Project Specification (Group)

Due: Friday, March 2, 2001

Overview

The goal of this assignment is to learn how to write a specification document. You will include a description of any functionality your product needs to provide as well as any of the constraints the system will have to meet. A common mistake is to start thinking of the "how." Focus here instead on the "what." You will use the specification to drive the architectural design you produce for the next group assignment.

Specification Document

Get together with your teammates and further develop your project idea. Take the CS160 Rapid Prototype and write-ups, together with your last write-up and figure out exactly what functionality will be required in your final system to meet the intended users' needs.

First, in the **Project Description** section, remind the reader about the ultimate goal(s) of the product you are developing. You can revise the Introduction from your 1st assignment according to comments given by the teaching staff or your manager.

Next, you will need to update the tasks & storyboarded scenarios that were used in CS160 with any changes you have decided the system needs. This means if you changed an existing feature, you may need to change an existing task. If you added a new feature, you might need to include it in an existing task or create an entirely new task that uses it. You should have at least five tasks in total. Please list all of the tasks here (old and new – highlight the updated portions) and include new storyboarded scenarios for any of the tasks that have changed or been added (you can sketch the storyboards online using a drawing program, DENIM, or on paper that you scan in). These scenarios are stories of how the system will be used and should be as concrete and descriptive as possible. Please perform **Use Case Modeling** for each of the tasks. This means you should create a Use Case Diagram for each task and detail a few scenarios for each.

The specification document is intended to detail several types of specifications for the product you will build. First, there are the **Functional Specifications**, which detail what basic operations your product will deliver. The tasks and scenarios mainly focused on the functional behavior. For the functional specifications we suggest that you first try to decompose the functionality of your system into self-contained features. For instance, a personal information management system will most likely contain a contact management facility, which is a self-contained feature. Describe and explain what each feature means, decomposing it into smaller features, if necessary. For example, for the contact management functionality, you might have editing, printing, and searching features.

Because your time is limited and you are not expert in predicting how long it will take to design and implement each of the features, it is important that you list the functional

features in **order of importance**. If one feature denotes a group of sub-features, then write these indented and underneath the main one (i.e., use the classic table of contents format). To the reader, it should be clear in what order the functional elements of the system will be implemented and which features depend on other features. For each function feature, say **how that function is to be evaluated or tested** in the final product.

Non-functional Specifications deal with the environment in which your product runs. This might refer to external standards that your system must adhere to, particular usability constraints (e.g., you might specify that 90% of test participants are able to complete all of your tasks within a set amount of time.) Non-functional requirements do not refer directly to services that the system will deliver; rather, they refer to the way in which those services must be provided or constraints that the environment of the system's use will impact on the development (e.g., "response to all user requests must be answered within 2 seconds in all cases"). Whatever you consider a non-functional requirement, it is mandatory that you be very specific about how that requirement is to be evaluated or tested in the final product.

A particular class of non-functional specifications are those concerned with the **Platform and Network Environment**. Here, you need to clearly indicate under which platform and network environment conditions you intend to operate on and any special development platform needs you might have.

Use the **Risk Assessment** section to inform us of the main risks that you see in developing the system. Remember that these span the substance of every section listed above (functional/non-functional specifications and platform). Assessing risk is important to help you identify what things might go wrong in designing the product you are proposing (such as required hardware not arriving or being faulty). For each risk, you need to identify what your plan will be to address the risk in the case it becomes a reality.

Here are some examples of what students in a software engineering class at Georgia Tech have created in the past. Note that they call these requirements documents and that there are other minor differences in the assignment.

Deliverable

Submit **two (2)** copies of a printed write-up of *about* **7 pages** of text in class (images are free in the page count). You must also put a copy of the write-up online. Your write-up should follow the outline below and will be graded using the guidelines detailed below.

1. Each team member's **name** and a **URL** to an online copy of this write-up
2. Product Description (1/4 page)
3. Tasks Descriptions (1/2 page)
4. Scenarios (1/2 page + screen shots)
5. Functional Specifications (4 pages or more if necessary)
6. Non-functional Specifications (1 page)
7. Platform and Network Environment (1/4 page)

8. Risk Analysis (1/2 page)

Writing and Experimentation Guidelines

Document Presentation (5 pts)

- Organization
- Document authors & URL
- Use of English

Project Description (10 pts)

Task Descriptions (10 pts)

- Describe 5 tasks and show any differences from 160 tasks

Scenarios (10 pts)

- Use images that show important interaction events on transitions
- Annotate images w/ other scenario details

Functional Specifications (25 pts)

- Number of functions
- Clarity of description
- Decomposition
- Test cases
- Prioritization

Non-functional Specifications (15 pts)

- Clarity
- Test cases
- Variety

Platform and Network Environment (10 pts)

- Target platform described
- Development platform described

Risk Analysis (15 pts)

- Risks identified
- Alternate strategies identified