



# CS-C2130 - Software Project 1, Lecture, 6.9.2023-1.12.2023

This course space end date is set to 01.12.2023    **Search Courses: CS-C2130**

/ departm... / Sections / project... / project...

Course feedback

Syllabus

## Project Manual

Scrum is a widely used process framework for managing complex product development. Learning to apply Scrum is one of the main educational goals of this course. Therefore, all teams must try using Scrum as defined in [the Scrum Guide](#), but also considering the exceptions mentioned in this Project Manual. If some Scrum requirements fit poorly into the project, the team may propose changes to them as described at the end of this document.

This Project Manual 1) summarizes briefly the requirements set in the Scrum Guide (Schwaber and Sutherland, 2020), and 2) describes the modified/additional requirements set by the course. Read [The Scrum Guide](#) (or the more concrete [Scrum Primer](#)) in order to understand better why and how to follow the instructions presented below. Note the [Changes between 2017 and 2020 Scrum Guides](#).

See also the following pages, which contain some additional requirements for the projects: [EESs](#), [Project Reviews](#), [Peer testing](#), [Gala](#).

### Contents

1. Scrum Team

2. Project Effort and Time Tracking

3. Product Vision

4. Product Backlog

5. Sprints

6. Sprint Planning

7. Sprint Backlog

8. Sprint Review
9. Sprint Retrospective

10. Sprint Change

11. Daily Scrum

12. Definition of Done

13. Process Overview

14. Technical Overview

15. Proposing Changes to the Scrum Requirements

### 1. Scrum Team

See [Scrum Primer](#) pp. 4-5.

The Scrum Team consists of the Product Owner (PO), Scrum Master and Developers.

*The Product Owner* is responsible for managing the Product Backlog so that the value of the product and the work of the Scrum Team is maximized. The PO may delegate some work to others, but remains accountable. The PO is a single person from the Client's organization.

*The Developers* take responsibility of delivering a potentially releasable increment of "Done" product at the end of each Sprint. They are SW Project 1 & 2 course students.

*The Scrum Master* ensures that both the developers and PO understand and adhere to Scrum. The Scrum Masters are students from the SW Project 3 course. In this project, the Scrum Master should at least:

- study, plan, teach and ensure the application of Scrum in the Scrum Team
- ensure the adherence to other course requirements described in the Project Manual
- prepare and lead the Scrum events
- manage team building (recruitment, team spirit, communication practices)
- oversee that the Developers learn to proceed the project on their own initiative
- initiate discussions on any problems, if the team does not react to them
- give tips on methods and tools that have been taught on the SSE major's core courses related to e.g. testing, requirements engineering, and software design
- work also as a developer, if the Scrum Master's responsibilities don't take the whole time budget (especially when taking >5cr course version)

### 2. Project Effort and Time Tracking

**Each student must allocate 25 \* (credits - 1) hours for the project (e.g. 10cr ~ 225h, 5cr ~ 100h).** The effort worth 1cr is spent before the project for the lectures, studying Scrum, forming the team, and selecting the topic.

The course requires that **weekly updated total effort spent per student per each Sprint is visible to the Scrum Team and the Coach**. A recommended way for time tracking is to use a backlog management tool that supports it, but a simple alternative solution is to use a spreadsheet (see [an example of a Time Tracking spreadsheet](#)).

### 3. Product Vision

The product vision is the overarching goal every project stakeholder shares. The input comes mainly from the PO, but it is important that the Developers formulate the Product Vision in order to show that they have understood what the project is about. The course requires that **the Scrum Team creates a short product vision**, which

- describes why the product is being build (the business view)
- describes the Product Goal, i.e. the desired state of the product in the end of this project (see Scrum Guide)
- characterizes the end users

### 4. Product Backlog

See [Scrum Primer](#) pp. 5-8.

**The Scrum guide requires having a Product Backlog which is an emergent, ordered list of what is needed to improve the product.** Typically, not all the items in the Product Backlog will be implemented during the course project. The Product Backlog includes primarily new customer features, but also other types of items (improvement goals, research work, even known defects).

The Product Backlog is never complete, and its earliest development only lays out the initially known and best-understood requirements. Higher ordered items are usually clearer and more detailed than lower ordered ones. The items that are likely to be selected to the next Sprint are refined so that any one item can reasonably be "Done" within one Sprint. Refinement adds details such as a description, order, and effort estimate (e.g. in story points). The course requires that **the description of those Product Backlog items that represent new software features follows some commonly used user story template** (see e.g. [Wikipedia-User story](#)).

