**Software Test Plan (STP) Template**

Items that are intended to stay in as part of your document are in **bold**; explanatory comments are in *italic* text. Plain text is used where you might insert wording about your project.

This document is an annotated outline for a Software Test Plan, adapted from the IEEE Standard for Software Test Documentation (Std 829-1998).

Tailor as appropriate. Where you decide to omit a section, you might keep the header, but insert a comment saying why you omit the element.

**Smith-Day**

****

**image from:** [**https://en.wikipedia.org/wiki/FreeCol**](https://en.wikipedia.org/wiki/FreeCol)

**FreeCol Final Project**

**Software Quality Assurance Plan**

**Version: 1.0 Date: (05/16/2018)**

**Document History and Distribution**

1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision #** | **Revision Date** | **Description of Change** | **Author** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

**1.** **Introduction 1**

**2.** **Test Items 2**

**3. Features To Be Tested 3**

**4. Features Not To Be Tested 3**

**5. Approach 3**

**6. Pass / Fail Criteria 5**

7. **Testing Process 5**

**8. Environmental Requirements 6**

**9. Change Management Procedures 7**

# INTRODUCTION

**The Software Test Plan (STP) is designed to prescribe the scope, approach, resources, and schedule of all testing activities. The plan must identify the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan**.

**1.1 Objectives**

The approach that we are planning on taking is more of a high level approach to the software. We are looking more at the parent classes which control more of the software than each individual classes, which could result in extensive testing.

**1.2 Testing Strategy**

**Testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item.**

The general level of testing our team decided to do was a broad, parent/controller class testing. This allows us to provide a more general level of testing, which in turn should allow us to make overall changes to the program, without having to test each individual item within the code. In regards to the general classes, we are planning on testing the general controller class, as well as the ones for the server, the GUI and the client. We are using the JUnit tests that were provided with the code a our benchmark to make sure that the code still operates as expected. We understand that this does not entirely check to make sure that the code is correct, but it will give us a definite marker for how the code should be operating.

**1.3 Scope**

**Testing will be performed at several points in the life cycle as the product is constructed. Testing is a very 'dependent' activity. As a result, test planning is a continuing activity performed throughout the system development life cycle. Test plans must be developed for each level of product testing.**

Our tests will be the control classes that control a large majority of the code. In this sense, it should allow us to test a large scope of the code rather than testing each individual class. Version control will be done from GitHub.

**1.5 Definitions and Acronyms**

*(Specify definitions of all terms and agency acronyms required to properly interpret the Software Test Plan. Reference may be made to the Glossary of Terms on the IRMC web page.)*

# TEST ITEMS

*(Specify the test items included in the plan. Supply references to the following item documentation:*

* *Requirements specification,*
* *Design specification,*
* *Users guide,*
* *Operations guide,*
* *Installation guide,*
* *Features (availability, response time),*
* *Defect removal procedures, and*
* *Verification and validation plans.)*

In this document, we will include:

**2.1 Program Modules**

*(Outline testing to be performed by the developer for each module being built.)*

**2.2 User Procedures**

*(Describe the testing to be performed on all user documentation to ensure that it is correct, complete, and comprehensive.)*

# 3. FEATURES TO BE TESTED

*(Identify all software features and combinations of software features to be tested. Identify the test design specifications associated with each feature and each combination of features.)*

# 4. FEATURES NOT TO BE TESTED

*(Identify all features and specific combinations of features that will not be tested along with the reasons.)*

Any features that originated from

# 5. APPROACH

*(Describe the overall approaches to testing. The approach should be described in sufficient detail to permit identification of the major testing tasks and estimation of the time required to do each task. Identify the types of testing to be performed along with the methods and criteria to be used in performing test activities. Describe the specific methods and procedures for each type of testing. Define the detailed criteria for evaluating the test results.)*

