

# Assignment 6 – First Sprint

---

## Due

Your **demo is due at the beginning of your lab in the week of November 3<sup>rd</sup>, 2014**. The demo will be part of the sprint review meeting, in which you have to demonstrate the working product to the product owner (TA).

**At each lab during the sprint**, you are going to have a Scrum Meeting with your ScrumMaster (TA) to go over the progress you made and the tasks you are going to work on for the next week. Continuous progress will be part of your mark for this phase too, so make sure to make progress during the first week.

The **peer evaluations are due at midnight on the day of your lab in the week of November 3<sup>rd</sup>, 2014 via handin**.

## Objectives

The goal of this assignment is to have you go through one sprint cycle and finish the high-priority items for the product. You will learn more about how to apply Scrum to a project, e.g. daily (in our case weekly) Scrum meetings and completing tasks and user stories. You will also learn how to develop in a small team using a source control mechanism to share code and how to communicate with each other through the issues / user stories / meetings and sharing the source code.

## Procedure – First Sprint

As you will remember from the lecture, the sprint includes coding as well as design and testing and other activities. You will first need to estimate how long each of your tasks (not just the whole user story) is going to take, if you haven't done so, and then **divide up the work based on the estimates**. Everyone in the team should end up with approximately the same workload. Obviously, the estimates might not be perfect, but you can redistribute work (tasks and user stories) with your teammates later on. Make sure that by the end of the first sprint you have implemented at least the user stories with high priority and if there is more time, also some of the other user stories. However, it might also happen that you will have to defer certain user stories to the next sprint (something that can happen in Scrum) if you are running out of time and you might have underestimated the time it takes to complete the user stories (if you have to defer stories, we will assess at the end of the sprint whether you made satisfactory progress or not). You can either split up the work by user stories and assign full user stories to one person, or, especially for bigger user stories and also to balance the workload, you can split user stories up by task and assign individual tasks by person, i.e. have multiple developers working on one story.

For the last assignment, you created **a first design** that will serve as a reference for communication, but this should not be set in stone. Adapt it as you go and however you need. The better you understand the problem and how to solve it, the easier it will be for you to come up with a good design. Also, you don't have to adapt the actual UML diagram during this sprint; the main purpose of the diagram is for your team to have a common understanding. However, for the actual sprint review meeting a UML diagram might help you for explain some parts or answer questions from your TA. You will be required to update your UML diagram during the second sprint, so you may prefer to keep it up-to-date as you go. The design and the architecture of your project should also help you to divide up the code into cohesive packages, i.e. put all modules that belong to a particular part of the project into one package and other modules in other packages. This will also make it easier to coordinate with other people when you all develop code at the same time and want to avoid conflicts.

As stated beforehand, **testing is also part of each sprint!** If possible, create some good automated test cases that you can run continuously and check whether your code still works properly. Some parts of the project, in particular the UI, are difficult to test by writing a test case, in that case you can write a test plan, i.e. write down the steps that a “tester” should follow to check whether certain parts work correctly or not (these plans might be somewhat similar to the acceptance criteria you wrote, since they are more from the perspective of using the product rather than the specifics of the code, but they will help you to find errors in your code). Writing the test cases ahead of time will allow you to understand better what code you have to write to complete a task / user story. We will check the tests that you wrote (including possible test plans). **You must at least have test plans. During this sprint, automated tests are optional.**

During the sprint, you might discover defects or bugs in your code or new tasks that you need to complete. In that case create a new issue (task or defect) and assign it to the Product Backlog or assign it to the current sprint if it should be done/fixed. Then you can either work on it yourself right away, ask someone else to fix it or bring it up at the next Scrum meeting if it blocks anyone. Usually, there should not be any new user stories created during the sprint and assigned to the current sprint, but if you think of some you can create new user stories and put them into the Product Backlog so that they can be discussed in the next sprint planning meeting.

*Groups using GWT:* Your demo must be from a single deployment to the App Engine (or other server, if discussed with your TA). There will be penalties if you demonstrate different parts of the application without integrating them.

*All groups:* Your demo must be performed in “production mode” (i.e. deployed correctly on a server) and not in your “development mode”.

## Procedure – Continuous Progress & “Daily” Scrum

Throughout this sprint, and the following one, you are expected to make continuous progress, not just the team as a whole, but each member of the team. Each lab from now on will have a short (approximately 5 minute) Scrum meeting at the beginning in which you will discuss the progress with your ScrumMaster (TA). Remember the three questions for this meeting are: (1) What did you do last week? (2) What are you going to do this week? (3) Is anything impeding your progress? The TAs will check that each one of you made continuous progress in terms of completing your tasks, coding and writing test cases. You have to **prepare for this meeting** so that you can quickly answer these three questions, i.e. know what the tasks are that you want to complete in the next week and what you have worked on.