### 5. Sprints

The Scrum Guide requires splitting the project into Sprints. Each Sprint starts with the Sprint Planning event and ends with the Sprint Review and Sprint Retrospective events.

The course requires that **the project contains at least 6 Sprints** (includes "Sprint 0"). **In the beginning of the project, the Scrum Team must plan 1) the start and end dates of the Sprints, and 2) allocation of the budgeted effort per Sprint per student.** The course recommends planning equal effort instead of equal duration for all the Sprints, considering e.g. exam weeks.

#### First Sprint ("Sprint 0")

In this course, the typical Sprint Goal for Sprint 0 is to set up the project so that everything is ready for starting software development work from the first day of the following Sprint. **Sprint 0 results must be presented also to the Coach** in addition to the normal audience. **At least the following tasks must be completed in Sprint 0:**

- [team building activities](#)
- [crafting the Product vision](#)
- [creating the initial Product Backlog](#) (initially known and best understood requirements)
- [prototyping, selecting and studying technologies](#) (decisions documented in the Technical overview)
- [selecting initial work methods and tools](#) (documented in the Process overview)
- [creating the initial Definition of Done](#)

#### Last Sprint

In this course, the last Sprint focuses on finalizing the product for the final delivery to the PO, and on presenting the results and experiences of the project to the course personnel and other stakeholders. **The last sprint must end before the last project review and contain at least the following tasks:**

- [acceptance testing by the Client](#)
- [bug fixing and finalizing](#) (usually no more new features)
- [handover to the Client](#) (both the system and any necessary knowledge)
- [preparing an excellent software demo and a project poster](#)
- [creating the final report slide-set](#)

### 6. Sprint Planning

See [Scrum Primer](#) pp. 8-11.

The Scrum Guide requires **having a Sprint Planning event which** contains three parts:

1. Why? Based on the proposal from the PO the whole Scrum Team defines a Sprint Goal that communicates why the Sprint is valuable to stakeholders.
2. What? Through discussion with the PO, the Developers select items from the Product Backlog to the Sprint. The Developers' knowledge about their past performance, their upcoming capacity, and their Definition of Done, help them decide how many items can be selected to be implemented.
3. How? The Developers plan the work necessary to implement the selected Product Backlog items. This is often done by decomposing Product Backlog items into small work items of one day or less. Small work items help 1) tracking the progress of the sprint, and 2) understanding what concretely needs to be done.

### 7. Sprint Backlog

See [Scrum Primer](#) pp. 9-10.

The Scrum Guide requires **having a Sprint Backlog which contains the Sprint Goal, the Product Backlog items selected for the Sprint, and the actionable plan for delivering them.** The Developers modify the Sprint Backlog throughout the Sprint as they learn more about the work needed.

**The course requires that the actionable plan must include an up-to-date list of the necessary work items with a name or description. Furthermore, if the work items are much larger than a day, they must have an effort estimate as hours or points.**

The course requires that **the Product Backlog and Sprint Backlogs are online in an adequate tool**. Some examples are JIRA, Trello (with suitable Scrum extensions) or a well organized Miro board.

### 8. Sprint Review

See [Scrum Primer](#) pp. 13-14.

The Scrum Guide requires **having a Sprint Review event at the end of each Sprint**. The Scrum Team and possibly other stakeholders review the results of the Sprint and possible changes in the environment. They collaborate on what to do next and may adjust the Product Backlog.

### 9. Sprint Retrospective

See [Scrum Primer](#) pp. 14.

The Scrum Guide requires **having a Sprint Retrospective event at the end of each Sprint**. In the retro the Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and their Definition of Done. The Scrum Team identifies the most helpful changes to improve its effectiveness, and may even add them to the next Sprint Backlog.

### 10. Sprint Change

In this course, a student spends only 30-40h per Sprint. Therefore, the Scrum events should be short leaving enough time for the development work. Furthermore, the Sprint Review, Sprint Retro and Sprint Planning events can be combined into a single Sprint Change event. The Sprint Planning event can be further shortened, if some effort is allocated during the Sprints for refining the Product Backlog (See [Scrum Primer](#) pp. 13).

### 11. Daily Scrum

See [Scrum Primer](#) pp. 11.

The Daily Scrum is a max. 15-minute event for the Developers to inspect progress toward the Sprint goal and adjust the upcoming work in the Sprint Backlog as necessary.

Daily Scrums improve communications, identify impediments, promote quick decision-making, and consequently eliminate the need for other meetings. If discussion is required, have one or more follow up meetings after the Daily Scrum with the needed people.

