

CSE 403, Winter 2012

PHASE 1 (40 points): Software Requirements Specification (SRS) and UI Prototype

parts 1,2 due Fri Jan 21, 11:30pm; part 3 due Mon Jan 23 at start of lecture

The SRS milestone artifact is a set of documents related to your project's requirements and high-level user interface (UI) design. Your SRS must contain the following three (3) major items:

1. Requirements Outline

Submit a document, 3-5 pages in total length, with concise yet descriptive answers to each of the following areas:

- a. **Product Description, Audience, and Feature Set:** What is your product? What is the target audience you expect to use the product? What are its major features? Include at least 4 major features you will provide, along with at least 2 other minor features or aspects you hope to complete.
- b. **Software Toolset:** What programming languages, data sources, version control, and other tools will you use? What, if any, software components will you attempt to use "off the shelf" versus implementing them from scratch? How well do your group members already know these tools and languages, versus how much will they have to learn? Explain why you chose these tools and languages, as well as why they are suitable for use on this project.
- c. **Group Dynamics:** For the most part, your group organization is up to you. However, **we require that you choose a single person to serve as your Project Manager (PM)**. Who will be your project manager? What will be the other members' roles? Will everyone share in the development, or will you have designated designers, testers, etc.? Why have you chosen these roles? If a disagreement arises, how will it be resolved? Be specific. Also justify your decisions by briefly explaining why you have chosen this structure.
- d. **Documentation:** What external documentation will you provide that will enable users to understand and use your product? This could take the form of help files, a written manual, integrated help text throughout the UI, etc. (Comments and Javadoc in source code don't count; they are developer docs, not user docs.)
- e. **Schedule / Timeline:** Provide a rough schedule for each member or sub-group within your team. For example, how long you believe your developers will spend working on each major feature listed in your product description? Who will work on the design, and how much time do you expect it will take? Which features are "beta" features? Your estimations should take into account the posted rough due dates for each major phase of the project. Provide reasonable guesses as much as possible, but you will not be graded on the exact accuracy of these predictions.
- f. **Risk Summary:** Describe at least three specific adjustments you will make if the project begins to fall behind schedule. No more than two of the adjustments you list can be feature cuts; at least one must be some other change or cutback, such as changing specific areas of testing, adjusting your group dynamics or time schedule, etc.

2. Use Cases

Submit the following use case documents:

- a. **Two (2) formal use cases** for scenarios you think are two of the most important to your product. They should be similar to Use Cases 1, 2, 3, and 5 from Cockburn's paper, following a similar syntax and actor/system interaction. The cases should include: primary actor, level, preconditions, minimal/success guarantees, a list of steps to the success scenario, a list of properly numbered and meaningful extensions, and a failure-handling remedy for each extension as appropriate. It is impossible to think of every possible failure case ahead of time. But you should list a reasonable set of extensions and remedies if reasonable ones exist. If you do not have a known remedy, your use case should state this and explain what will be done to investigate possible remedies.
- b. **One (1) casual use case** in paragraph form for one other scenario that you think is important to your product, perhaps of lower importance than the two you chose to depict formally. This use case should be similar to Use Case 4 from Cockburn's paper, describing the main success scenario first in paragraph form, then listing each extension and its remedy (if known) in a second paragraph.

(continued on back)

3. User Interface Prototype

Submit diagrams containing rough sketches of your product's user interface. These diagrams should depict the major UI used to complete the use cases you submit. For example, if one of your use cases is to Purchase Stocks, you should draw the initial UI that is presented when the user wishes to purchase a stock, along with any other major windows, messages, etc. that appear as the user navigates through this use case. From looking at these UI diagrams, the customer should be able to clearly see how your product will be able to successfully complete each use case you are submitting. Briefly represent paths through some major extension(s) as appropriate in your UI diagrams.

Submit as many prototype screens as are necessary to cover your use cases; this must be **at least two (2) UI diagrams**. A "diagram" is a depiction of a complete major screen, web page, window, etc. of your system. The diagrams can be drawn by hand or computer, or can be screenshots of an actual prototype. (Since it often requires a lot of time to create an actual code prototype, we generally do not recommend doing so for this deliverable phase.)

If a window leads to a dialog box, drop-down box, etc., include it as a sub-diagram within one of your two overall diagrams. For full credit, there should be at least some dynamic content to your prototype, such as expanding lists, pop-up dialogs, etc. as described in the reading.

To help the grader/customer understand how to "use" your prototype, it would be helpful if you labeled which items and sheets go together, for example, by putting numbers or names on the back of papers. For sub-diagrams such as pop-up windows, drop-down boxes, etc., it is helpful to put a corresponding number/name in the place that sub-diagram appears. If there is anything non-obvious about how your UI is used, you may include brief written instructions about how the user is expected to walk through the UI.

Your diagrams do not need to be beautiful to get full credit, but they should be legible and reflect some forethought about what options will need to be shown and how the user will use the software. Part of your grade comes from choosing appropriate UI elements to effectively accomplish the desired task.

(Your group will not be "locked in" to using the exact UI represented by these diagrams on your actual final project. The purpose of a prototype is to try out a set of user interface ideas, see how they work, and revise/modify the UI as needed.)

Submission and Grading:

Submit your Requirements Outline and Use Cases online through the link provided on the course web site. You may submit your UI prototype online as well if you like, but since it is likely to be on paper, you may also bring it to lecture on its due date to turn it in at the beginning of lecture.

Part of your grade will come from the plausibility, thoughtfulness, and level of detail of your work. For example, if you are listing features, take care not to forget important aspects that would reasonably be required of a project such as yours. When listing software tools, list a plausible set of tools for all aspects of the project and defend your choices. When listing group dynamics and a schedule, choose a group strategy that makes sense for your team and for the time given.

In use cases, sloppy or incomplete work often leads to deductions. We want to see that you have given due thought by choosing substantial use cases that are important to the core functionality of your product. You should also list a reasonable set of steps in the various scenarios that can occur in these use cases. Take care not to omit important steps or details. Make sure that the state of the system at the end of any path through the use case matches the guarantees that the use case claimed would occur. A portion of your grade will come from the format of your use cases. It should match that described in the reading in content structure, ordering, numbering, and style.

A small part of your grade comes from the looks or aesthetics of your documents. They do not need to be beautiful or excessively formatted, but your customers need to be able to read them and extract information from them. This means they should be clearly written in a professional writing style in complete sentences as appropriate, with proper spelling and grammar, clear wording, and formatted with a enough organization to present your ideas clearly to the reader.

Remember that part of your grade comes from having a meaningful in-person interaction with your customer before the phase is due to show your progress, ask questions, get feedback, and generally make sure you are on the right track.