Iteration 2

Tag your code: we'll only grade what's tagged.

Meetings for this iteration should be scheduled from Feb 17 to Feb 22 (Wed-Monday).

This page in a nutshell:

- Iteration 2 is similar in format to iteration 1, except with different deliverables (see below).
- Someone else should be moderator and scribe this time. We aim for everyone to be in each role at least once throughout the semester.
- Your team page should be updated to reflect your progress (see below).
- Convince us you’re following a process – it’s part of your grade.
- We want some code this time. They should be in your repository and tagged.

Meeting Schedule

As discussed during your Iteration 1 meeting, you are expected to identify the meeting times and dates for the remainder of your iteration meetings/presentations. This should be completed as soon as possible. See the meetings page for your team's TA/Professor (or email them directly) for more details.

Iteration 2 Deliverables

We are looking for the following during your Iteration 2 presentation:

- You need to convince us that you are following the process that your team agreed on. Remember that in cs428 we grade the process not only the product.
- You should have addressed any issues that were discussed during the previous iteration meeting. This includes making sure that your wiki page has been formatted properly, ensuring that your user stories are well described, setting up any resources necessary for your project, updating your future plans and any other issues that were brought up during the first meeting.
- The wiki should be updated with your progress for the current iteration:
  1. You should merge use cases from various pairs into one common set of use cases.
  2. You should revise your user stories and use cases as necessary.
  3. You should divide your user stories among your iteration plans (which user story is going to be done in which iteration) and update your estimates accordingly.
  4. You should update your plan and estimated time for completing the remaining tasks.
  5. You should provide actual time for tasks you completed.
  6. You should reevaluate project risks, etc.
- You should have at least a spike (and preferably much more code) to demonstrate that you have things going. For this iteration, your spike could be a sample project that you have modified.
- Since you need to have code, you'll also need to have automated tests for that code. Please use the testing framework that your team agreed on.
- Your code should also be checked into your repository (this doesn't have to be the SVN repository from EngrIT, but you should be actively using some version control system). Follow the best practice patterns for version control: use a trunk/master properly, don't check in anything that can be easily generated, keep the repository organized, etc.

Iteration Meetings

- It is expected that teams keep notes about their meetings (especially, but not limited to, the iteration meetings). What did the team accomplish? What are the issues causing trouble for the team? What needs to be done? So for each meeting, someone in the team will be the scribe. The notes can be in the form of minutes or anything else you agree on with your TA. Look to your peers to see which method works for them and for you. The bottom line is to create a transparent project that the TA can follow and more importantly that your teammates can follow to ensure excellent team communication. Leverage the wiki and the tools you have chosen to their fullest extent. These notes need not be excessively long (longer is not always better), but they should contain enough detail so that your team members and the TA know what is going on.
- There should be a different leader and scribe for each iteration meeting. Everyone should be able to and is expected to contribute to the discussion.
- All team members should attend all meetings. Any absences should be arranged ahead of time with the team and with your TA.

Documentation for Your Project... Coming Soon
As part of your final project submission you will be expected to compile a documentation for your project and the software you developed. We will provide more information about the different types of information that will be expected in the near future. It should not be a surprise, however, that this documentation will include information on the requirements and design of your system, as this is an important aspect of the course. As you watch the lectures, read the required readings, and do the homeworks on requirements and design, reflect on how this material should be applied to your project. For instance, you should expect the requirements and design portion of your project documentation to include use cases, class diagrams, and sequence diagrams as discussed in the lectures.