Aalto University

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

Lecture 2: Scrum Basics and Applying it in the Course Project

Jari Vanhanen

11.9.2024

Agenda

- **Next steps on the course**

- Scrum basics and applying Scrum on this course
- Additional requirements for the course projects

- Accenture Quality Award, Jarno Hilvenius/Accenture

Next Steps (1/2)

- Team formation
  - **We 11.9. Scrum Masters ensure that Teams-sheet info is up-to-date**
    - Core team's common preferences of worktimes, topics etc.
    - Note that topics H, I, J, P, Q, U were pre-agreed (already assigned to certain teams)
  - **Th 12.9. Free developers list their Preferred teams**
    - Based on the info on the Teams-sheet
  - **Fr 13.9. Teacher assigns the free developers to the teams**
    - Set-up the team's communication channel(s)
    - The whole team should meet, and discuss once more about the preferred topics
- Scrum Simulation
  - **Register ASAP** (even before Friday)
    - List also food allergies and dietary restrictions
  - 5 teams are needed already to the first session on Mo 23.9. 16-20.

Next Steps (2/2)

- Trainings for the Scrum Masters
  - **Th-Fr 12.-13.9. 8:00 - 16** CSM Training
    - Remember the pre-task
  - **We 18.9. 16:15 - 18** Scrum Master Lecture
    - Scrum Master's role in the course project and in the Scrum Simulation

- Choosing the project topic
  - **In September**, decide team's favorite topics and prepare a team "CV"
  - **1.-6.10.** Send the Team "CV" to 3-5 Clients
    - Include at least one less popular topic that you will "certainly" get
    - You can mention to one Client that they are your first option
    - Keep the *Team's favorite topics* column on the Teams-sheet up-to-date
    - Reserve 10-minute meetings on the course's GoogleSheet
  - **We 9.10. 16:15 - ~18:30** Meetings with the Clients
    - Finalize your *Team's favorite topics* column on the Teams-sheet

Agenda

- Next steps on the course

- **Scrum basics and applying Scrum on this course**
- Additional requirements for the course projects

- Accenture Quality Award, Jarno Hilvenius/Accenture

Goals of This Lecture

- Teach you the basics of Scrum
  - Roles
  - Scrum Events and Artifacts
  - Terminology

- After this lecture
  - You know the basics of Scrum and how it can/should be applied in the course project
  - You are able to participate in the Scrum Simulation in the developer role

- This lecture is based on
  - Scrum Guide 2020 and Scrum Primer
  - CS-C2130 Project Manual

Scrum

- Iterative and incremental agile software development framework for **managing** product development

Does not cover design, implementation, or concrete testing practices

- Process framework
  - not a process, technique, or method
  - every team must decide the specific tactics for using Scrum

Why Process?

- A process defines how an organization, in your case, a team works together to achieve its goals

- Why do you think having an understood/agreed upon process is or is not beneficial?

Scrum Roles

- Scrum Team
  - Product Owner (PO)
  - Developers
  - Scrum Master

Max. 10 people

Product Owner (PO)

- Accountable for maximizing the value of the product
- Identifies product features
- Prioritizes the features
- Interacts regularly with the developers
  - e.g. planning the Sprints, reviewing the Sprint results
- One person
- May delegate some work to the developers, but remains accountable

In the course projects some POs may expect quite a lot innovation also from the developers

The Developers

- Develop the product and provide ideas to the PO about how to make the product great
- Cross-functional team
  - together they have all expertise necessary to deliver a potentially shippable product each sprint
- Are self-managing: high degree of autonomy and accountability
- Every developer is just a developer, no other roles

