Test Plan Template:

Prepared by:

Sinéad Howard

27-04-2020

TABLE OF CONTENTS

1. INTRODUCTION

2.0 OBJECTIVES AND TASKS

2.1 Objectives

2.2 Tasks

3.0 SCOPE

4.0 Testing Strategy

4.1 Unit Testing

4.2 System and Integration Testing

4.3 Performance and Stress Testing

4.4 User Acceptance Testing

4.5 Batch Testing

4.6 Automated Regression Testing

4.7 Beta Testing

5.0 Test Schedule

6.0 Control Procedures

7.0 Features to Be Tested

8.0 Features Not to Be Tested

9.0 Resources/Roles & Responsibilities

10.0 Schedules

11.0 Risks/Assumptions

12.0 Tools

1.0 INTRODUCTION

A brief summary of the product being tested. Outline all the functions at a high level.

2.0 OBJECTIVES AND TASKS

2.1 Objectives

Describe the objectives supported by the Master Test Plan, eg., defining tasks and responsibilities,

vehicle for communication, document to be used as a service level agreement, etc.

2.2 Tasks

List all tasks identified by this Test Plan, i.e., testing, post-testing, problem reporting, etc.

3.0 SCOPE

General

This section describes what is being tested, such as all the functions of a specific product, its existing

interfaces, integration of all functions.

Tactics

List here how you will accomplish the items that you have listed in the "Scope" section. For

example, if you have mentioned that you will be testing the existing interfaces, what would be the

procedures you would follow to notify the key people to represent their respective areas, as well as

allotting time in their schedule for assisting you in accomplishing your activity?

4.0 TESTING STRATEGY

Describe the overall approach to testing. For each major group of features or feature combinations,

specify the approach which will ensure that these feature groups are adequately tested. Specify the

major activities, techniques, and tools which are used to test the designated groups of features.

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 one.

4.1 Unit Testing

Definition:

A unit test tests one small unit at a time. A unit is defined as the smallest testable part of the code and varies depending on language. Unit tests should be designed to determine that a particular piece of code does exactly what it is meant to do when it is meant to do it. It is done using the white box testing method i.ie the item is known to the tester as they choose the inputs and know the expected outputs. Unit testing is the first test to be carried out.

There are many benefits to unit testing including:

* Makes it easier to detect flaws when changing or rewriting code
* Makes code easier to reuse
* Writing the tests in advance makes the testing process faster in the long run
* Catching a defect at a lower level cost less to repair than if it is caught at a later stage of testing

Specify the minimum degree of comprehensiveness desired. Identify the techniques which will be

used to judge the comprehensiveness of the testing effort (for example, determining which

statements have been executed at least once). Specify any additional completion criteria (for

example, error frequency). The techniques to be used to trace requirements should be specified.

Participants:

Unit testing is usually carried out by the developers or sometimes by an independent external developer . this is because they wrote the code and know better than others what the output of the code is meant to be.

Methodology:

Describe how unit testing will be conducted, including a description of tests to be carried out. Who

will write the test scripts for the unit testing, what would be the sequence of events of Unit Testing

and how will the testing activity take place?

4.2 System and Integration Testing

Definition:

Tests the interactions between separates units when they are integrated together. Tests should be written that detects defects that occur when separate components interact with each other and therefore is usually conducted on a complete system. It is also done after the individual components have been tested themselves. Each integration is tested as soon as it is added to the system until the system is complete.

Some of the benefits of system and integration testing include:

* Helps to detect bugs early
* The integration testing can be carried out during development
* Finds errors in the interface

Participants:

Integration testing to be carried out by the software tester

be responsible for this activity.

Methodology:

Describe how System & Integration testing will be conducted, including a description of tests to be

carried out Who will write the test scripts for the unit testing, what would be sequence of events of

System & Integration Testing, and how will the testing activity take place?

4.3 Performance and Stress Testing

Definition:

Performance testing sets the standards for the game. It is used to determine how the game will react under normal circumstances.

Stress testing determines how stable the system is under pressure. It tests how the system reacts to extreme loads and how it recovers from it. Stress testing is done to ensure that the game does not crash under extreme circumstances

Some of the benefits of stress testing include:

* Helps check the games performance under various circumstances
* Allows testers to monitor the game at time of failure
* Can check for data security issues and privacy issues during stress test

Participants:

Who will be conducting Stress Testing on your project? List the individuals that will be responsible

for this activity.

Methodology:

Describe how Performance & Stress testing will be conducted, including a description of tests to be

carried out Who will write the test scripts for the testing, what would be sequence of events of

Performance & Stress Testing, and how will the testing activity take place?

4.4 User Acceptance Testing

Definition:

While all the other tests have been created under the developer guidelines, User acceptance testing is used to evaluate the game in accordance the user’s guidelines. It is carried out in respect to the users needs. It is the last level of testing before a system is released publicly. Also known as alpha and beta testing

Some of the benefits of user acceptance testing include:

* Reduces the risks of bugs making it through to production
* Improves user experience
* Removes the chance of testing bias

Participants:

Who will be responsible for User Acceptance Testing? List the individuals' names and responsibility.

Methodology:

Describe how the User Acceptance testing will be conducted, including a description of tests to be

carried out Who will write the test scripts for the testing, what would be sequence of events of User

Acceptance Testing, and how will the testing activity take place?

4.5 Batch Testing

//Does not apply to video game as it requires a manual user input for testing

4.6 Automated Regression Testing

Definition:

Regression testing is the selective retesting of a system or component to verify that modifications

have not caused unintended effects and that the system or component still works as specified in the

requirements.

Participants:

Methodology:

4.7 Beta Testing Participants:

Methodology:

5.0 TEST SCHEDULE

Include test milestones identified in the Software Project Schedule as well as all item transmittal

events.

Define any additional test milestones needed. Estimate the time required to do each testing task.

Specify the schedule for each testing task and test milestone. For each testing resource (that is,

facilities, tools, and staff), specify its periods of use.

6.0 CONTROL PROCEDURES

Problem Reporting

Document the procedures to follow when an incident is encountered during the testing process. If a

standard form is going to be used, attach a blank copy as an "Appendix" to the Test Plan. In the

event you are using an automated incident logging system, write those procedures in this section.

Change Requests

Document the process of modifications to the software. Identify who will sign off on the changes

and what would be the criteria for including the changes to the current product. If the changes will

affect existing programs, these modules need to be identified.

7.0 FEATURES TO BE TESTED

Identify all software features and combinations of software features that will be tested.

8.0 FEATURES NOT TO BE TESTED

Identify all features and significant combinations of features which will not be tested and the

reasons.

9.0 RESOURCES/ROLES & RESPONSIBILITIES

Specify the staff members who are involved in the test project and what their roles are going to be

(for example, Mary Brown (User) compile Test Cases for Acceptance Testing). Identify groups

responsible for managing, designing, preparing, executing, and resolving the test activities as well as

related issues. Also identify groups responsible for providing the test environment. These groups

may include developers, testers, operations staff, testing services, etc.

10.0 SCHEDULES

Identify the deliverable documents. You can list the following documents:

- Test Plan

- Test Cases

- Test Incident Reports

- Test Summary Reports

11.0 RISKS/ASSUMPTIONS

Identify the high-risk assumptions of the test plan. Specify contingency plans for each (for example,

delay in delivery of test items might require increased night shift scheduling to meet the delivery

date).

12.0 TOOLS

List the Automation tools you are going to use. List also the Bug tracking tool here.