User Manual

Objectives

The goals of this assignment are to:

(1) Fully understand *what* you are trying to build by precisely defining the external behavior of your FCS implementation
(2) Provide end-user documentation to your quality assurance team
(3) Give you a chance to experience the life of a technical writer

Introduction

The first step in building a piece of software is to understand *what* you are going to build. You must have *what* you are trying to build firmly in mind before trying to figure out *how* you are going to build it.

Requirements elicitation plays a key role in understanding *what* you are going to build. The high-level product requirements are captured in a Product Requirements Document (PRD). While the PRD constrains many aspects of the final product, it is high-level enough that there are usually many different ways to implement the product and still meet all of the requirements.

Among the many possible forms that the final product might take, we must make specific choices and determine exactly what system we intend to build. Therefore, the next step in understanding *what* we are going to build is to write a Functional Specification, which provides a detailed description of the final system's external behavior. The Functional Specification is much more detailed and specific than the PRD, but it still addresses *what* is to be built rather than *how*.

In this class, the PRD and Functional Specification for FCS have been given to you. However, the Functional Specification intentionally leaves out some details that need to be worked out before you will completely understand *what* you are trying to build. For example, the PRD says little or nothing about the user interface provided by the system. The purpose of this assignment is to fill in the missing details by writing a User Manual for your intended implementation of FCS.

The User Manual is the documentation provided to end users of the product. These people are often non-technical. They want to know *what* the product does and how to make it perform its functions, but they really don't care about *how* the program works internally. Because the User Manual helps people learn how to use the product, it must provide a very detailed description of the product's external behavior.

Assignment

Your assignment is to write a user manual for your implementation of FCS. Your manual must be detailed enough that a novice user with no background with FCS could pick up your manual, read it, and be productive using your implementation of FCS. It must explain how to install and execute the software, as well as provide details on how to use the program. This will serve two primary purposes:

(1) After writing your manual, you will understand exactly *what* you are trying to build

(2) Your quality assurance team will need this documentation to learn how to use your system

Writing a user manual for software that doesn't yet exist presents some challenges. First, user manuals usually include screen shots as part of their explanations of how to use the program's features. Since your software doesn't exist yet, you may need to create fake screen shots using a drawing program, or create a quick-and-dirty prototype of the program's user interface that can be used to make screen shots. Second, as you design and build your version of FCS, you will probably find that you want to change your mind about how users will interact with your program. The manual you turn in is not binding; you will be free to change these details as the project progresses, but you will be required to update your manual to bring it into synch with the current state of the software. Remember, another team will be relying on your manual to learn how to use your product.

Deliverables

One hard copy of your user manual

Grading

Your user manual will be graded based on how effectively it helps a novice user learn to install and use your FCS program. It will be judged on completeness (did you leave anything out?), and on the clarity of its explanations. The goal is to teach people how to use your software. Making your manual fancy is less important than making it communicate effectively.