

# CS 480 / CS 481 ~ PROJECT TEST PLAN GUIDELINES

The purpose of the Test Plan is to provide a description of all testing activities to be performed on your product. This includes (as appropriate) testing at the unit or component level, testing during subsystem and system integration, and testing of the completed system. You will perform most of the testing yourself, however, you are required to conduct a Beta Testing activity with someone other than yourself doing the testing. You may also want to have the customer or end user perform some testing.

Testing is performed for at least two reasons. The first and most important reason is to find defects in the code that could cause problems for the user. The second reason is to show that your product implements the requirements and that it works as intended under specified conditions. Remember that testing cannot **PROVE** that your product will always function correctly. Even after a significant amount of testing there can still be obscure defects that haven't been encountered. Through a comprehensive and effectively implemented testing process you want to reduce the risk of significant defects going undetected and you want to build customer and user confidence that the product will perform as intended.

As with other documents required in this course, please follow the outline provided. If a particular section or subsection is not appropriate for your project, include the section or subsection and then indicate a response such as "none" or "not applicable."

The following is an outline template for your System Test Plan.

---

Title Page Your Test Plan must include a title page indicating the name of the document, the project, the author, and the date prepared.

---

1.0 Purpose In this section provide a very brief statement of purpose that will indicate what information is contained in the document.

---

2.0 Overall Testing Approach In this section briefly describe the testing approach you will use on your project. This should include how you will test individual functions, modules, or components, how you plan to combine or integrate modules to form larger collections of code to test, and how you will generally approach testing at the system level. Be sure to indicate how you will ensure that your testing will provide a comprehensive exercising of the entire product and will have a high probability of finding defects that could cause problems for the user. You may find it appropriate to do risk-based testing. In this approach you concentrate a greater percentage of testing on portions of your system that are thought to have greater potential for defects or if that portion of the system fails has a more significant consequence for the user. The

usage scenarios that you documented in your initial User's Manual draft can be a good source of information for planning your system testing strategy.

In addition to functional testing you may need to conduct performance tests to show that your system meets specific performance requirements. You may also want to test your system under conditions in which the system is pushed to its limit of capabilities. If your system must run in several different environments, you need to test the system in these environments to ensure that it does function properly.

---

**3.0 Features To Be Tested** Provide a concise list of the features you will be testing. For each feature identify the general approach you will use to verify that the feature works correctly. You do not need to list every requirement in your SRS. Each feature can be representative of many related requirements. A full test case description (with specific inputs and anticipated results) is not expected, but the general approach you plan to take for each feature should be clearly evident.

---

**4.0 Feature Not To Be Tested** In some circumstances product features are not tested. This section must identify any features that you aren't planning to test. In most cases for CS 480 project this section should be marked as "None," however, if you are modifying an existing product you may want to indicate if some of the existing product's capabilities will not be tested. Be careful that you aren't leaving yourself open to potential problems due to interactions between existing code and code that you add to the product.

---

**5.0 Entrance and Exit Criteria** In this section describe for each level of testing to be performed, the entrance criteria that will be used to determine when the code will be ready to test. Also describe the exit criteria that will be used to determine when sufficient testing at each level has been performed.

---

**6.0 Regression Testing** Regression testing is testing performed after a component or system has been modified to determine that other capabilities of the component or system still function correctly. Briefly describe the level of regression testing you will perform on your product and the criteria you will use to determine when and how much regression testing to perform.

---

**7.0 Test Environment** Describe the physical computer system environment and software environment that will be used for testing your product.

---

8.0 Code Control Describe the mechanism that will be used to provide source and object code control during testing. Code control mechanisms are important in ensuring that you test the exact code you think you are testing.

Describe the process you will use to document tests performed on your product and suspected defects discovered during all phases of testing. Also describe the process you will use to ensure that each suspected defect is analyzed to determine if it is or is not a problem. Describe the criteria you will use to determine if a problem is significant enough to require a fix or can be categorized as a "no fix." For problems requiring a fix, describe how you will ensure that the problem is actually fixed, tested, and verified as being fixed.