

SWT II 2017/18 Shared Retrospective

I like...

- Collaboration within teams
- Trying Scrum at the edge
- Students in PO and SM roles
- Guets lectures
- Meetings with customers
- Test-Driven Development (good practice for SWTI)
- Code Reviews
- SM looked after teams
- Swapping the SM role
- learnt how to do meetings (SM)
- rspec (pretty neat testing framework)
- Cucumber (have to understand story through writing a test)
- PO was design expert
- PO reviews before sprint end

I wish...

- to learn things and apply them in the real world
- reflect on the process
- there was another iteration
- POs had more meetings
- User Stories were independent
- the role of PO had switched
- Ruby on Rails was deprecated for something else
- More communication over team boundaries (architecture role?), knowledge transfer
- more dynamic programming, things take too long
- take location into account when forming teams
- teams did what tutors told them

I wonder...

- how to deal with async work
- how can a PO switch be implemented?
- how to introduce POs to the topic (tutorial?)
- how to make students pay attention to errbit
- how to introduce meeting organisation?
- how to find a technology that everyone knows?
- how to better explain RoR convention over configuration?
- whether we should use Java/PHP/Node
- how to introduce students to RoR/migrations?
- how to motivate students to assign themselves roles
- whether SWTII should be before SWTI
- how to make purpose of course more clear?
- how to ensure community ownership?
- how to motivate students to communicate?