

## SPRINT 2

### 1 Introduction

The following sections contain a summary of what we expect you to do in the end of Sprint 1 and throughout Sprint 2. We summarise requirements using the user story format, so that you get a high-level overview of what is needed, a description of who is the main stakeholder, and a rationale of why we want to see it. This might help you understanding how to conduct the different steps.

There are a number of technical constraints. These are listed in Section 5.

### 2 Finishing up Sprint 1

1. **US1:** As a Product Owner, I want to be able to follow the group's reflections about Sprint 1 and what they learned from that sprint's way of working, so that I can judge the group dynamics and their understanding of the Scrum process.

Make sure you have discussed the sprint before you meet the TA, otherwise that short time slot will not be well-spent. Focus on the following three questions in your discussions:

1. What worked well?
2. What did not work well (and why)?
3. What would you like to change in coming sprints?

To support this discussion, try to focus on the different parts of the process: activities you did, communication between the team members, decision making, and outcomes in terms of artefacts. In your discussion with the TA, you will focus on this retrospective!

### 3 Planning Sprint 2

1. **US2:** As a Product Owner, I want to know who is the group's ScrumMaster, so that I have a clear point of contact.
2. **US3:** As a Product Owner, I want to be able to get a high-level overview of a group's sprint plan, so that I can adjust in case my priorities are different, or the planning seems unrealistic.
3. **US4:** As a Product Owner, I want to be able to clearly know what is needed for a group to complete a sprint (definition of done), so that I can assess the outcome after the sprint.

4. **US5:** As a group, we want to have a clear task list, so that we can see progress and identify who is responsible for what.

To start off, randomly choose one group member as a ScrumMaster for this sprint. From now on, you will do this in each sprint, every time choosing someone who has not yet been ScrumMaster.

To plan Sprint 2, try to decide which user stories you need/want to implement. This requires that you discuss what is most important to do (e.g., do you have user stories that are needed for all other stories?), how much work you can do in a sprint, and also think about customer value (e.g., what would be most valuable to have?). Move the selected user story issues in GitLab to your sprint backlog (an additional issue list/label on GitLab) and get your PO's feedback electronically (via mail/piazza). Finally, you need to agree with the PO what it means that you have completed the sprint (the "definition of done", DoD). What will you demonstrate? A list of running test cases? Some demonstration using the console? Anything else? Your PO might require you to change things (based on his/her priorities and his/her domain knowledge). Once you agreed on the scope/the sprint backlog, proceed breaking down all user stories for Sprint 2 into tasks that can be handled by individuals. Finally, let all your group members pick tasks (do not assign).

## 4 Running Sprint 2

1. **US6:** As a Product Owner, I want to be able to see that all individuals in a group have tried Test-Driven Development.
2. **US7:** As a Product Owner, I want to be able to see that the group's code has at least 50% statement coverage, so that I can be confident in the product quality.
3. **US8:** As a Product Owner, I want to be able to see sufficient comments in the group's code, so that I have an easier time understanding it.
4. **US9:** As a Product Owner, I want to be able to get a quick overview of the standup meetings and important decisions, so that I can estimate the progress of the group.

During Sprint 2, you will start implementing the system that you have proposed, based on your sprint planning. This means that you start implementing, in Python, the functionality in your sprint backlog. We require not only that the tasks in your sprint backlog are actually implemented, but also documented and tested in a proper way. For documentation, make sure that you comment all of your code in a way that makes it understandable. Try to document high-level constructs (e.g., what does a function do?) and non-obvious code parts. For testing, your code needs to have at least 50% statement coverage through unit tests. Additionally, at least one task per team member needs to be implemented in a test-driven way. Demonstrate this by pushing first your tests to the repository, then developing (and ultimately pushing) the actual functionality<sup>1</sup>.

Throughout the sprint, keep doing standup meetings and discuss in your group. This will be assessed in the next sprint retrospective. Keep your decision protocol that you have started in Sprint 1, and additionally summarise the standup meetings.

---

<sup>1</sup>We are aware of the fact that this is something we can 100% check. We trust you that you actually do this properly.

## 5 Technical Constraints

There are a number of practical technical constraints we impose:

- You have to use **python 3.\***
- You have to use the **unittest** library for writing unit tests
- You have to use the **coverage** library for calculating test coverage and generating HTML coverage reports
- For Sprints 2 and 3, you are not supposed to implement any (graphical) frontend. It is sufficient (and wanted) that you "demonstrate" your functionality through sending WebSocket messages.
- Any function that should be exposed to the outside (e.g., that will in the future be used by a graphical frontend or another component) has to be accessible through WebSocket messages. You can follow the example that will be uploaded on Canvas after L7 to see one way to do this.

You have to follow all of these constraints.

## Submission and Assessment

**Sprint 2 deadline: Wednesday, 18th September, 23:59**

Make sure all relevant files are committed on time. Any late commits will be ignored.

For Sprint 2, make sure the following parts exist:

1. Code in src/app subfolder
2. Test code in src/test subfolder
3. Decision protocol and summary of standup meetings in the docs/sprint2 subfolder
4. Generated HTML coverage report in docs/sprint2 subfolder
5. Maintained/updated product and sprint backlogs on GitLab

The individual part is submitted through Canvas (assignments are named by sprint) - as in Sprint 1.