

CSE 403, Spring 2013

Group Project Specification

Your team has been funded to produce the software project outlined in your proposal. The "customer" hiring you to write the product is the wealthy firm, HuskySoft. The HuskySoft CEO intends to pay your team for its services by awarding points. HuskySoft has managers ("TAs"), one of whom will meet regularly with your group to discuss its progress.

The overall scope of your project currently includes the following deliverables and other graded items. However, we caution that HuskySoft sometimes changes its mind, and we reserve the right to adjust phases and deadlines if necessary.

Phase	Name	Points	Due	Description
0	LCO	10	Thu Apr 4 9:30am	Project Proposal ("Lifecycle Objectives"), <i>already done</i>
1a	ZFR	20	Sun Apr 21 11:30pm	Zero Feature Release Code
1b	SRS	40	Sun Apr 21 11:30pm	Software Requirements Spec and UI Prototype
2a	ALPHA	50	roughly Sun May 5	Initial Implementation Release Code
2b	SDS	50	roughly Sun May 5	Software Design Specification
3a	BETA	60	roughly Sun May 19	Second Implementation Release Code
3b	TEST1	50	roughly Sun May 19	Testing Resources #1
4a	V1	60	roughly Sun Jun 2	Feature-Complete "Version 1.0" Code
4b	TEST2	60	roughly Sun Jun 2	Testing Resources #2
5	???	20	roughly Sun Jun 9	A surprise!...
	total	300		

Part of your grade on each phase will come from performing the following activities during the project:

- In-person customer meetings once during each phase 1-4 (one meeting per group)
- Peer evaluation surveys once during each phase 1-4 (one submission *per person*)
- Weekly progress wiki posts **every Sunday by 11:30pm** (one post per group)

The next several pages describe some of these items in more detail. Future specs that will be posted on the class web site describe each phase and its expectations in detail.

(Potential) Specification Changes:

Part of the nature of software engineering is that things can (and often do) change while you are in the middle of a project. We reserve the right to amend or alter any contents of this or other documents during the quarter. If any such changes are made, they will be posted and announced clearly to everyone. We promise not to make last-minute changes to any phase unless absolutely necessary; any major changes will be posted at least one week before that phase is due.

Customer's General Project Requirements:

The customers do not know exactly what they want, but they do have the following general requests:

- The product must be an **Android** mobile application or a **Ruby on Rails** web application.
- The product must have some non-trivial **data component** and a multi-computer communication aspect. In other words, it should be network enabled, or connect to a remote database back-end, or be a client-server or web application, etc. If your original proposal did not carefully take this into account, please do so now.
- The product must have some meaning or context outside of computer science. For example, it cannot be a source control system or a web-based Java interpreter.
- The product should display the **HuskySoft logo** (which we'd like you to design) on its UI.
- Your product should have a way of generating (fictional) "**revenue**." A possible way to do this would be an ad-based approach, by providing space on your UI for an ad image. Note that revenue is not the same as profit.
- The product should be as **usable** as possible, even for people who are not expert computer users.
- The product must be **robust** against common errors such as invalid user input, lost network connections, etc.
- If your project is web-based, it must have a **public URL** that others can use to access it. It should work properly in most modern standard-compliant browsers (e.g. Firefox, Chrome, Opera) and not be tied to any one browser. If it is Android-based, you should have a public URL with an installable **.apk** archive containing your app.
- If your project is Android-based, it should run on the **latest SDK** of Android (4.2, "Jelly Bean"). You may choose to support prior versions if you like, but this is the one that is required to work.
- Your team must use all of the **software tools** mentioned below in developing your project; no exceptions.

Beyond these requirements, you are largely free to make decisions of your own. You should, however, talk to your customers as you plan this project in order to make sure your product meets their needs. For full credit, you should discuss your proposed project details in some way with your customers before submitting them.

Required Software Tools:

You *must* use all of the following software tools on your project.

- **Git:** Version control system.
- **GitHub:** Online repository where you can check in your files.
- **Phabricator:** Project management tool for diffs, code reviews, bug tracking, wiki, and more.

If your project is web-based using Ruby on Rails, or your Android app uses a remote database, you should also use:

- **Heroku:** Web/database hosting for dynamic web applications

Part of your grade comes from using these tools appropriately. In particular, version control is not simply a file backup tool. Do not check in work to your group's Git repository that does not compile and run properly, or code that does not fully implement some new feature, fix, or otherwise represent a complete work task. Incomplete work should not be checked in. Checkins should also be code reviewed by another group member in Phabricator to assure code quality.

GitHub, Phabricator, and Heroku require the user to log in. You must set up your group's resources so that the instructor and all TAs are able to **access your resources** with full permissions. This is very important so we are able to evaluate your work. You may lose points if you do not properly provide access to these resources in a timely manner.

Brief **tutorial** information about these tools will be provided by the instructor and TAs to help familiarize you. You are welcome to use other tools in addition to the above, but not ones that essentially replace the above tools. Please ask us if you are unsure about the suitability of a particular tool.

In-Person Customer Meetings and Discussions:

This document is a partial specification for your project's requirements, but much information is intentionally left out. This is to encourage your group to **ask questions** of "the customer." You may ask these questions in lecture, by email, or on the message forum. Major turns that do not reflect questions with the "customer" may not receive full credit.

At various points during the project, your group should arrange to **meet with your primary customer(s)** to discuss its progress. This meeting will count as a small portion of your group's grade. Your group should come prepared to discuss what has been done, what is left to do, what is likely to be left out, any current problems or risks, and some specific questions you have for the customers as the release draws near. Your "project manager" plus at least two other group members must be present at each of these meetings.

The exact ranges of dates and times for these meetings will be announced in class and on the web site.

Peer Evaluation Surveys:

At the end of each "phase" of the project, each member of the team must **fill out a survey** about the team's progress and the contribution of each group member. This information will help us track potential group issues and make sure that everyone is contributing successfully to each project. Part of your grade will reflect whether you've filled out this survey.

Weekly Progress Wiki Posts:

Every week by Sunday night, your group must make a **"weekly progress report" post to your Phabricator wiki** that describes what your team is currently working on. Write this post as though your audience is the customer TAs. In your post each week, briefly discuss the following information:

- Your team's progress so far.
- What each member of your group is working on this week.
- What transpired at your team's latest in-person meeting (your team is expected to meet at least once a week) (this is also sometimes called the "minutes" of the meeting, though you do not need to list every small detail).

Submission and Grading:

Most phases will be considered **"turned in"** when you check in your code to your GitHub. We will grade your work by grabbing the latest version of the files from your repository unless you instruct us to do otherwise.

Some phases may require written documents to be turned in on paper in lecture. Make sure your project's name and all group members' names appear atop each document. Only one copy of documents should be submitted for each group.