Scrum Master

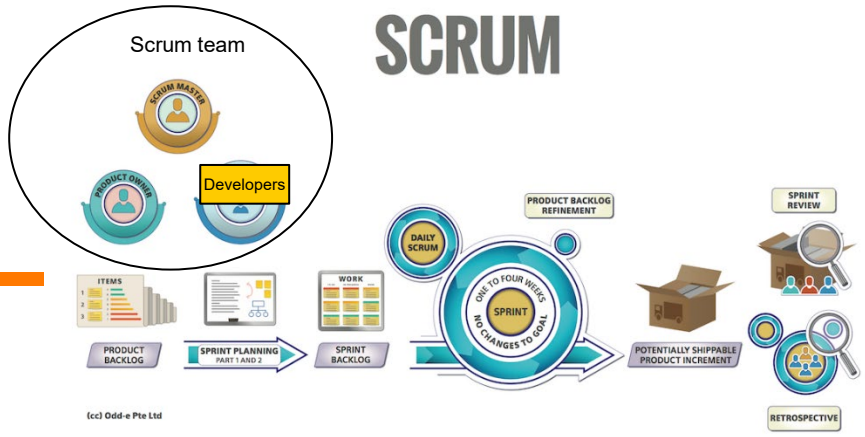
- Helps the Scrum Team apply Scrum effectively
  - positive, productive and time-boxed Scrum events
- Is NOT the manager of the team members, NOR a project manager OR team lead
- Is a coach and teacher, especially Scrum principles and practices
- Serves the team, e.g. helps to remove impediments, protects from outside interference, helps to adopt good work practices

Scrum Events and Artifacts

- Events
  - Sprint
    - Sprint Planning
    - Daily Scrum
    - Sprint Review
    - Sprint Retrospective
- Artifacts and commitments
  - Product Backlog
    - Product Goal
  - Sprint backlog
    - Sprint Goal
  - Increment
    - Definition of Done (DoD)

Sprint

- Time-boxed development cycles of 1-4 weeks
- Never extended: ends exactly when planned, contents give flexibility
- The output of every sprint is: “Potentially Shippable Product Increment”, which means that item chosen for that sprint are “Done” (according to the Definition of Done)
  - System is integrated
  - Fully tested
  - End-user documented
  - Potentially shippable



Sprints (CS-C2130)

- At least six Sprints
  - 225h / 6 = 37.5h (per student)
- Plan in the beginning of the project
  - start and end dates of all Sprints
  - effort allocation per person per Sprint
- First Sprint (“Sprint 0”) and last Sprint differ from the normal Sprints
  - contain some tasks defined by the course

Blue text = requirements set by the course (not part of Scrum)

Product Backlog

- Ordered list of what is needed to improve the product
- Includes “items”, e.g. new sw features, major engineering improvement goals, research work, (known defects)
  - user stories, epics
- Is regularly refined (“grooming”)
  - splitting, estimating, re-estimating items
  - developers estimate the size of the items
    - e.g. as story points
- Commitment: Product Goal
  - a future state of the product

User Stories and Epics

CS-C2130: Describe SW features as user stories

- User story
  - Basic format: “As a <role> I <want> [so that <benefit>].” [1]
  - Can be in other formats, as long as the above aspects are covered
  - Can be implemented in **one** Sprint
  - Works well for functional requirements, less well for quality attributes

- Epic
  - Basically a “big user story”, i.e. can’t be implemented in one Sprint
  - Usually broad in scope, short on details
  - Commonly needs to be split into multiple, smaller stories before the team can work on them
  - Compound stories, complex stories [2]

[1] Lucassen et al. (2016). The Use and Effectiveness of User Stories in Practice. In Requirements Engineering: Foundation for Software Quality. REFSQ 2016.
[2] Cohn, M.: User stories applied: for agile software development. Addison Wesley, Redwood City (2004)

Product Vision (CS-C2130)

1. Why?

- explain why the product is being built (the business view)

2. What?

- Product Goal, i.e. the desired state of the product in the end of the course project

3. For Whom?

- characterize the end users

Created based on the project proposal and further discussions with the PO

Sprint Planning: Topics 1 & 2

- Participants: PO, Developers, Scrum Master
- Understand
  - 1. **WHY** this Sprint is valuable
    - Sprint Goal
  - 2. **WHAT** can be done
    - items from the Product Backlog
- Discussion
  - PO explains
  - Developers ask questions
  - Joint decision on what can realistically be included

“WHY & WHAT?”

