

Assignment 4 – Sprint Backlog

Due

24 hours prior to your lab in the week of October 13th, 2014 for everything except the peer evaluation. Late assignments will not be accepted and the whole team will receive 0 for the assignment. No exceptions will be made.

Peer evaluation is due on the day of your lab in the week of October 13th, at midnight (after your lab).

Objectives

The purpose of this assignment is to introduce you to sprint planning, estimating the effort required for completing user stories, creating a sprint backlog, breaking down the tasks required to complete a user story and defining the goal of your first sprint.

Procedure – Sprint Backlog

1. UPDATE USER STORIES BASED ON FEEDBACK: At the beginning of your lab your TA (product owner) will go over the product backlog that you created for this week's lab and provide you with some feedback. Your first task is to update your user stories based on TA feedback. You should do this before you continue with the rest of the assignment. We will not grade Assignment 4 submissions if the user stories have not been updated to reflect your TA's feedback. Your user stories will form a contract between your team and your TA. If you want to update your user stories at any point during the project, you must have permission from your TA. Your final project will be graded based on the user stories that your TA has agreed to.

The TA in the role of the product owner will then tell you the priorities he/she has with respect to the user stories (1 high, 3 low). Stories with priority 1 should definitely be in the first sprint. Some stories with priority 2 could also be moved to the sprint backlog of the first sprint, but it depends on what your team thinks it can commit to during the coming sprint.

2. DETERMINE GOAL OF SPRINT 1: From the prioritized user stories your team then has to determine a goal of the sprint, i.e. a short description of what the sprint will attempt to achieve. See more at <http://www.mountangoatsoftware.com/scrum/sprint-planning-meeting>. Enter your sprint goal onto your team's wiki on GitHub.

3. MAKE ESTIMATES FOR STORIES: Now that you have established the sprint goal, your team will play a deckless version of "planning poker" (http://en.wikipedia.org/wiki/Planning_poker): For each user story that will potentially go into the first sprint, each of you will (on a scrap of paper) write down an estimate for the number of hours you believe that story will take. You then reveal your estimates simultaneously. Then, you discuss why everyone believes their estimate, and you arrive at a team estimate for that story. At that point, your team should already start taking notes about the tasks that people think are involved in each user story. Once you complete the planning poker game, your team has to determine which user stories you can commit to for the first sprint based on the estimated effort and the priorities given by the product owner. If you want to play a more official game of planning poker, you can print your own cards (e.g. from http://www.wis.win.tue.nl/2R690/doc/agile_planning_poker.pdf) and play during a team meeting.

4. MAKE A MILESTONE FOR YOUR SPRINT IN GitHub: Once you determine the user stories for the first sprint, you need to create a new Milestone for your Sprint. In GitHub, create a Milestone called Sprint 1, and assign all of your chosen user stories to this Milestone (ie, they will no longer be assigned to the Product Backlog Milestone as they will now be assigned to the Sprint 1 Milestone).

5. BREAK STORIES INTO TASKS: Now, your team has to break down the user stories from Sprint 1 into the tasks that are involved in completing each of the user stories.

6. STORE TASKS IN GitHub: Create a new Milestone in GitHub called Sprint 1 Tasks. For each user story from Sprint 1, break it down into the tasks that you must complete in order to accomplish that user story. For each task, create a new Issue describing the task and assign it to Sprint 1 Tasks. You must also list the Tasks in the parent user story from Sprint 1 using their Issue Number from GitHub. In many issue tracking systems, there is a better way to break user stories into tasks. However, we wanted to use GitHub for issue tracking rather than using separate systems for source control and issue tracking.

The discussions you had during planning poker might help you break down your user stories into tasks. However, since you are not necessarily familiar with all the technology required to accomplish the user stories, your team will most likely have to do some research (for example to determine how and where data would be persisted and how difficult it is going to be). The research will help you to better break up the user stories into tasks. This is important since the TAs will evaluate whether you broke up your user stories into appropriate and realistic tasks and whether you are covering all of what is required.

Here is an example of a user story broken down into tasks. This is not necessarily a “perfect” solution, but it’s a good one and captures most tasks that have to be done to complete the user story. However, you should consider what else there might be that has to be done to complete the user story.

1. As a user, I can create and load data into the system in order to create a visualization.

Priority: 3

Acceptance Criteria:

- The user can specify a data file.
- The system imports the data and displays a message, or an error if the file did not meet the format standards.

Story Points: 8

Task Breakdown:

- specify data schema
- specify import format
- configure RPC mechanism for loading data remotely
- specify web-based storage mechanism (e.g. JDO)
- write test cases

Procedure – Peer Evaluation

Each person needs to submit a peer evaluation. This evaluation is individual and not shared with the group. Please submit this evaluation as soon as you finish your group work for this assignment. It is due the day of your lab in the week of October 13th, 2014 at midnight (after your lab).

You can find information about the peer evaluation here - <http://www.ugrad.cs.ubc.ca/~cs310/peerEval.pdf>

Deliverables

24 hours prior to your lab in the week of October 13th, 2014

- Your user stories in your Product Backlog must now be updated based on your TA feedback and also contain all the priorities you got from your TA and story point effort estimation
- Your team must have the sprint backlog (Sprint 1 Milestone) created in GitHub, including all user stories assigned to this sprint. User stories have to have the required information such as the acceptance criteria, and the list of associated tasks.
- Your team must have the tasks for Sprint 1 in a Milestone called Sprint 1 Tasks on GitHub.

After you finish your assignment and before midnight of the day of your lab in the week of October 13th, 2014,

- Each of you must submit the peer evaluation form via handin (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/a4-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 a4-X-peereval` (again, replace X with your team name)

Other notes about handin

- You can confirm that your assignment was handed in correctly by using the `-c` flag (ie, `handin -c cs310 a4-X-peereval`)
- If you need to overwrite a previous version that you handed in, using the `-o` flag (ie, `handin -o cs310 a4-X-peereval`) Assignments will be accepted by handin up to the due date.
- 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 hand in your peer evaluation, you will receive 0/10 on the peer evaluation grade, so your maximum grade on the assignment will be 90/100. Your teammates will not be penalized.

2. (90%) the Sprint Backlog (including the breakdown into tasks)

NOTE: this part will be adapted according to the peer evaluations, i.e. the peer evaluation will be used as a percentage-multiplier in calculating your grade

- a. Completeness – the sprint backlog must be complete, including a complete list of tasks for each user story; the product backlog must be complete and contain the required information for each user story, including the story points.
- b. Appropriateness and Realism – the user stories must be broken down into realistic and appropriate tasks
- c. Clarity of the tasks – the tasks must be understandable and unambiguous
- d. The wiki page on GitHub must contain your sprint goal