Final Presentation

Please sign up for the Final Presentation 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

Important procedures

For the period after Iteration 3 demo meeting and before the final presentation meeting, there will be only one role for the team: the team/development leader. This team/development leader will be selected by the team from the team/development leaders for the past three iterations.

The final meetings are between Dec 10 - Dec 14 (Monday-Friday).

The Final Presentation meetings will last likely for 60 minutes for each team. So plan for this length.

Before the presentation

You must turn in the following:

1. A tagged copy of your code and tests in the repository. Everything necessary to run your program shall be in there. Please see the Final Code Submission page.
2. Documentation in PDF format (not word/ pages/ open office/ etc). Please see the Final Documentation page.

During the presentation

Ensure that your team comes prepared to the meeting. Please have an agenda which MUST cover the following:

1. Follow the MARS roles that we have been using for the Iteration meetings; since we will be looking at code, everyone shall have something to say about the portions that he/she worked on. For the final presentation meeting, the moderator role will be taken by one of the team/development leaders for the past three iterations.
2. Briefly introduce the main goal of the project.
3. Describe the new proposed use case and the distribution of the user stories across the iterations to give the course staff a quick overview of the whole project.
4. A demo of your final deliverable showing how it works. Be sure to demonstrate all the features that were implemented (please mention those things that are not working fully as well).
5. A walk through of the final code. This code shall be clean and well-refactored. We won't have time to go through all the code but each member of the team shall be prepared to talk about some portion of the code.
6. A short demo of your wiki page which will serve as evidence that your team actually had a well-defined process throughout the project. See below for details.
7. Run the “git-log” command to see your team's commit messages. This will serve as evidence that all team members contributed, made steady progress and did not just rush things towards the end.

After the presentation

Each student must turn in a peer evaluation form within 24 hours of the presentation. More details below.

Important Evaluation Criteria

We are going to use the contents of your team's wiki and Git repository to evaluate whether you followed the XP process. We are looking for the following:
Wiki

- Planning game (division of user stories into iterations).
- User stories (description of user stories similar to the suggested sample format i.e., distribution of user stories per iteration, break down of user stories into tasks, proper estimates associated with each user story).
- Minutes of each iteration meeting as recorded by the scribe.
- Other evidence to convince us that you followed the XP process: meeting schedules, chat logs (e.g. Skype/Google chat logs), etc.
- The wiki has a history system so we can check what/when changes were made. It would be suspicious if a multitude of drastic changes were made during the last 24 hours prior to the Final Meeting.

Git

- We will check your team’s Git logs to see what/when changes were made. It would be suspicious if a multitude of drastic changes were made during the last 24 hours prior to the Final Meeting.

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 tests pass, then refactor the code and commit the changes. That way you can show you're following the test-code-refactor cycle.

Peer/Self Evaluation

Peer Evaluations are confidential; your team members will not know what you write about them. So please be honest when you do the evaluations. Additionally, you shall write a personal evaluation of your own contributions to the project.

Note that the evaluation.txt template can be downloaded here. Each one of you must download this file, and replace <netID 1>, <netID 2>, ... <netID 7> with the Net IDs of your project teammates, respectively, and answer the questions asked and commit it to be your private git repo folder (NOT the team project git repo folder) under a subdirectory named as "peereval" once you are done.

Grading Rubric

The final submission is worth 50 points.

Process (25 points)
- Must have followed XP. To evaluate that, we will check all the items mentioned in the Important Evaluation Criteria section.

Code (10 points)
- Must address issues from the Final Code Submission page.

Tests (10 points)
- Must have tests for each task.
- Manual tests are acceptable only if discussed beforehand with the course staff supervisor (and only under very special circumstances).

Documentation (3 points)
- Must address the goals/issues from the Final Documentation page.

Packaging (2 points)
- All necessary files (including code, tests, documentation, configuration files, etc.) must be in your team's SVN repository.
- If certain dependent libraries are not in the repository, then there must be proper instructions on where to retrieve them.

Misc (no point allocated but possible point deductions depending on severity)
- Wiki shall be up-to-date; minutes of all meetings with the course staff must be on the wiki.
- Any remaining issues with the team/course staff need to be fully resolved.