Test Plan

<Project Name>

<Authors>

<Date>

<Note: everything surrounded by <> is either a placeholder to be filled in or a comment to be deleted. All tables contain fictional example entries which should be removed.>

<This plan follows the style of *IEEE Standard for Software Test Documentation*, IEEE Std 829-1983.>

Table of Contents

1. Introduction 3
1.1. Objectives 3
1.2. Tasks 3
2. Test Items 3
3. Features to be Tested 3
4. Features not to be Tested 3
5. Approach 4
5.1. Code Inspection 4
5.2. Unit Testing 4
5.3. Functional Testing 4
6. Item Pass/Fail Criteria 4
6.1. JUnit tests 4
6.2. Functional Tests 4
7. Suspension Criteria and Resumption Requirements 5
8. Test Deliverables 5
9. Testing Tasks 5
10. Environmental Needs 5

11. Responsibilities 5
12. Staffing and Training Needs 5
13. Schedule 5
14. Risks and Contingencies 6
15. Approvals 6
16. Do List 6
17. Revision History 6

1. Introduction

<Summarize the software items and software features to be tested.>

For instance:

The project that this test plan is written for is the Java Word Processor (JWP) that was provided to…….

The JWP is broken down into six java packages……..

1.1. Objectives

This plan intends to describe how the creation of tests cases will be accomplished …….

Perform tests on the JWP application to find as many bugs and errors as possible with as much code coverage as possible……

1.2. Tasks

- Unit Tests using JUnit
- Functional Testing
- Code Coverage using the EMMA

2. Test Items

<Identify the test items including their version/revision level.>

For instance:

The following modules in the JWP application were tested……

Our testing target will be 90% method coverage of all primary functional areas of the application. This includes all code that will be executed as part of normal operation of the program. Areas which will be not be tested specifically include

- Unused Classes
- Unused Methods
- Low-risk Trivial methods (get/set methods)
- Supporting Library Methods (assumed to be externally tested)

3. Features to be Tested

<Identify all software features to be tested. This may be a list of numbered requirements.>

4. Features not to be Tested

<Identify all features that will not be tested.>

For instance:

A number of features were not included for testing in this test plan. Features that were not tested include get and set methods, any methods that did not contain any executable code, and any methods that were not called anywhere in the execution of the application. In addition, any method whose functionality was tested by executing a test on another method was not tested. For example, if method X calls method Y, and method X is confirmed to be working properly, then it can be concluded that method Y is working as well.

5. Approach

<Describe the overall approach.>

For instance:

Two approaches, code inspections and unit/functional testing, will be used to find defects in the JWP project. Code inspections will be used to find an initial set of defects, and……

5.1. Code Inspection

<Summarize your code inspection process for part 2>

5.2. Unit Testing

<Summarize your unit testing process>

5.3. Functional Testing

<Summarize your functional testing process>

6. Item Pass/Fail Criteria

<Specify the criteria to be used to determine whether each test item has passed or failed testing.>

6.1. JUnit tests

For instance:

All JUnit tests should pass/fail depending on each test case coded……..

6.2. Functional Tests

For instance:

Each functional test has its own pass fail criteria based on the test that is being preformed.

Also if the program displays unexpected behaviors during the test then this is a failure case and needs to be reported……

7. Suspension Criteria and Resumption Requirements

<Specify the criteria used to suspend testing, such as abnormal termination. Specify what conditions must be met before testing may be resumed.>

For instance:

If the program crashes or stops unexpectedly; find out what caused that error and report on it and then continue with the test cases

8. Test Deliverables

<Identify the deliverable documents, such as this test plan, test specifications, and test results.>

9. Testing Tasks

<Identify the set of tasks necessary to prepare for and perform testing.>

10. Environmental Needs

<Specify the necessary properties of the test environment, such as hardware, software, and network connectivity.>

11. Responsibilities

<Identify the groups responsible for planning, executing, and reporting the results of tests.>

12. Staffing and Training Needs

<Specify milestones for each type of testing, e.g.., completion of unit testing, completion of system testing.>

13. Schedule

<Specify milestones for each type of testing, e.g.., completion of unit testing, completion of system testing.>

14. Risks and Contingencies

<Identify the high-risk assumptions of the test plan. Specify mitigation strategies and contingency plans.>

15. Approvals

<Specify the names of all persons who must approve this plan.>

16. Do List

<List of items to be completed in THIS artifact.>

#	Who	Due	What
1	Groucho	4/7/2003	Add code inspection plans
2	Zeppo	4/1/2003	Add acceptance test plans

17. Revision History

Date	Who	Revision
2/2/2003	Groucho	Added regression test plans