

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

Experience Exchange Session 1 (EES 1)

1.11.2023

Jari Vanhanen

Agenda

1. **Course Practicalities**
2. Community of Practice for the Scrum Masters
3. Brief introduction of the status of the projects (1-2 min per team)
4. Discussions about the proposed topics in small groups
5. Tips from previous Scrum Masters and developers

Course Practicalities

- Project Review 1 schedule limitations, **DL 6.11.**
 - Signing the project contract, **DL 6.11.** (recommended)

 - Scrum Master's credits: 5-8cr
 - update your planned number of credits to the Student List on the Google sheet

 - Peer teams have been assigned (see the Projects page)
 - 8h peer testing during the latter half of the project
 - other forms of collaboration also allowed and recommended

 - Public materials
 - in order to share materials among the teams and other stakeholders, public materials are submitted to the course as a web link that will be published in MyCourses
 - minimum requirement is an online folder, but you may create a real web page for the project as well
 - submit materials that may contain confidential information as e-mail attachments
-

Agenda

1. Course Practicalities
- 2. Community of Practice for the Scrum Masters**
3. Brief introduction of the status of the projects (1-2 min per team)
4. Discussions about the proposed topics in small groups
5. Tips from previous 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.

Possible Topics to Discuss among the Scrum Masters [2]

- Challenges faced
 - sharing how other Scrum Masters have worked in a similar situation
- Scrum Master's role and responsibilities
- Understanding Scrum
- Team building activities
- Communication
- Tools
- Collaboration with the PO and coaches
- Course requirements
- ...

[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
 - F-2-f or Zoom 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 volunteer(s)?

EESs

- **15.11. 16-18, no EES**
 - but T3 can be used by the scrum masters
- **22.11. 16-18, EES on Testing**
 - topics
 - practical testing strategies
 - static code checks
 - code reviews
 - continuous integration and quality gates
 - for developers and scrum masters
 - guest Antti Ahonen
 - registration and discussion topics by 21.11. 13:00
 - max. 3 persons/team
- **January, EES on Technology architecture**

Agenda

1. Course Practicalities
2. Community of Practice for the Scrum Masters
- 3. Brief introduction of the status of the projects (1-2 min per team)**
4. Discussions about the proposed topics in small groups
5. Tips from previous Scrum Masters and developers

Brief introduction of the status of the projects

- 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?

- 1-2 minutes per team

Agenda

1. Course Practicalities
2. Community of Practice for the Scrum Masters
3. Brief introduction of the status of the projects (1-2 min per team)
- 4. Discussions about the proposed topics in small groups**
5. Tips from previous Scrum Masters and developers

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

Agenda

1. Course Practicalities
2. Community of Practice for the Scrum Masters
3. Brief introduction of the status of the projects (1-2 min per team)
4. Discussions about the proposed topics in small groups
- 5. Tips from previous Scrum Masters and developers**

Tips for the Scrum Masters

- Get the project started
 - It is imperative to organize a kick-off meeting and start working as soon as possible.
 - Make sure the team is working the required hours from the beginning of the project.
 - Keep track of the working hours.
- Communication with the team will naturally get better with time
 - However, one can put a bit more effort at the beginning and help the team members to open up.
- Keep yourself motivated. Motivation spreads throughout people
- Ensure that the team achieves the Sprint goals
 - It will foster motivation and team cohesion.
- Do not steal the spotlight. Let the developers be in front of the project
- Plan internal demos
 - It will help the developers to be motivated and get constant feedback from their peers.

Based on the interviews of the Scrum Masters on the SW Project 3 course by Jhosimar Aguacia Fisco in his MSc Thesis
https://aaltodoc.aalto.fi/bitstream/handle/123456789/42684/master_Aguacia_Fisc%C3%B3_Jhosimar_2020.pdf?sequence=1&isAllowed=y

The most common problems in our course projects

most common != most serious

Problem	Developers median (N=88)	Managers median (N=26)
Tech. skills - The team members were inexperienced with the implementation technologies	5,0	6,0
Testing - The developers took the testing tasks less seriously than coding tasks	5,0	5,0
Testing - The amount of testing was lower than planned	5,0	4,5
Testing/Quality requirements - Converting quality requirements into concrete tasks was difficult	4,0	5,0
Estimation - Implementing tasks with the desired quality level required more effort than estimated	4,0	4,5
Estimation - Estimates for tasks that required learning activities were poor	4,0	4,5
Testing - Selecting the testing tools and practices was difficult	4,0	3,5
Estimation - Effort estimation was considered as an unhelpful activity	4,0	3,0
Testing - The requirements were specified on too general a level for supporting testing	3,5	2,5
Motivation - Paid work overrode the course project	3,0	5,0
Testing - For a long time, the system was too unfinished for testing	3,0	4,0
Task management - Started tasks remained uncompleted	3,0	4,0
Task management - The tasks were planned on too general a level	3,0	4,0

Scale: "not at all" (1) – "very much" (7)

Vanhanen, J., et al., [Software engineering problems and their relationship to perceived learning and customer satisfaction on a software capstone project](#). *Journal of Systems and Software* Vol. 137, pp. 50-66, 2018.

Vanhanen, J. at al., Software Engineering Problems Encountered by Capstone Project Teams, *International Journal of Engineering Education* 30(6(A)), pp. 1461-1475, 2014.