The course requires **having the "Daily" Scrum at least once per week**. If the team cannot work together regularly, having an online meeting (even asynchronously) is better than having no meeting at all.

### 12. Definition of Done

See [Scrum Primer](#) pp. 8.

The Scrum Guide requires **having a Definition of Done (DoD)** so that the Scrum Team can have a shared understanding of what it means for a Product Backlog item to be complete. As the Scrum Team matures their DoD will expand to include more stringent criteria for higher quality. Sometimes it is useful to define [DoDs for different levels such as a feature, sprint, and release](#).

The course requires that the **DoD must consider adequately at least:**

- [unit testing](#)
- [functional system testing](#)
- [usage of a specified coding standard](#)

Furthermore, the **course requires that in addition to aiming at functional correctness (=no bugs), the Scrum Team must identify the most relevant quality attributes** (at least one, but not too many) , **and consider them appropriately in DoD, Product Vision and/or Technical Overview**. Typical [quality attributes](#) include, e.g., usability, security, performance, and compatibility (e.g. targeted operating systems or browsers and their versions, or phone models). "Good usability" or "adequate performance" are vague goals, but quantifying them with some concrete metrics helps to understand and evaluate them better.

See also the information about the [Peer testing](#) practice used on this course.

### 13. Process Overview

The course requires **documenting briefly the currently used work practices and tools** so that the Scrum Master, developers, PO, Coach and other course personnel can understand how the team works. A picture is worth a thousand words. The main goal is not to produce a document but to adopt good work practices that can be realistically used. Sprint Retros will certainly lead to changes in the work practices.

**The Process overview must contain at least the following content** (or links):

1. Project schedule and effort

- [start and end dates of all Sprints](#)
  - [allocated effort per person per Sprint](#)
  - [other events such as project reviews](#)
2. Description of the recurring events of the Sprints (how and when)

- [Sprint Planning](#), "Daily" Scrums, [team work sessions](#), [Sprint Review](#), [Sprint Retrospective](#)
3. Description of other main practices and tools

- [related to project work \(backlogs, time tracking, communication etc.\)](#)
  - [related to development work \(version control, testing etc.\)](#)

### 14. Technical Overview

The useful extent and content of technical documentation depends largely on the project. Generally the documentation should consider two things:

1. Helping the Scrum Team during the project, e.g., in communicating about the design or in dividing responsibilities.
2. Meeting the client organization's needs after the project. Therefore, focus on what is complicated or not obvious from the code or code comments.

The course sets only minimal requirements for the technical documentation:

1. **Document briefly the most important architectural design decisions.** Such design decisions may address, e.g., how the system is divided into its parts, what responsibilities these parts have, what are the interfaces, and what are the used frameworks and technologies.
2. **Document one or more relevant views of your architecture design.** See e.g. [4+1 architectural view model](#).

Otherwise the Scrum Team needs to decide themselves what needs to be documented. The course provides no template for the document.

### 15. Proposing Changes to the Scrum Requirements

Sometimes strict usage of Scrum may not be either educational or productive for a project. For example, the PO may be too busy to take care of all her responsibilities, or the developers fail to arrange regular team work sessions, and must sometimes work as a distributed team.

However, if in any way possible, the Scrum Team must first try working according to all the requirements mentioned in this Project Manual. Thereafter, they may propose changes to individual requirements, if they fit poorly to their project. A proposal must be sent to the Coach. It must:

- show that the Scrum Team has truly tried to follow the corresponding requirement(s) and understands their purpose
- describe the problem and the proposed solution

### Additional Materials

- [The Scrum Guide](#) (available also in [Finnish](#))
- [The Scrum Primer](#)
- [Scrum Glossary](#)
- Slides of the Scrum lectures
- Course Schedule

Last modified: Thursday, 8 February 2024, 1:59 PM

#### Previous activity

← Gala

#### Next activity

Peer Testing →



#### Tuki / Support

##### Opiskelijoille / Students

- MyCourses instructions for students
- email: [mycourses\(at\)aalto.fi](mailto:mycourses(at)aalto.fi)

##### Opettajille / Teachers

- MyCourses help
- MyTeaching Support form

#### Palvelusta

- MyCourses rekisteriseloste
- Tietosuojailmoitus
- Palvelukuvaus
- Saavutettavuusseloste

#### About service

- MyCourses protection of privacy
- Privacy notice
- Service description
- Accessibility summary

#### Service

- MyCourses registerbeskrivning
- Dataskyddsmeddelande
- Beskrivning av tjänsten
- Sammanfattning av tillgängligheten