Sprint Planning: Topic 3

- Participants: Developers, Scrum Master (PO reachable for questions)
- Focus on
  - 3. **HOW** to implement the selected items
- May contain:
  - Overall design
  - Splitting Product Backlog items into tasks
    - building the Sprint Backlog
  - Estimating items/tasks
  - Renegotiating scope

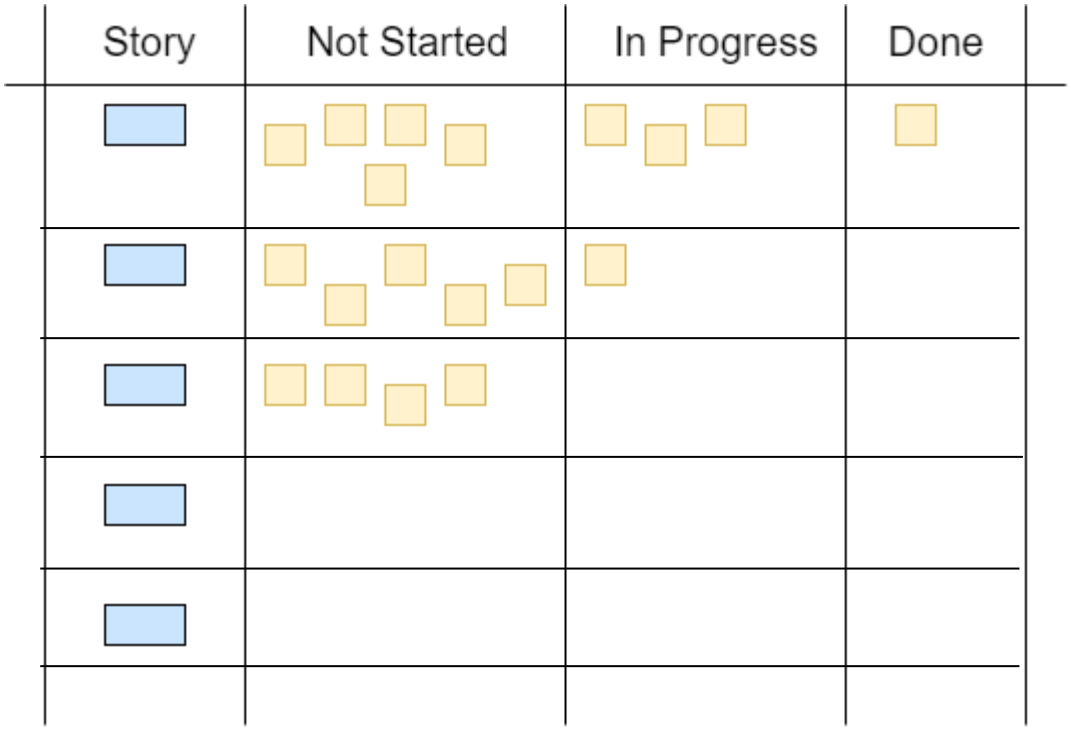
“HOW?”

Sprint Backlog

- Sprint Goal
- Sprint Backlog items
  - Some items from the Product Backlog, and the necessary tasks
  - Attributes of the tasks
    - name/description
    - effort estimate as hours or story points
      - at least if the tasks are large (>1 workday)

- Product and Sprint Backlogs should be in some backlog management tool
  - Jira, Trello, …
  - a well-organized Miro board is allowed too

Example: Scrum Board



Ongoing Sprint

Daily Scrum Meeting

CS-C2130
- At least once per week

- Participants: Developers, Scrum Master, (PO optional)
- Many possible structures and techniques possible, but
  - focuses on progress toward the Sprint Goal
  - produces an actionable plan for the next day(s) of work
- Improve communications, identify impediments, promote quick decision-making
  - less need for other meetings
- Max 15 min
  - If discussion is needed: follow-up meetings agreed and held afterwards

Definition of Done (DoD)

- Everyone must understand what “done” means for a Product Backlog item
  - possibly also for Sprints and releases
- When a Product Backlog item meets the DoD, an Increment is born
- Scrum Team must define their own DoD (and follow it!)
  - … and improve it when needed
- Typically things like
  - code is implemented, commented, integrated
  - automated unit tests have been written, and pass when executed

CS-C2130 minimum DoD:
- unit testing
- functional system testing
- coding standard

More Quality Assurance (CS-C2130)