*(For each level of testing there should be a test plan and the appropriate set of deliverables. Identify the inputs required for each type of test. Specify the source of the input. Also, identify the outputs from each type of testing and specify the purpose and format for each test output. Specify the minimum degree of comprehensiveness desired. Identify the techniques that will be used to judge the comprehensiveness of the testing effort. Specify any additional completion criteria (e.g., error frequency). The techniques to be used to trace requirements should also be specified.)*

**5.1 Component Testing**

*(Testing conducted to verify the implementation of the design for one software element (e.g., unit, module) or a collection of software elements. Sometimes called unit testing. The purpose of component testing is to ensure that the program logic is complete and correct and ensuring that the component works as designed.)*

**5.2 Integration Testing**

*(Testing conducted in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated. The purpose of integration testing is to ensure that design objectives are met and ensures that the software, as a complete entity, complies with operational requirements. Integration testing is also called System Testing.)*

**5.3 Interface Testing**

*(Testing done to ensure that the application operates efficiently and effectively outside the application boundary with all interface systems.)*

**5.4 Security Testing**

*(Testing done to ensure that the application systems control and auditability features of the application are functional.)*

**5.5 Performance Testing**

*(Testing done to ensure that that the application performs to customer expectations (response time, availability, portability, and scalability)).*

**5.6 Regression Testing**

*(Testing done to ensure that that applied changes to the application have not adversely affected previously tested functionality.)*

**5.7 Acceptance Testing**

*(Testing conducted to determine whether or not a system satisfies the acceptance criteria and to enable the customer to determine whether or not to accept the system. Acceptance testing ensures that customer requirements' objectives are met and that all components are correctly included in a customer package.)*

**5.8 Beta Testing**

*(Testing, done by the customer, using a pre-release version of the product to verify and validate that the system meets business functional requirements. The purpose of beta testing is to detect application faults, failures, and defects.)*

# 6. PASS / FAIL CRITERIA

*(Specify the criteria to be used to determine whether each item has passed or failed testing.)*

**6.1 Suspension Criteria**

Suspension of all testing will commence when the code is not functioning properly. Using version control, the team will revert back to the last working product and resume testing

**6.2 Resumption Criteria**

Suspension will resume after a working version of the code has been restored.

**6.3 Approval Criteria**

Approval of the changes will be done after the code has been checked with the unit testing to make sure that it runs completely

# 7. TESTING PROCESS

**7.1 Test Deliverables**

The provided ANT build commands export an XML file containing the results of the Unit tests. In addition, PMD provides markers within the code that show where areas of concern.

**7.2 Testing Tasks**

Each team member will run tests according to their addins to Eclipse, and then modify the code according the results of the testing/

**7.3 Responsibilities**

Each group member is responsible for their respective testing features, in addition to making sure that the code is pushed to GitHub, in addition to having the code run correctly.

**7.4 Resources**

Resources for testing include, but are not limited to: Eclipse marketplace, developers website, ReadMe guide, and information from tests.

**7.5 Schedule**

# 8. ENVIRONMENTAL REQUIREMENTS

**8.1 Hardware**

Computer 1:

Computer2:

* i7 Processor
* Nvidia GeoForce GTX 1060
* 16 GB RAM

**8.2 Software**

* Eclipse
* GitHub

**8.3 Security**

The testing environment is done completely within Eclipse and the plugins used in Section 8.2. The updated code is then uploaded to GitHub to be merged between the testing team.

**8.4 Tools**

Tools being used in Eclipse are as follows:

* + CodePro
  + PMD
  + FindBugs
  + Eclemma

Each tool was used according to its’ respective user guide and support. All came from the Eclipse marketplace, or were installed through Eclipse.

**8.5 Risks and Assumptions**

*(Identify significant constraints on testing such as test item availability, test resource availability, and time constraints. Identify the risks and assumptions associated with testing tasks including schedule, resources, approach and documentation. Specify a contingency plan for each risk factor.)*

# 9. CHANGE MANAGEMENT PROCEDURES

*(Identify the software test plan change management process. Define the change initiation, change review, and change authorization process.)*