When you are coding and completing a task, make sure to regularly commit your changes (at least twice a week!). If you have automated test cases, you should run your complete suite of test cases every time you make a significant change.

## Procedure – Sprint Review Meeting & Sprint Retrospective Meeting

At the end of the first sprint, there will be a sprint review meeting with your Product Owner (TA) in which you have to demonstrate your working product. The product owner will then declare which items (user stories) are considered “done”. Incomplete items will be added back to the product backlog, and in our case to the next sprint. However, this should not happen very often. You will have 5 minutes to demonstrate your product (5 minutes is the absolute maximum and will be strictly enforced!). Go through the user stories and show how your product provides the functionality described in your user stories. Every team member should participate in the demo and I highly recommend that you practice your demo before you meet with your product owner. You may want to bring a demo script so that you are confident that you won’t forget to demo anything. The TA will also ask you questions about your implementation and design (these questions will be directed towards individuals, so you should have a general idea about the project, its implementation and design).

Once you finish demonstrating the product and you finish answering the TAs questions, you will have 4 minutes to reflect with your TA on the progress you made and discuss questions such as what went well, what could be improved, what did we learn, what are we still unsure of and what actions will we take (sprint retrospective meeting).

After the sprint retrospective meeting, you will have a short sprint planning meeting (4 minutes) for the next sprint in which your product owner will tell you about the user stories that should be completed in the second sprint and the new features that he/she would like to be included in the sprint. And then the next sprint starts (story pointing, breaking up tasks, designing and coding, but now all as one big sprint rather than split up into different assignments). Be sure to check the website for Assignment 7 so that you know exactly what to do in the second sprint.

## Procedure – Team Assessment

Each person needs to submit a peer evaluation at the end of the sprint. This evaluation is individual and not shared with the group. Please submit this evaluation as soon as you finish your sprint review and sprint retrospective meeting (note the deadline above). The peer evaluation will influence your overall grade for Sprint 1.

## Deliverables

### In the sprint review meeting:

- Demonstration of your working product with all user stories for the sprint completed (or most and some being deferred to next sprint; we will assess whether or not you made enough progress). This demo must be done from a single application that is deployed to the App Engine (or other server, if discussed with your TA).

### Each week (briefly discussed in Scrum meetings):

- Regular individual contributions by completing tasks and delivering change sets to the stream

### After you finish your assignment and before 11:59pm on the day of your lab in the week of November 3<sup>rd</sup>, 2014:

- Each of you has to fill out the peer evaluation form and submit it online (don't forget to include your team name in the evaluation). You can find the form at <http://www.ugrad.cs.ubc.ca/~cs310/peerEval.pdf>

### How to use handin for your peer evaluation:

1. Put the file in your `~/cs310/a6-X-peereval` directory (where you must replace X with your team name according to the list of groups I posted on Piazza)
2. Run the command `handin cs310 a6-X-peereval` (again, replace X with your team name)

### Other notes about handin

1. You can confirm that your assignment was handed in correctly by using the `-c` flag (ie, `handin -c cs310 a6-X-peereval`)
2. If you need to overwrite a previous version that you handed in, using the `-o` flag (ie, `handin -o cs310 a6-X-peereval`) Assignments will be accepted by handin up to the due date.
3. If you are using the web handin, you must zip your assignment first because it will only accept zip files.

## Grading Scheme:

1. (10%) Peer Evaluation (completing the evaluation and giving helpful comments)

**NOTE:** If you do not complete the peer evaluation, you will receive 0/10 on the peer evaluation grade and you will incur an additional 5 mark penalty so your maximum grade will be 85/100. Your teammates will not be penalized.

2. (10%) Individual continuous progress: regular individual contributions
  - a. Reasonable code contributions to the project
  - b. Reasonable fraction of the entire team workload
  - c. Participation in the Scrum meeting (including being prepared for each meeting)

3. (80%) Demo

**NOTE:** this part will be adapted according to the peer evaluations and individual continual progress, i.e. the peer evaluation and individual continuous progress grades will be used as a percentage-multiplier in calculating your individual demo grade.

- a. Completed and demonstrated a reasonable amount of the user stories
- b. Demonstrated user stories fulfill acceptance criteria
- c. Demonstrated user stories correspond to initial sprint backlog
- d. Demo is functional (few defects, no show-stopping defects...)
- e. Demo is well-planned
- f. [Groups using GWT] Demo is given from a single application that is deployed to the App Engine (or other server, if discussed with your TA)
- g. [All Groups] Demo is performed in production mode, not development mode
- h. Test plans or automated tests exist

## BONUS

There will be a bonus for the best project team at the end of the second sprint. We take many factors into consideration when deciding which team(s) should get the bonus marks. We are looking for teams that work well together, communicate clearly, work continuously, and create a good application.