- Quality attributes (non-functional requirements)
  - e.g. usability, security, performance, compatibility
  - identify the most relevant ones (at least one, but not too many)
    - consider them appropriately in DoD
- Peer testing
  - by some other team on the course
  - outsiders can often spot different bugs and improvement ideas than the developers themselves
  - you must plan how to best utilize the other team (when & what)
  - at least 8 man-hours per team using Session-based exploratory testing

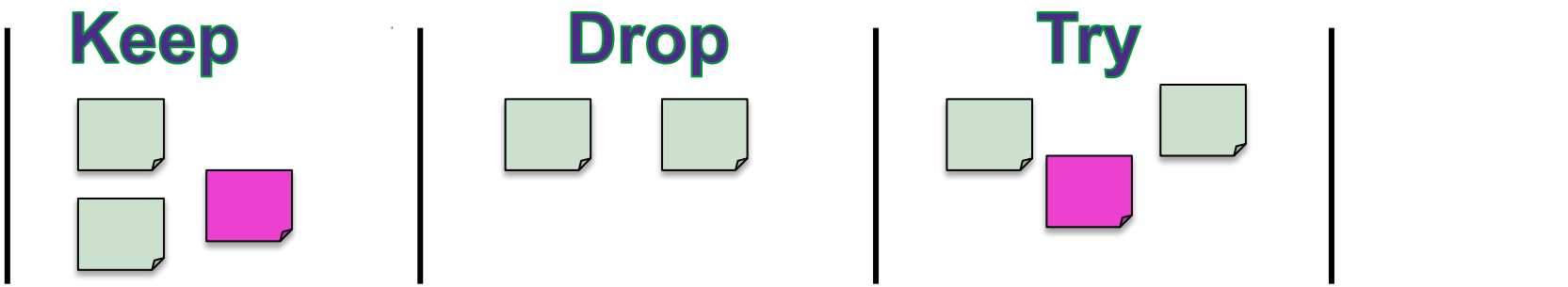
Peer testing:
https://mycourses.aalto.fi/mod/page/view.php?id=1200402

Sprint Review

- Participants: Developers, PO, Scrum Master, other stakeholders invited by the PO
- Inspection of the Increment and adaption of the Product Backlog, if needed
  - Hands-on inspection of the real software running live
  - In-depth conversation
  - What was accomplished in the Sprint?
  - What has changed in the environment?
  - Determine future adaptations

Sprint Retrospective

- Inspecting how the last Sprint went with regards to individuals, interactions, processes, tools, and DoD
- Participants: Developers, Scrum Master, PO (optional)
  - Developers discuss **what's working** and **what's not working** and agree on **changes to try**
  - Usually, the Scrum Master facilitates
  - Different techniques, try different ones!



Being Efficient: Doing a Sprint Change

- In one sitting
  - Sprint Review
  - Sprint Retrospective
  - Sprint Planning
- Requires access to the Product Owner

Sprint 0 (CS-C2130)

- Sprint Goal
  - "Set up the project so that everything is ready for starting sw development work from the first day of the following Sprint."

- Main tasks
  - product vision and initial Product Backlog
  - prototyping, selecting and **studying technologies**
  - deciding work methods and tools, e.g.
    - communication channels, team work sessions
    - practicalities of the Scrum events
    - backlog management, time tracking, version control

How to study efficiently as a team?

- Results presented to the PO and **to the Coach**

Last Sprint (CS-C2130)

- Focuses on finalizing the product for the final delivery to the PO

- Some tasks
  - bug fixing and finalization (no more new features)
  - acceptance testing by the Client
  - handover to the Client (both the system and any necessary knowledge)
  - preparing an excellent software demo and a project poster

Scrum Theory - Three Empirical Scrum Pillars

- Transparency
  - process and work results must be visible
    - Scrum artifacts provide transparency
- Inspection
  - Scrum artifacts and the progress toward agreed goals must be inspected frequently
    - done in the Scrum events
- Adaptation
  - Based on inspection, process or product my need adjustment

Want to Know More about Scrum?

- Scrum Guide
- Scrum Primer

Read the CS-C2130 Project Manual

1. It summarizes briefly the requirements set in the Scrum Guide.
2. It describes the modified/additional requirements set by the course.

