CSE 430
Autumn 2013

Requirements (Phase 1b)
Due Friday, October 18, 2013

Objective

Establish a solid definition of your project from which to base a design and an implementation. This definition acts as a reference for developers and an informal contract between the team and the "customer" (cse430 staff and/or external clients if applicable).

Background

The requirements document (sometimes referred to as a Software Requirements Specification or SRS) describes the main goals of your project and the main features of the product your team will develop. While your project involves ultimately building a software product, you will also need to organize and manage the process of how you do it as a team (Phase 1a touched on some issues of team organization).

Note that sometimes the user interface specification is included as part of an SRS document; in this course I have made it a separate deliverable (see Phase 1c). The most flexible software separates what the software does from how it does it. If you can keep form and function separate, then you can swap out different user interfaces more easily than if they are tightly intertwined. Additionally, the user interface prototype may be drawn on paper.

Assignment

Note that the requirements document in this course will be treated as a "living document". Therefore, you will be asked to provide updates to it at periodic points in the development cycle. For ease of editing and distribution, this document must be in electronic form.

Provide concise yet descriptive answers to each of the following questions:

Product Description:

- What is your product?
- What (generally) does it do? (What problem does it solve?)
- What is the target audience you expect to use the product?
- What are the alternatives? Indicate their pros and cons, and explain why yours is different and better from the user's perspective.

Software Toolset:

- Will you use any software tools or libraries in addition to the required ones?
- How well do your group members already know the tools and languages you will be using, versus how much will they have to learn?
- Briefly explain why you chose these tools and languages, as well as why they are suitable for use on this project.

Feature Set:

What are your product's major core features? Include at least 3–4 major features your product will provide, along with at least 2 other minor features or aspects you hope to complete.

Schedule/Workload:

You will have roughly 2 weeks to work on each real coding phase of the project: the initial Alpha version, the second Beta version, and the third "Version 1.0" (V1) completed version.

- How will you divide up the work between these phases?
- Why have you chosen this particular division of the labor?
- For each of the three phases, which features do you expect to be complete, which will be started, and which will not yet be started?
- Briefly describe any dependencies or relationships among the features that would impact scheduling. While not required at this time, you may consider a Gantt Chart (http://en.wikipedia.org/wiki/Gantt_chart) to help you visualize your project's schedule.

Risks, Cuts, Adjustments:

It's very difficult to predict exactly how much work a group will finish in a given amount of time.

- What are you most worried will prevent your team from completing the project?
- Describe at least three specific adjustments you will make if the project begins to fall behind schedule.
    - Two of the adjustments you list can be feature cuts, but 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.

Use Cases Documenting Requirements:

Write two (2) formal use cases for scenarios you think are two of the most important to your product. (Refer back to the product description.) They should be similar to Lecture #5, slide 13, following a similar syntax and actor/system interaction. The cases should include: primary actor, level, preconditions, triggers (what starts the use case), minimal/success guarantees (end condition), 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.

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 Lecture #5, slide 10, first describing the main success scenario first in paragraph form, and then listing each extension and its remedy (if known) in a second paragraph.

A use-case tutorial is available at http://www.cragsystems.co.uk/use_case_tutorial/

Submission and Grading:

Submit your requirements document online using the appropriate Drop Box linked on the course website. 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 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 describing workload and 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 TA before the phase is due to show your progress, ask questions, get feedback, and generally make sure you are on the right track.