

System Test Plan

Objectives

The objective of this assignment is to give you experience developing a *system test plan*. In the next assignment you will execute your test plan and report the results.

Assignment

The assignment is to develop a plan for performing a comprehensive system test of your development team's FCS implementation (i.e., you're going to write a System Test Plan). To do this you need a copy of your development team's User Manual. Based on the FCS PRD, Functional Specification, and User Manual, your task is to design a set of test cases that will thoroughly test your development team's FCS implementation.

The focus here is on *system* test. Therefore, you should design black box tests that will test the product from an external, end-user perspective. You will have only a relatively short time to plan and execute your tests, so you won't be able to include all of the possible tests that come to mind. Rather, you'll have to be selective in deciding which tests to include and which to leave out, always preferring tests that give you a lot of bang for the time spent. Specifically, you should focus on testing scenarios that real users are likely to encounter, not ones that are theoretically possible, but highly unlikely to occur in practice. You should prefer tests that don't just retest the same things that earlier tests have already covered (your tests should not be redundant). Instead, design tests that are likely to test different paths through the code and discover new bugs. The PRD and Functional Specification list many cases that should be tested, and you should be sure that your test plan accounts for all of them (your test plan should be traceable to the PRD and Functional Specification, and vice versa). Beyond that, you can include tests for scenarios that aren't specifically mentioned in the PRD and Functional Specification, but that real users are likely to encounter. An important goal of test planning is to achieve broad coverage of the product's features and not leave any functionality totally untested (similar to code coverage in white box testing, except from an external perspective). Your test cases should be designed for manual execution. You are not required to automate your system test cases.

You should document your test plan in a System Test Plan Document. This document will consist primarily of test case descriptions, but should also include a section that introduces the scope of the test plan and gives an overview of your general approach to testing the system. More specifically, your System Test Plan Document should contain at least the following items:

(1) Introduction & Overview

- a. **Introduction:** Precise description of the software that's being tested
 - i. Name, version, etc. with appropriate references to related documents such as requirements spec, functional spec, etc.
- b. **Exhaustive list of feature groups to be tested**
- c. **Approach:** Describe your overall approach to testing, main activities, techniques, and tools used

- d. **Environment:** Describe necessary hardware, software, tools, etc.
- (2) **Test Design Specifications (one for each feature group to be tested)**
 - a. A detailed description of what techniques will be used to test this group of features
 - b. How tests will be run and results analyzed (visually, comparison tools, etc.)
 - c. List of individual test cases for this feature group (list their identifiers)
- (3) **Test Case Specifications** (provide the following for each individual test case)
 - a. **Test case identifier:** Unique identifier for the test case
 - b. **Test case description:** Description of the test case, including features being tested and anything else that seems relevant
 - c. **Input Specification:** All inputs by value, range of values, file names, etc. Identify anything that's relevant as input for the test case
 - d. **Output Specification:** All output values and messages. If appropriate, include response times or information that reflects non-functional requirements
 - e. **Special Procedures:** List anything that's unusual in terms of setup, tester actions, or analysis of output

Deliverables

One hard copy of your System Test Plan Document

Grading

Your system test plan will be graded based on the following criteria:

- (1) Coverage: Does the plan cover all of the features/use cases mentioned in the PRD and Functional Specification (does it demonstrate traceability)? Does it cover other interesting scenarios not specifically mentioned in the PRD and Functional Specification? Would passing these tests provide a high level of confidence in the product's quality?
- (2) Test Quality: Do the test cases represent useful equivalence classes? Was boundary value analysis done effectively? Are the test cases overly redundant? Are the test case descriptions clear? Are the inputs and outputs specified clearly and completely? Etc.
- (3) Document Readability: Is the test plan well written? Is it easy to read? Is information easy to find?