Project Milestone #2:
Detailed Planning, Specification, Design, and Test Plan

Goal: To refine the scope and feature set of the accepted project ideas, to perform detailed design, and to propose a workable test plan for your product

Due Dates:

Part I: In-class presentation – Wednesday, July 13, in class

Part II: Electronic submission – Thursday, July 14 @ 10pm

Submission: A turn-in link will be available on the course web shortly.

Introduction

The purpose of this group assignment is to accomplish the detailed planning, specification, and design required before you implement your ideas in code. In addition to the deliverables described in Boehm [2] and elsewhere, we ask that you prepare a test plan [5] that reflects your vision of what you are going to test and how you are going to test it. This is needed because testability is often found to be highly correlated with design quality – it you can't test it easily, there is likely some problem in the design.

Your team's results at the end of this stage will be presented as the Lifecycle Architecture (LCA) review milestone. You need to convince your audience that you (a) understand well what it is that you are building and (b) have a solid idea how to approach building it. For more background material on the content of the LCA review, refer to the in-class discussions on the topic, including requirements gathering and specification and design techniques, as well as the reference materials cited below.

Deliverables

Much of the material for this review is an elaboration of the material used in the LCO review. You can draw on that as a starting point but do not feel compelled to stick too close to it, especially if you believe that changes are necessary to improve its focus and/or scope. Since your goal at this stage is to accurately define the actual product to be built, the result of the LCA milestone should leave fewer options and open items (as compared to the LCO deliverable), and contain more decisions and detail.

We will expect to see the following documents:

1. Overview presentation. A set of PowerPoint presentation slides summarizing the LCA elements for your product. This is the pitch that your team will give to the class. As there will be two teams to get through in 1 hour, time your pitch to last about 20 minutes to allow plenty of time for follow-up questions.
In wrestling about what to include versus what to skip in your presentation you will likely ask yourself the same kinds of questions that your audience will wonder about as they listen to your pitch. One such question is, "What are the key aspects that the audience (managers, fellow developers, customers, etc.) needs to understand in order to decide whether this project will succeed in bringing value to the company and its customers?"

2. Specification document. This document should accurately and as completely as possible reflect the product you are building from the point of view of both the customers (what they want to see built) and the product administrators (what service modules they will need in order to correctly run your product, even if the customers don't see directly many of these). See the lectures and the below referenced article by Joel Spolsky [3] for suggestions on content. Also, see the articles on writing specifications by Alistair Cockburn, cited on the course web under 'Useful Resources'.

3. Architecture document. This document is a detailed definition of the system and software components. It should carefully and clearly identify the modules and interfaces between modules required to implement the system [4]. (The modules should be specific, not just "client", "server", or "GUI".) Difficult or high-risk areas

of the design should be clearly identified along with an analysis showing why you believe they will not become "show stoppers" for the project. This section should include both the design of the system as seen by the customer, as well as that of any administrative modules.
Hint: Good interface definitions are invaluable at shedding light on the quality of a design.

4. Team structure, schedule, and task assignments. This document should describe your team structure (how you have organized the team, what roles different people play, etc.), and elaborate on milestones (external and internal), task descriptions (broken down to a reasonable level of detail, e.g., not just "server functionality", but "server logging functionality and keeping track of game scores across all clients") and initial task deadlines, as well as the team member(s) responsible for each individual task. This should reflect your *actual* plan of work, possibly including items that your team has already completed (e.g., while preparing for the LCA milestone).

5. Test plan document. This document should describe what aspects you plan to test and what not, why this is reasonable for your product from the point of view of the customers, as well as how specifically you plan to test those aspects in a disciplined way (as opposed to just playing with it randomly to see if it eventually breaks somewhere). This implies specifying a set of actual test cases by following a methodology like the one described in [5].

Note: Some of the above issues have not been covered in class yet, so it is normal that they may sound unfamiliar to you at this point. By the time the LCA milestone arrives, we expect to have discussed in class all the necessary parts.

Mechanics

For this and all remaining group assignments in the course, you will be working in your team of 5-6.

While you are not strictly limited to any fixed (maximum) number of pages for the individual documents, please keep in mind that conciseness is a virtue and you don't want managers or other developers like you to find it daunting to read long pages and as a result to altogether skip reading what you have written.

Note that the LCA presentations *precede* the actual submission deadline by a day. The intent is that you prepare your materials and present them to the class, get preliminary feedback on it, then adjust your final submitted documents by reflecting the concerns that have been raised by the audience.

As a follow-up to the LCA submission, we will schedule 30-minute informal discussion sessions on Monday, July 18, for post-LCA reviews with each team.

Evaluation

In our evaluation of your work, we will be looking to see that you have addressed all the necessary elements of an LCA review and have made reasonable decisions related to each. (Refer to the lectures, the Boehm paper [2], and the test plan article [5] for more information about what is needed.)

Technological Resources

The department gives us access to web servers and database servers for your use, if needed.
If your product will depend on the availability of special software, be sure to ask upfront as we may or may not be able to get you access to it and in all cases will need time to check and make arrangements.

Turn-in

Please have one person from each team do the turn-in so that all files from your group appear in the same place. The turn-in link will appear on the course web page shortly.
Note: eSubmit, the UW Catalyst tool we use for managing electronic submissions, accepts files of size up to 2MB each, so plan accordingly.

Resources

Rapid Development, Steve McConnell.

Anchoring the Software Process, Barry Boehm (USC)
http://citeseer.ist.psu.edu/boehm95anchoring.html

Painless Functional Specifications, Joel Spolsky
http://www.joelonsoftware.com/articles/fog0000000036.html

Software Architecture, David Garlan
http://www-2.cs.cmu.edu/afs/cs/project/able/ftp/encycSE2001/encyclopedia-dist.pdf

How Do You Spell Testing? – A Mnemonic to Jump-Start Testing, James Bach
http://www.satisfice.com/articles/sfdpo.htm