Agile Software Development

L06 – Agile at Scale

Ezequiel Scott
ezequiel.scott@ut.ee
Agenda

• Recap

• Lessons learnt from Homework 1

• Exam warm-up

• Agile at Scale
Recap

✓ Software development processes, agile terminology
✓ Current state of Agile worldwide
✓ Test-driven Development (TDD)
✓ Handling requirements in ASD
  ✓ Writing user stories
✓ Organizing User Stories
✓ Refactoring and Code smells
✓ Scrum
Agenda

• Recap

• Lessons learnt from Homework 1

• Exam warm-up

• Agile at Scale
Homework 1: Yahtzee

Learning goal:
- To develop a scoring app for Yahtzee
- To practice TDD

The assignment:
• 5 parts describing different scoring rules
• 1 part related to testing
How?
Lesson learnt from homework 1

Possible strategies to apply TDD and show evidence about it (commits):

1) Following the assignment step by step (Part 1, 2, 3…)
2) Using a draft repo and copying the content to another
3) Fixing the commits messages
4) Other

What strategy did you use to solve the homework regarding the commits messages?

- I followed the homework step by step, sequentially: 88% (22)
- Two step approach: 0% (0)
- I "fixed" the commits as I was doing the homework: 0% (0)
- I completed a step and then committed test followed by solution: 12% (3)

Total Votes: 25
Notes about homework 1

Possible strategies to work in pairs:

1) One solves the problem, the other makes coffee

2) Sequential split:
   (one works on parts 1-3, the other works on parts 4-6)

3) Pair programming

4) Using different branches
   (per feature / per team)
Branch strategies

• There are several strategies for branches that address different needs

• Simple workflow:

  ![Diagram showing branch strategies](image)

  - Keep master green
  - Experiment on your feature branch
Another branch strategy

- Feature for future release
- Major feature for next release
- From this point on, “next release” means after 1.1
- Severe bugfix for production: hotfix 1.0.1
- Merge bugfix into dev
- Start of release branch for 1.1
- Bugfixes from pre-release should be continuously merged back into dev
- Tag 1.0
- Tag 1.0.1
- Tag 1.1
- Tag 1.2
Continuous Integration (CI)
Continuous Integration/Deployment (CI/CD)

1. compile
2. unit test
3. integration test
4. functional test
5. quality checks
6. package
7. deploy if green
8. rollback if red
9. notify

Triggered on commit/push

Deploy

Keep master green

Experiment on your feature branch
More branch strategies?

• There is no silver bullet but...
• We have:
  • Best practices
  • Patterns (proven solutions to recurring problems)
Agenda

• Recap
• Lessons learnt from Homework 1
• Exam warm-up
• Agile at Scale
Exam warm-up

• Kahoot!

• Exam-like questions

• Multiple choice: only one answer is correct

• Participation is optional

• Nicknames are allowed

• Price for the winner (16 euros)*

(*) this price is valid for eligible users. The price consists in notes and coins of chocolate which are printed as Euros. The sum of the printed labels (euros) is equivalent to sixteen euros. Claim your price in DELTA r3011.
Agenda

• Recap
• Lessons learnt from Homework 1
• Exam warm-up
• Agile at Scale
Agile at Scale

• Why scale?
  • We need to go faster?
  • Do we have a lot of people?
• We scale **to get more done in a given timeframe**
• It doesn’t mean to add more people necessarily
Team size is critical

• Research suggests 4.6 people is the “perfect team size” [1]
• Scrum guide suggests 3-9 people [2]
• Too small or large teams might struggle with the delivery of complex products
• The larger a team size, the greater **lines of communication** between team members, making it harder to create trust and a common purpose.

Several ways to scale...

SCALING METHODS AND APPROACHES
The Scaled Agile Framework® continues to be the most popular scaling method cited by respondents (35% this year compared to 30% last year). As a percentage of all responses, SAFe® outdistances the next nearest response, Scrum of Scrums, by 19%.
Scrum of Scrums (a.k.a. Meta-Scrum)
https://www.agilealliance.org/glossary/scrum-of-scrums/

(3) The Scrum of Scrums proceeds otherwise as a normal daily meeting, with ambassadors reporting completions, next steps and impediments on behalf of the teams they represent.

(2) Depending on the context, ambassadors may be technical contributors, or each team’s Scrum Master, or even managers of each team.

(1) Each daily scrum within a sub-team ends by designating one member as “ambassador”
Scaled Agile Framework (SAFe)

https://www.youtube.com/watch?v=RXzurBazN-I

**Team level:** team working under Scrum, Kanban, Scrumban

**Program level:** a team made of multiple teams working on Potentially Shippable Increments (PSIs)
- Agile Release Train (ART): A train containing features
- Release Train Engineering
- Program manager

**Portfolio management:** Allocate budget along investment themes that will be addressed by the Trains later.
Squads are cross-functional teams consisting of around 6–12 functional specialists from different teams. A group of teams organised into a department, usually working on a large feature. They are organised around a specialism, such as engineering or architecture.

Doesn’t it look similar to the other models? See https://bit.ly/2H1AFEV.
What’s next?

• Next 3 lectures/practices:
  • Web application development (Orlenys)

• Homework 2:
  • Will be published on 27.10

• Project release: 10.11
  • Project introduction (lecture)