**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.

**AKQ**



**Final Project - FreeCol**

**Software Quality Assurance Plan**

**Version: (1) Date: (04/13/2018)**

**Document History and Distribution**

1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision # | Revision Date | **Description of Change** | **Author** |
| 1 | 04/13/2018 | Initialization | Kolbe |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1. Introduction 1](#_Toc511376909)

[2. Test Items 3](#_Toc511376910)

[3. Features To Be Tested 3](#_Toc511376911)

[4. Features Not To Be Tested 3](#_Toc511376912)

[5. Approach 3](#_Toc511376913)

[6. Pass / Fail Criteria 5](#_Toc511376914)

[7. Testing Process 5](#_Toc511376915)

[8. Environmental Requirements 6](#_Toc511376916)

[9. Change Management Procedures 7](#_Toc511376917)

# Introduction

(Note 1: The Software Test Plan guidelines were derived and developed from IEEE Standard for Software Test Documentation (829-1998)).

*(Note 2: The ordering of Software Test Plan (STP) elements is not meant to imply that the sections or subsections must be developed or presented in that order. The order of presentation is intended for ease of use, not as a guide to preparing the various elements of the Software Test Plan. If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content.)*

*The Introduction section of the Software Test Plan (STP) provides an overview of the project and the product test strategy, a list of testing deliverables, the plan for development and evolution of the STP, reference material, and agency definitions and acronyms used in the STP.*

**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**

*(Describe, at a high level, the scope, approach, resources, and schedule of the testing activities. Provide a concise summary of the test plan objectives, the products to be delivered, major work activities, major work products, major milestones, required resources, and master high-level schedules, budget, and effort requirements.)*

The goal of this project is to improve and ensure the overall quality of the open-source game FreeCol. Our objectives include instituting and maintaining unit tests that provide a minimum of 90% coverage in “critical” classes, providing the same level of coverage in mutation testing, bug detection and correction, and overall improvements to the game’s flow.

Our goal was to meet weekly to discuss progress and divide the current workload, with the goal of tangible advancements made between each meeting. The major steps of the test plan were to gain familiarity with the game, identify the most critical and impactful classes in the program, develop unit tests, develop and adapt to mutation tests, and use knowledge of the program’s functionality to detect and correct bugs within those classes.

**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.** (*This may appear as a specific document (such as a Test Specification), or it may be part of the organization's standard test approach. For each level of testing, there should be a test plan and an appropriate set of deliverables. The test strategy should be clearly defined and the Software Test Plan acts as the high-level test plan. Specific testing activities will have their own test plan. Refer to section 5 of this document for a detailed list of specific test plans.)*

*Specific test plan components include:*

* *Purpose for this level of test,*
* *Items to be tested,*
* *Features to be tested,*
* *Features not to be tested,*
* *Management and technical approach,*
* *Pass / Fail criteria,*
* *Individual roles and responsibilities,*
* *Milestones,*
* *Schedules, and*
* *Risk assumptions and constraints.*

**1.3 Scope**

*(Specify the plans for producing both scheduled and unscheduled updates to the Software Test Plan (change management). Methods for distribution of updates shall be specified along with version control and configuration management requirements must be defined.)*

**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.**

This testing plan is to be discussed at each weekly meeting and updated as necessary, including both the progress made during the previous week and any changes to the plan itself that the team deems necessary. Version control will be maintained through Git, as with the project itself, and will likewise be reflected in the “Revision History” chart in the document itself.

**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.)*

**2.1 Program Modules**

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

For each new module to be constructed, unit tests with a minimum of 90% coverage must be included and passing at 100% before the changes are pushed to the main branch. Within a week of the institution of a new module, the unit tests must be adapted to satisfy mutation testing with a coverage of at least 90% of the mutants “killed”.

**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.)*

This project will not involve testing of graphics or the improvement of any graphical artifacts, as they are outside the scope of this team.

# 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**

(*Specify the criteria used to suspend all or a portion of the testing activity on test items associated with the plan.)*

**6.2 Resumption Criteria**

*(Specify the conditions that need to be met to resume testing activities after suspension. Specify the test items that must be repeated when testing is resumed.)*

**6.3 Approval Criteria**

*(Specify the conditions that need to be met to approve test results. Define the formal testing approval process.)*

# 7. Testing Process

*(Identify the methods and criteria used in performing test activities. Define the specific methods and procedures for each type of test. Define the detailed criteria for evaluating test results.)*

**7.1 Test Deliverables**

*(Identify the deliverable documents from the test process. Test input and output data should be identified as deliverables. Testing report logs, test incident reports, test summary reports, and metrics' reports must be considered testing deliverables.)*

**7.2 Testing Tasks**

*(Identify the set of tasks necessary to prepare for and perform testing activities. Identify all intertask dependencies and any specific skills required.)*

**7.3 Responsibilities**

*(Identify the groups responsible for managing, designing, preparing, executing, witnessing, checking, and resolving test activities. These groups may include the developers, testers, operations staff, technical support staff, data administration staff, and the user staff.)*

**7.4 Resources**

*(Identify the resources allocated for the performance of testing tasks. Identify the organizational elements or individuals responsible for performing testing activities. Assign specific responsibilities. Specify resources by category. If automated tools are to be used in testing, specify the source of the tools, availability, and the usage requirements.)*

**7.5 Schedule**

*(Identify the high level schedule for each testing task. Establish specific milestones for initiating and completing each type of test activity, for the development of a comprehensive plan, for the receipt of each test input, and for the delivery of test output. Estimate the time required to do each test activity.)*

*(When planning and scheduling testing activities, it must be recognized that the testing process is iterative based on the testing task dependencies.)*

# 8. Environmental Requirements

(Specify both the necessary and desired properties of the test environment including the physical characteristics, communications, mode of usage, and testing supplies. Also provide the levels of security required to perform test activities. Identify special test tools needed and other testing needs (space, machine time, and stationary supplies. Identify the source of all needs that is not currently available to the test group.)

**8.1 Hardware**

*(Identify the computer hardware and network requirements needed to complete test activities.)*

The hardware used by our testing team must be capable of running FreeCol at maximum graphical settings, as well as running the Eclipse IDE and all necessary addons as described in 8.4 Tools.

**8.2 Software**

*(Identify the software requirements needed to complete testing activities.)*

The software required for this testing plan includes the Eclipse IDE and various addons (as described in 8.4 Tools), as well as the FreeCol program itself.

**8.3 Security**

*(Identify the testing environment security and asset protection requirements.)*

*//I think we can safely leave this section blank; we aren’t concerned with keeping our work secured.*

**8.4 Tools**

*(Identify the special software tools, techniques, and methodologies employed in the testing efforts. The purpose and use of each tool shall be described. Plans for the acquisition, training, support, and qualification for each tool or technique.)*

The steps described in this test plan will be carried out in the Eclipse IDE, with version control implemented through Git (via the eGit Eclipse addon). Mutation tests are to be developed using PIT Mutation testing. Cyclomatic complexity is to be determined with Google CodePro through Eclipse. Code coverage is to be calculated with the EclEmma Eclipse addon.

Each of the above tools has been used in previous classwork and projects, and no further training has been deemed necessary.

**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.)*

Time and manpower are the greatest constraints on this test plan. This was factored in to the ordering of our objectives and timeline, and an attempt was made to keep these steps in a logical order that would allow for maximum completion even if not all steps were completed.

# 9. Change Management Procedures

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