Agile Software Development

L6 – Scrum (part II) and Agile at Scale

Ezequiel Scott
ezequiel.scott@ut.ee
Agenda

• Lessons learnt from Homework 1
• Recap
• Scrum
• Agile at Scale
• Exam warm-up
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 HW1 - strategies

There are different 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
Notes about HW1

There are different strategies to work in pairs:

1) Sequential split:
   (e.g., one works on parts 1-3, another works on parts 4-6)

2) Pair programming

3) 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]

- Keep master green
- Experiment on your feature branch
Continuous Integration (CI)
Continuous Integration/Deployment (CI/CD)

Triggered on commit/push

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

Keep master green

Experiment on your feature branch

Deploy
Continuous Integration/Deployment (CI/CD)

Triggered on commit/push

1. compile
2. unit test
3. integration test
4. functional test
5. quality checks
6. package

DevOps: Automating Software Delivery and Operations (LTAT.06.015)
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
More branch strategies?

• There is no silver bullet but...

• We have:
  • Best practices
  • Patterns (proven solutions to recurring problems)
Agenda

• Lessons learnt from Homework 1
• Recap
• Scrum
• Agile at Scale
• Exam warm-up
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 (part I)
Agenda

• Lessons learnt from Homework 1
• Recap
• **Scrum**
• Agile at Scale
• Exam warm-up
The Scrum framework

1. Organizing and Preparing User Stories
   - As an User
     - I want
2. Planning the Sprint Backlog
   - Sprint Backlog
   - Planning Poker
3. Closing the Sprint
   - Feedback
   - Self-reflection
   - Celebration
   - Improvements
4. Controlling and Monitoring Sprint Work
   - Product Increment
   - Product Integration
5. Delivering the Product

Roles:
- Scrum Master & Scrum Team
- Agile Coach
- Product Owner
The Scrum framework

- **Planning the Sprint Backlog**
  - Sprint Backlog
  - Planning Poker

- **Organizing and Preparing User Stories**
  - Building of Product Backlog
  - Configuration of development environment
  - Distribution of workstations

- **Closing the Sprint**
  - Feedback
  - Self-reflection
  - Celebration
  - Improvements

- **Controlling and Monitoring Sprint Work**
  - Product Increment
  - Product Integration

- **Daily Scrum 24 HS**

---

**Product Owner**
- As an User I want
  - As an User I want
  - As an User I want

**Scrum Master & Scrum Team**
- Agile Coach
Building a Release Plan

The allocation of User Stories to iterations is done according to **story points**, **velocity** and **sprint goals**.

Example:
16 user stories
100 story points in total
Building a Release Plan

Example:
16 user stories, 100 story points in total

If our velocity is 20 story points per iteration, we can expect it to take 5 iterations. Each iteration (sprint) should have a sprint goal.

The allocation of User Stories to iterations is done according to **story points, velocity** and **sprint goals**
The initial velocity

There are three ways to get an initial value for velocity (Cohn, 2004):

1. **Using historical values**, but they are not always available

2. **Run an initial iteration** and use the velocity of that iteration, but we might not have the time to do it

3. **Take a guess**, but it might be inaccurate (making simple assumptions can help, e.g., 1 story point = 1 ideal day of work)
The Scrum framework

- **Organizing and Preparing User Stories**
  - As an User
  - I want
  - As an User
  - I want

- **Planning the Sprint Backlog**
  - Sprint Backlog
  - Planning Poker

- **Closing the Sprint**
  - Feedback
  - Self-reflection
  - Celebration
  - Improvements

- **Controlling and Monitoring Sprint Work**
  - Product Increment
  - Product Integration

- **Daily Scrum 24 HS**

- **MTAT.03.295** | Agile Software Development | © Ezequiel Scott 2021

- UNIVERSITY OF TARTU
## Daily Scrum

(using Kanban boards → ScrumBan?)

