Iteration 2

Please sign up for the Iteration 2 meeting with your course staff supervisor. To sign-up, please look here (accessible to only logged-in users) for the course staff supervisor in charge of your team, then enter your team name (e.g., T800) in one of the available slots for that person shown in the available time slot wiki page for that person:

Tiantian's Meeting Slots
Qing's Meeting Slots
Peilun's Meeting Slots
Sameksha's Meeting Slots

Timeline

Iteration 2 is the period Starting at the end of the Iteration 1 meeting and Ending at the beginning of Iteration 2 demo time between Nov 12 - Nov 16 (Monday-Friday).

Iteration 2 is worth 15 points.

We will use most guidelines that we had for Iteration 1. Based on our experiences with Iteration 1, we have also listed the things that your group shall pay more attention to.

General Guidelines and Suggestions:

- Find a tool that works best for your team to communicate. Many past CS427 teams used GroupMe or Slack as their main method of communication and recommended it.
- Whenever you run into issues, the earlier you inform us, the chances are higher that your issues will be resolved on time.

Before the meeting:

- Ensure that your wiki page is up-to-date, following the format that we suggested.
- Your team's wiki shall be frequently updated to show continuous progress, not moments before the meeting, and definitely not during the meeting. (It's fine for the scribe to keep meeting notes on Wiki but not for the rest of teamwork to be done on Wiki during the grading meeting.)
- Ensure that the user stories are listed and broken down into tasks for all iterations prior to the meeting.
- For this iteration, future iterations, and future demos, ensure that your project has some project code along with automated tests for the code components. Only project code alone is not enough. You will need automated tests as well.
- Please make sure that you commit the code often into the Git repository. Please ensure
  - to include meaningful commit logs/messages
  - that your pair's name or netid is included in the commit message when working in pairs
  - not to commit automatically generated files (such as class files or the .git directory) into the repository
  - that you contact the course staff supervisor as soon as possible if you're unsure how to organize your Git; there are other structures that could make sense based on your project, e.g., you may have two top-level subprojects with their own histories, say, .../T800/core/ and .../T800/plugin/
  - An easy way to show that you're following XP is to have several commits for each story/task: first commit some new tests that you added (even if failing, although in general it's not good to have failing tests), then commit the code that makes the test pass, then refactor the code and commit the changes. That way you can show you're following the test-code-refactor cycle.
- Tag the contents of your Git repository at the end of Iteration 2 as Iteration-2.
- Ensure that your project has collective code ownership. While the main authors of the code will be more familiar with a certain component, we do expect that everyone has – at least – a high level overview of all the different components in the system (even those that they did not work on exclusively). All members shall be prepared to answer questions about the system.
- Ensure that you identify risk items and their risk-management actions.
- Bring up any important technical issues/questions with the course staff supervisor before the meeting especially if your team is running into serious problems (e.g., you cannot checkout/compile/run your code, or you have no tests). Please do not wait to bring up your problems during the meeting, since you lose points that way.

During and after the meeting:

- Ensure that your team comes to the meeting prepared with an agenda.
- Bring one presentation laptop that will be used to connect to the projector. Make sure that everything that your team is going to talk about during the meeting is installed on the presentation laptop. Make sure that the presentation laptop can build and run the presented code and project. Do not switch between laptops during the meeting, because this wastes time.
• If your presentation laptop does not have a built in VGA adapter (this is true for Macs), please bring your own VGA adapter. Most Siebel Center conference rooms expect VGA for connecting to projectors.
• I2CS students will work with Boyang to find a screen sharing solution that works for all members of the team.
• Your team members shall assume the roles of MARS again (see Iteration 1 for more information). However, each member shall pick a different role from that in the Iteration 1 meeting. Please make sure each team member has active participation during the meeting.
• Ensure that your team takes proper minutes for the meeting so you know what to act on next. The minutes shall be posted on the wiki within 24 hrs of the meeting. Also specify in the minutes the role of each team member during the Iteration 2 meeting.
• Bring up any important issues regarding the future direction of the project during the meeting with the TA. Iteration 2 is a good time to discuss the scope of your project (because your team has a better idea on the pace of the planning game) and set the expectations for your final product.

Again, as a reminder, the final project is not just about the final product but also your team’s software development process. So you need to demonstrate that you are following XP closely as your software development process and that you are making steady progress toward completing your final project.

Checklist

• Do you have a meeting agenda?
• Have you decided on your MARS roles? Make sure each student has a different role from the last iteration. For an iteration meeting, the moderator role will be taken by the team/development leader for that iteration.
• Have you frequently updated the Wiki?
• Have you frequently synced your commits with Git?
• Did you tag your work as Iteration-2 before the meeting?
• Have you identified risk items and their risk-management actions?
• Have you notified us of important technical issues in advance?
• Can your code be run on all your team members' computers, especially on the presenter's computer?
• Can your tests be run on all your team members' computers, especially on the presenter's computer?
• Has each pair made at least 2 Git commits?
• Has each pair written (or updated) at least 2 automated tests?
• Does your commit message include your pair's name or netid? Did you say what your pair did in the commit message (you can just refer to a task or story on Wiki, need not copy all the text)? Also, you shall not have any blank commit messages.
• Has your plan been updated with better estimates?
• If you're presenting with a Mac, do you have a VGA adapter?
• Are the meeting minutes posted on the Wiki within 24 hours? Your minutes shall have enough details to recall what was discussed in the meeting.
  For example, "discussed user stories" is not detailed enough, but "the course staff supervisor suggested that user story 1 shall be broken down more" is better.
• Are you prepared to answer questions about each Git commit log?
• Did you make enough progress towards the completion of the final project?