

April 4th, 2017

Software Engineering – The Final Mile

This document summarizes the deliverables each team needs to make between now and the end of the Semester. All grades from individual assignments (worth 50% of your grade) as well as the first two project deliverables (worth 20%) are posted on Sakai. Please check for any errors ASAP.

Check list:

All of these items are described below.

Due Date	Item	Where to submit	✓
Weds 3 rd May	Project Presentation	Turn slides into dropbox. Make sure Github and Jira are up to date.	
Weds 10 th May	Team Final dossier	Create either a pdf or html files. Upload as a zip or single pdf to dropbox. Github and Jira must also be up to date. Let us know as soon as you have done this as I'm going to ask the TA to check the projects out and try to run them. Releasing a bit early allows time for problem solving!	
Weds 10 th May	Individual Retrospective	Submit via Sakai assignments.	
Optional	Post-course survey	I would like to get your suggestions and feedback for how to improve the class for its next offering. I will send out a survey link via Qualtrics (anonymous) right after grades are submitted and ask for your anonymous feedback. (Thanks!)	

Remaining Deliverables:

1. **Weds 3rd May:** Final Sprint and Project Presentation.
The grades for GP-3 Checkpoint (from our schedule <http://sarec.nd.edu/courses/SE2017/ClassSchedule.pdf>) will be merged with the Final Product presentation. It is therefore worth 10 + 5 points (15% of final grade).

Specifically, you are expected to:

- a. **Complete your third sprint.** If you have any questions or concerns about the scope of work you will be able to complete by the end of the third sprint – please talk with Professor Huang during group time in class or in the office. **By now, you should know exactly what you are able to complete for your final deliverable.**
- b. **Conduct 2-3 field tests:** As discussed in class, this means that you should find a couple of people to try out your product. (Note: I plan to bring 1-2 people to class on Wednesday who understand this domain). If you discover anything really problematic you can (i) add new user stories or tasks to your Jira backlog, and/or (ii) if you have time, fix the problems. You probably won't have time to fix the problems you identify, but I want you to experience how users interact with your system.
- c. **Unit Tests:** We will discuss unit tests in a bit more depth on Wednesday and provide some examples of tests we've developed for Dronology. We also plan to show you the build server that Michael has been setting up to run tests automatically. Coming on Wednesday in the first 30 minutes of class! Every team is expected to have unit tests for at least 2-3 classes. I am not expecting full coverage given the timeline for this course. A good rule of thumb would be to provide solid test case coverage of at least one class, or at least 5 individual test cases. If you have questions about sufficiency – please see the instructor.
- d. **GP-3 Checkpoint:** As in the previous checkpoints, I will check Github and Jira. For Github I will check for code quality, consistent activity from all team members, merged code, documented code, and features covered. In Jira I will check for activity by all team members in implementing user stories – and that the SCRUM board is accurate and updated

- e. **Team Presentation:** In the final week of class we will do all team presentations **on Wednesday between 5.00pm and 7.00pm**. Because we have an extra long class on Wednesday, you may leave class early on Monday if you wish. Professor Huang will be in class for the entire period as long as any teams remain. Details about Team presentations are shown below.

2. Final Project Dossier (Document)

Each team needs to turn in final project documentation. You can choose to submit either a pdf file or a collection of connected html files that launch from index.htm. You should include:

- a. Product overview and background
- b. Requirements (extract from Jira using the reporting features). You can present these as a list of user stories (or EARS) requirements.
- c. Your architectural/class design. Make sure it looks professional.
- d. GUI design – one possibility is to present this somewhat like a high level user manual (i.e. don't provide all the low-level details – but show the primary flow of a major use case).
- e. Some key classes – with explanation. (There is no need to print out all of your code, but you could showcase some main parts of it).
- f. Sample unit test cases.
- g. Include something about your process – tests conducted, Jira, Github etc. Provide a link to your Github accounts. (Let me know if you wish them to be made public).

Here are some examples from previous students projects (however, these projects were conducted over a 16 week period and so include more functionality than is expected of your projects).

<http://sarec.nd.edu/studioProjects/MedFleet/>

<http://sarec.nd.edu/studioProjects/OnAMission/>

This is due by Wednesday 10th May. Turn it in via your team's Dropbox account. If you turn it in sooner, please ping me via email – and I will check that it has all the necessary components.

3. Individual Retrospective

Each person must write an individual response of 900-1,200 words in response to the following prompt:

Software Engineering has been defined as the “application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software”. Summarize and discuss lessons you have learned about Software Engineering through this Software Engineering course – especially with respect to lessons learned in your project. What were the greatest challenges you faced – and how did you overcome them? What would you do differently if you were to engage in another project? Is there anything you wish you had known before embarking on the project?

This is due by Wednesday 10th May. Turn it in via Sakai assignments.

Team Presentations:

All presentations will be on Wednesday May 3rd – our final night of class.

Each team will have 8 minutes to present plus 4-5 minutes for Q&A, and each pair of teams (i.e. SAC1/SAC2, DRAT1/2, and Maint1/2) will have an additional <=4 minutes to present an introduction to the problem they addressed. TLE will have a total of 15 minutes to include everything. We will have a longer than usual class meeting on Wednesday 3rd to accommodate all of the talks.

Each project type (except TLE) will have the following schedule:

- 4 minutes to describe the problem you are solving. (Feel free to reuse your context slides from the earlier presentations)
- Team 1: 6-8 minutes presentation + 3-4 minutes Q&A
- Team 2: 6-8 minutes presentation + 3-4 minutes Q&A

TLE will get 15 minutes (3 minutes problem, 8 minutes presentation, 4 minutes Q&A). The (longest-case) schedule scenario is approximately:

5.00pm	Setup
5.10pm	Introduce Panelists
5.15pm	Maint 1 and 2
5.45pm	TLE
6.00pm	SAC 1 and 2
6.30pm	Design Rationale 1 and 2
7.00pm	Prize for best project

What to include in your Presentations?

The presentation provides your team with the opportunity to present your work to an external panel of Software Developers and other IT project stakeholders. (I'm finalizing the panel now and will let you know as soon as it is finalized). Plan your presentation around the following scenario.

Imagine that your team had been hired as summer interns at Prism Corp – a mid-sized company develops tools to support Software Engineers. They hired a number of summer undergrads to explore four different tool ideas – all of which they believe have potential for development as a saleable product. Now at the end of the internship they want to see which tool (or tools) to take forward into full production. Your job is to make a case that your tool is the right one for them to take forward.

Some specific suggestions:

- i. Describe the purpose of the tool (joint presentation).
- ii. Show an overall diagram (marchitecture, architecture, or UML diagram).
- iii. What is special about your tool? Is it particularly scalable, reliable, usable?
- iv. Describe the major use case(s) and showcase how your tool supports these features.
- v. If you have a recording of a field-test, consider showing a short snippet of it.
- vi. If you wish, you may talk about the process you used, who developed which parts etc.
- vii. What other features do you plan for it?
- viii. Why should your tool/plugin get selected for productization?

Make sure that everyone in your team plays some part in the presentation. Plan and practice in advance so that your transitions are fluid and the presentation is clear.

Presentation scores will primarily be awarded based on the following:

- Impressive functionality will help – but those points will primarily be assigned to other parts of the grading scheme.
- Excellent presentation quality – both oral and good use of slides. Clear message planned and communicated to the audience.
- Well-executed team coordination
- Knowledgeable Q&A