<table>
<thead>
<tr>
<th>Story</th>
<th>To Do</th>
<th>In Process</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>As a user, I...</td>
<td>Code the...</td>
<td>Figure out how...</td>
<td>8 hrs</td>
</tr>
<tr>
<td></td>
<td>Design a...</td>
<td>8 hrs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test the...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Figure out how...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>As a novice user, I...</td>
<td>Code the...</td>
<td>Figure out how...</td>
<td>8 hrs</td>
</tr>
<tr>
<td></td>
<td>Design a...</td>
<td>SC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test the...</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MTAT.03.295 | Agile Software Development | © Ezequiel Scott 2021
Daily Scrum

* Work to complete the sprint backlog
• All members **self-assign** at least one task
• Produce high quality code
• Document as necessary
• Keep in mind the **Sprint Goal**
• The team works to achieve a **product increment**
Monitoring Progress Toward Goals

• **At any point in time**, the total work remaining to reach a goal can be summed.

• Various **projective practices** upon trending have been used to forecast progress, like burn-downs, burn-ups, or cumulative flows. These have proven useful. However, these do not replace the importance of empiricism.
The Burndown chart

Can be used at any point in time:
- To show daily progress
- To show project progress

During the first iteration, 20 story points were burnt.

Figure 11.4 Burndown chart for the project in Table 11.2.


Figure 11.5 A daily burndown chart.
Burndown Chart (cont.)
Daily Scrum

- On **each day of a sprint**, the team holds a daily scrum meeting.
- They are strictly **time-boxed** to 15 minutes.
- All team members are required to attend scrum meetings.

**Daily meeting**

- What did you do yesterday?
- What will you do today?
- Are there any impediments in your way?

[https://www.agilealliance.org/daily-scrum-is-it-a-waste-of-time/](https://www.agilealliance.org/daily-scrum-is-it-a-waste-of-time/)
Daily Scrum

• On **each day of a sprint**, the team holds a daily scrum meeting

• They are strictly **time-boxed** to 15 minutes

• All team members are required to attend scrum meetings

• **Some improvements** → (Downey and Sutherland, 2013)

Daily meeting

- What did **WE** achieve yesterday on **Priority 1**?
- What was **OUR** contribution on **Priority 1** worth in Story Points?
- What is **OUR** plan for completing **Priority 1** today?
- What, if anything, is blocking **US** or has the potential to slow **US** down today?

The Scrum framework

- **Product Increment**
- **Product Integration**
- **Feedback**
- **Self-reflection**
- **Celebration**
- **Improvements**

**Closing the Sprint**
- Building of Product Backlog
- Configuration of development environment
- Distribution of workstations

**Controlling and Monitoring Sprint Work**

- **Daily Scrum 24 HS**

**Planning the Sprint Backlog**
- Sprint Backlog
- Planning Poker

**Organizing and Preparing User Stories**

- **Product Owner**
  - As an User I want ---
  - As an User I want ---
  - As an User I want ---

**Delivering the Product**

- **Scrum Master**
- **Scrum Team**
- **Agile Coach**
Sprint Review

• **Goal**: to inspect the *outcome* of the Sprint and determine future adaptations

• Prepare a demo to validate the customer’s User Stories

• Ensure that the product works under acceptable conditions

• Receive and leverage the *feedback* on the increment

• **Timebox**: 4 hours max for a 1-month Sprint
Sprint Retrospective

• Goal: to increase quality and effectiveness
• Each team holds a private meeting
• Reflect on what happened along the sprint
• Each team member is asked to identify specific things that the team should:
  ✓ Start doing (AN IMPROVEMENT)
  ✓ Stop doing (THE BAD)
  ✓ Continue doing (THE GOOD)

• The result is at least one action item included on the following Sprint’s Sprint Backlog
• Timebox: max 3 hours for 1-month sprint
Agenda

• Lessons learnt from Homework 1
• Recap
• Scrum
• **Agile at Scale**
• Exam warm-up
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** (\(C\)) between team members, making it harder to create **trust** and a **common purpose**

\[ C = \frac{n(n-1)}{2} \]

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%.
Agile at scale research

Figure 2: Temporal distribution of the primary studies by method category

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.

They are organised around a specialism, such as engineering or architecture.

A group of teams organised into a department, usually working on a large feature.

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

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

- Kahoot!
- Exam-like questions
- Multiple choice: only one answer is correct
- Participation is optional
- Nicknames are allowed
What’s next?

• Next 2 lectures/practices:
  • Web application development (Orlenys)
• Guest lecture: 02.11 10:15-12:00 NEW!!
  • Tankut Senturk and Mikk Kard from Heathmont Group
• Homework 2:
  • Release: 19.10 / Deadline: 29.10
• Project release: 02.11
  • Project introduction (12:15-14:00)