

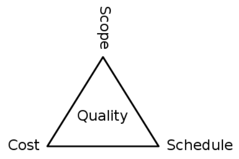
(Personal) History and Context

- 1990s Interest in Prolog and Software Engineering
- 1993 Software Engineering introduced at the University of Melbourne – 2 year-long projects
- 1995 I returned to Australia and Melb Uni
- 1996 Melb Uni degree first accredited SE degree by Engineers Australia
- 2001 Joint EA/ACS Board established
- 2002 Taught year long subject with 120 students
- 2010 I moved to Swinburne as Dean of Faculty of ICT
- 2016 Back responsible for Melbourne yearlong project with a mandate to make it agile

Year-long Software Engineering project

- Extremely valuable subject
- Requirement for Engineers Australia / Australian Computer Society accreditation for software engineering degrees
- Why?
  - – Contextualising subjects
  - – See the whole software lifecycle

Project Management Triangle



Waterfall: Scope fixed, cost and schedule can blow out

Agile: Cost and schedule fixed, scope can blow out

Student projects have fixed cost (student hours) and schedule (university semester) Hence should be agile!!

(Old) Process

- Produce documents: SQAP, SRS, SAD, SDD, code, test plan, final report
- Assess: 75% process, 25% product
- Appropriate for 1990s
- Hard to change once embedded

What needed to change

- Working code over documents: *assess code and other artifacts produced by agile process*
- Insist on professional tools
  - Don't get to choose in industry
- Grade per sprint – discourage hero efforts

Challenges

- Allow for individual contributions (somewhat debatable)
- Melbourne model (3+2) and flexible approach
- Many students take the requirements subject concurrently
- Cultural issues (2/3 students are from China)

Approach

- Assigned into 10 person teams by teaching team
- Two semesters of 12 weeks each
- Use of Atlassian tool set – Confluence, Jira, Bitbucket, Bamboo, HipChat
  - Continuous review of sites
- 6 week Project Inception
- Weekly stand up in class from Week 7
- 2 week 'trial sprint'
- 4 sprints of 4 weeks each
- Separate team meetings

Assessment

- 4 sprints of 4 weeks each each 15% 60%
- Presentations and videos 10%
- Individual contributions 30%
  - *Reflection 5%*
  - *Professional Development 5%*
  - *Individual contribution to team 20%*

Projects 2017

- Hospital Scheduling – client: Royal Melbourne Hospital
- TalkBit – client: Dept. Audiology and Speech Pathology
- Rehab Robot – client: Victorian Pediatric Rehab Service
- Teaching accounting – client: Dept Accounting
- Aboriginal Place Names – client: VACL
- 3D Visualisation – client: DST Group
- Beaton Gallery – client: Beaton Family
- *Can show videos if interested*

Project 'Highlights' and Learnings

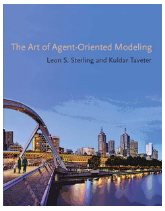
- Successful hospital delivery
- Rehab robot used in clinical practice
- NICA built its skill database

- IT support essential
- Managing difficult people is needed
- Creative projects risky
- Too much client knowledge potentially dangerous

Teaching team

- People and management skills more important than technical skills
- Reverse mentoring opportunities
- Networks very helpful
- Better to use casuals or those with industry interest

Innovations


The Art of Agent-Oriented Modeling
Leon S. Sterling and Kuldar Taveter

- Goal Models
- 2 minute videos
- Reports on guest lectures
- Assessing communication skills
  - Oral and written
  - Peer feedback
  - Chairing meetings

Conclusions

- Agile better suited for student projects
- Need to be flexible
- Real client projects motivate students and allow for avoiding the overspecifying of assessment
- Room for innovation

Thank you

- Any questions?