

# CS-C2130 / CS-C2140 / CS-E4910 Software Project 1 / 2 / 3

## Experience Exchange Session 1 (EES 1)

6.11.2024

Jari Vanhanen

# Agenda

1. Course Practicalities
2. Experiences from a former Scrum Master, Esa Vikberg
3. Brief introduction of the status of the projects (1 min per team)
4. Discussions about the proposed topics in small groups

# Course practicalities

- Signing the project contract, **DL 6.11**.
- Scrum Master's credits: 5-8cr
  - update your planned number of credits to the Student List on the Google sheet
- Peer teams for peer testing are listed on the [Projects page](#)
  - 8h peer testing during the latter half of the project
  - other forms of collaboration also recommended

# Project Review 1 (PR1)

Read more from the [Project Reviews page](#).  
(agenda, content of presentation, required materials)

- Schedule published on the CS-C2130 Google sheet
- Submit the required materials to all stakeholders 24h before PR1
  - send a weblink by e-mail
    - online folder or a real web page for the project
  - the links will be published in MyCourses
    - in order to share the materials among all teams, all clients, quality award jury, ...
  - don't add materials that may contain confidential information to the web page
- Learning diaries are submitted individually to MyCourses
  - 1) educational observations, 2) own contribution to the project
- Use some student's laptop for the presentation
  - and for Zoom, if needed
    - Zoom-link created by the course personnel, if needed

# Experience Exchange Sessions (EESs)

- Sharing experiences and tips among the teams
- Hearing about the status of the other projects
- Typical agenda
  - small pre-task
  - possibly an intro by an expert
  - status of the projects
  - discussions
- 0.5p to the participating teams
- **8.11. Setting-up the project, adopting Scrum**
  - for scrum masters and developers
  - guest Esa Vikberg
- **22.1. Testing and quality assurance**
  - for developers and scrum masters
  - guest Antti Ahonen / Tech Excellence Oy
- **5.2. Technology architecture**
  - for developers and scrum masters
  - guest Jarno Hilvenius / Accenture
- **28.4. What did we learn?**
  - for scrum masters and developers

# Community of Practice (CoP)

- CoP is a *group of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly* [1]
- CoP provides a way for practitioners to **share tips and best practices**, **ask questions** of their colleagues, and **provide support** for each other.
- Used in many organizations
  - recommended in agile scaling frameworks (SAFe, LeSS)
  - software project course at IT University of Copenhagen [2]

[1] Etienne and Beverly Wenger-Trayner. "[Introduction to communities of practice - A brief overview of the concept and its uses](#)". 2015.

[2] Maria Paasivaara: "[Teaching the Scrum Master Role using Professional Agile Coaches and Communities of Practice](#)". ICSE (SEET), 2021.

# CoP for the Scrum Masters on the Software Project 3 course

- Why
  - you will learn more
  - the projects will be more successful
- How
  - Online chat
  - Online document for collecting best tips and tools
  - Meetings
    - even with a small number of Scrum Masters
    - informal discussion while having coffee / lunch together
    - more formal meetings with an agenda
      - discussing some agreed topic(s)
      - discussing some materials everyone reads in advance
      - presentations/demos by some participants

Any volunteers who could organize the CoP (with the help from the course personnel)?

# Status of the projects

- Who is your client and what is the main goal of the project?
- First impressions of the project, e.g.
  - main "achievements" so far
  - main strengths of your team
  - the biggest challenge of the project, any major problems?
  
- 1 minute per team



# Discussion rounds

- Parallel discussions in tables: A, B, C, D
  - when your topic is assigned to a table for the current round, you must go to that table
  - otherwise, you may choose any table
- Student, who proposed the topic
  - should introduce the topic
    - and then everyone around the table can share their thoughts
  - write down 1-3 highlights/best tips etc. from the discussion
    - present them to the other groups after the discussion, or write them to the Google sheet

# Learn to fail, fail to learn

What went well, what didn't go well, what can be improved?

# Who am I?

- Software Project 1 & 2 – 2017-2018
  - Scheduling tool for Startup Sauna
- Software Project 3 – 2022-2023
  - Year on the Rails for VR
- Coach – 2023->
  - 2023: AR application for Beamex, Github scanner for Valo security
  - 2024: Analog gauge reader for Beamex
- PhD studies at Aalto 2022-2024
  - Extended Reality, VR Sauna
- HW/SW Engineer at Elisa 2024->
  - Distributed Energy Storage

# Learn to fail

- Previous year EES voted retrospective as the most valuable scrum event
- If you see something is not working, write it down, bring it up
- Try things for a sprint, see if it works
  - Fail quick
- Communicate transparently
- Don't take criticism personally, don't defend mistakes

# Fail to learn

- I have done the project 2 times and failed in some ways. I look forward to failing in new ways
- We remember our own failures decades later
- It's much easier, when choosing methods, to rule out some, based on prior experience
  - What will work then? Decide as a team. Try it for a sprint and evaluate.



Some good failures

# Corporations are slow

- Start of project, getting the contracts signed can take ages. How fast you got it can be an indication of how fast you can expect other things.
  - **Don't count on it though**
- If you notice any dependencies that are required from the client side, bring them to the attention of your PO yesterday.
  - Devices, software, accounts.
- You also can't expect your PO to react to things as fast as you do.
  - PO will not have an emergency meeting with you at 8PM on a Friday.
  - Company IT certainly won't fix your access rights on the weekend when you had planned to work on the project.

# Distant Scrum Master

- Developers look up to the scrum master as the person to go to with questions.
- PO looks to the scrum master as the member of the team with an idea of the big picture.
- During SP<sub>1&2</sub>, our scrum master didn't take part in joint work sessions, nor even the final project review.
- For both team spirit and keeping track of what's going on, it's good for Scrum Master to be there, when work is happening.



# Workload estimation

- Constantly overpromising and underdelivering
  - Makes your team look unprofessional, disappoints PO.
- Constantly under-promising and running out of things to do
  - Very frustrating for the team. Requires asking more things to do mid-sprint, which Scrum doesn't love.
- What somewhat works:
  - Prioritize the things in the backlog.
  - Make the sprint goal achievable and take in the things needed for it.
  - Taking in a little bit extra, or preparing for next sprint, makes sure you don't run out.
  - Communicate with PO during the sprint.

## Communication with team

- “I think that you guys not telling us in anyway and just going directly to the coach and [PO] about us basically sacrificing (parts of) our Christmas break is in the category of assholery. I know that we are lacking time from sprint 0 but you should at least have talked to us first about how we are planning to make up that time. I know that you said when [dev] started to push back that we wouldn't be required to work during the break but mentioning that then doesn't change fact that you did all this shit without once saying anything to us. You already kind of it seems promised [PO] that we would do stuff on the break.”

# Work practices

- “For me the way we are currently doing stuff does not work. All these meetings planned for the same day cause me huge amounts of stress and makes it really hard for me to get anything done in my own time - both for this course and for others. When my plan is disrupted like at such short notice that it is quite hard for me to adapt it immediately. I would suggest we instead plan more recurring working times and such that would always be the same time every week.”

# Reporting to the coach

- Coaches grade you on how you deal with failures
- Recognizing problems gets you points
- Finding solutions gets you points
- Implementing solutions gets you points
- Not telling the coach about it in any way loses you all those points
  - Mention it in the project reviews.
  - It will also remind the PO of the stuff you've gone through.

Failures are  
expected

- This is a course, you are not paid developers, you are here to learn.