In order to understand why and how to follow the Project Manual, you must read Scrum Guide / Scrum Primer

Agenda

- Next steps on the course

- Scrum basics and applying Scrum on this course
- **Additional requirements for the course projects**

- Accenture Quality Award, Jarno Hilvenius/Accenture

Additional Requirements for the Course Projects

- Process Overview document
- Technical Overview document
- Time Tracking
- Project Reviews

These are not a part of Scrum.

Document – Process Overview

- Document briefly the *currently* used work practices and tools **so that all stakeholders can understand how the team works**

- Minimum content
  - Project schedule and effort distribution
    - start and end dates of all Sprints
    - allocated effort per person per Sprint
    - dates of other events such as project reviews and team work sessions
  - Description of Scrum events
    - Sprint Planning, “Daily” Scrums, Sprint Review, Sprint Retros
    - e.g. preparations, participants, content of the event, expected results
  - Description of other main practices and tools
    - related to project work (backlog management, time tracking, communication, teamwork sessions etc.)
    - related to development work (version control, testing etc.)

See the template

Producing a document is not the main purpose. The most important thing is to adopt good work practices that can be realistically used.

Document – Technical Overview

- Very project specific
  - Decide with the PO what needs to be documented
- General goals
  - **Helping the Scrum Team during the project**
    - e.g., in communicating about the design or in dividing responsibilities
  - **Meeting the Client's needs after the project**
    - e.g., helping some new developers fix bugs or develop new features
- Minimum content
  - Document briefly the **most important architectural design decisions**
  - Document **one or more relevant views** of your architecture design
    - Google e.g. 4+1 architectural view model.

Time Tracking

- Total effort spent per student per each Sprint
  - includes everything related to the project
    - also studying new technologies, communication, …
  - must be visible also to the Coach
  - must be updated at least weekly
    - impossible to remember what you did two weeks ago
    - It is best to report own hours immediately
  - if someone falls behind or works extra in one Sprint, update the remaining hours in the coming Sprints accordingly

	M1	M2	M3	M4	M5	M6	M7	SUM
Sprint 0	40	35	35	20	20	35	35	Xxx
Sprint 1	20	40	40	55	55	40	40	Xxx
…	..	..	..	..	..	..	..	..
Sprint N	20	40	40	25	25	40	40	xx
Total	100	225	225	225	225	225	225	xxxx

(realized hours and remaining hours)

- Some backlog management tools support time tracking
- A simple spreadsheet can work too
  - if you are not interested in task level tracking

An example can be found in the Project Manual

Project Review

- November, February and May
- Participants
  - student team, PO, coach, teacher, and possibly some other people

- Team presents data on the project status
  - summary of the previous Sprints
    - Sprint Goals, Product Backlog items, and other achievements
    - main findings from the Sprint Retros
  - evaluation of software quality
  - spent effort per person per Sprint
- Team presents the results (mainly a software demo)
  - plan and rehearse

- After each project review, PO and coach evaluate the project

Project Reviews:
https://mycourses.aalto.fi/mod/page/view.php?id=1200397

Progress / Final report template (slides):
https://mycourses.aalto.fi/mod/resource/view.php?id=1200406

Project Review – Submit the Materials to the Stakeholders

- Product vision (Template available)
- Product Backlog
- Sprint Backlog of the current Sprint
- Process overview (Template)
- Definition of Done
- Technical overview
- Progress report (PR1-2) / Final report (PR3) slides (Template)
- Test session charter(s) for peer testing (Template) (PR3)

- Learning diaries (submitted individually by each student to MyCourses)

Send a link to the materials to the teacher, coach and PO

- 24 hours before each project review
- the link will be published in MyCourses

Tips for Working as a Remote Team

- Try to work f-2-f as much as possible, especially in the beginning of the project ☺

- Plan joint working sessions – just like you would if you were collocated
- Use e.g. Zoom and keep it open for the session
- Always use video when you are communicating in a meeting
- Have (e.g. Slack) channel for the team, always on

Proposing changes to the Scrum and Course Requirements

- Try to follow the rules (for real) first
- If you really need to change them
  - Make a motivated proposal to your coach
  - Try the changed version

Agenda

- Next steps on the course

- Scrum basics and applying Scrum on this course
- Additional requirements for the course projects

- **Accenture Quality Award, Jarno Hilvenius/Accenture**