

Test Plan

Version X.Y – yyyy.mm.dd

Created yyyy.mm.dd

Project Name

Name 1 (role-if any)

Name 2(role-if any)

...

Name N(role-if any)

Table of Contents

1 INTRODUCTION.....	3
1.1 DOCUMENT SCOPE.....	3
1.2 INTENDED AUDIENCE.....	3
1.3 PROJECT IDENTIFICATION.....	3
2 SOFTWARE QUALITY GOALS.....	3
2.1 OVERVIEW.....	3
2.2 METRICS.....	3
2.3 TOOLS.....	3
3 TESTING PROCESSES.....	3
3.1 OVERVIEW.....	3
3.2 VERSION CONTROL.....	3
3.3 CODE REVIEWS.....	4
3.4 OTHER TEST PROCESSES.....	4
4 TEST CASES.....	4
4.1 UNIT TEST CASES.....	4
4.2 INTEGRATION TEST CASES.....	5
4.3 SYSTEM OR USER INTERFACE TEST CASES.....	5
4.4 TRACEABILITY MATRIX (BONUS 5 POINTS).....	5
5 APPENDIX.....	6

1 Introduction

1.1 Document Scope

Here you should describe what the purpose of this document is, and what it is not (e.g. it does not describe how a user uses the system.) No-one likes to have to read a few pages to figure out if they're looking at the right document.

1.2 Intended audience

This section should describe the stakeholders you are targeting this document at. This is another way to let people quickly discover whether they're reading the wrong document.

1.3 Project Identification

Identify the project that this plan applies to (codename if any, links to other documents).

2 Software Quality Goals

2.1 Overview

Explain which quality requirements are addressed by this testing plan, and which are not.

2.2 Metrics

Identify metrics you will use to assess the quality of your software. Describe what is the benefit of using the metric, and how it is to be used. What is the target numerical value to be achieved to meet the relevant requirement? Discuss assumptions made by the metric and its appropriateness for your project. Examples: LOC (lines of code), cyclomatic complexity, code coverage.

2.3 Tools

Find and describe tools relevant to your project that can gather data, calculate the metric, and analyze the results.

3 Testing Processes

3.1 Overview

Will you be using a testing framework? When are you going to use which process?

3.2 Version Control

Here you will explain how you will use version control in relation to testing. For example, if in your implementation plan you mentioned that some scripts would run when code is committed, this is a good place to define what these scripts will do. You may refer to your implementation plan instead of repeating information. Making sure that your code parses correctly in the particular language you're using is a good idea here.

3.3 Code Reviews

Describe how and when you will perform code reviews (a.k.a. inspections). If you already addressed this in your implementation plan, you may refer to it and add here any details you left out earlier. Example:

Completed code and document inspection sheets will be checked into their respective folders in the Documentation folder in CVS. Unit tests must all pass before doing code reviews, to save the time of the reviewers.

- Author
 - o Will print a copy of the code with line numbers continuous on all pages.
 - o Will print unit test results
 - o Deliver the code in advance for the reviewer to examine
- Inspector
 - o Will meet with the author, discuss issues found by line number and provide an annotated copy
 - o Will make sure that the code follows coding standards.

3.4 Other Test Processes

For example, how unit tests will be conducted; when will the metrics mentioned earlier be calculated.

4 Test Cases

This is where you should spend most of the effort for this deliverable. You should have a sub-heading for each type of testing you intend to do. Advanced testing, which we are not expecting you to do in this class, would include the identification of data input equivalence classes, boundary value analysis, and error classes.

4.1 Unit Test Cases

Show the test cases for unit testing. Refer to p. 398 of the book for information to include in a formal test case. Example:

Test Case 9865

Module or Class: IntegerList

1) Testing: debugPrint() method

Method:

- i. Create an IntegerList
- ii. Enter the integers 5, 7, and 9
- iii. Call debugPrint()

Expected Result: The numbers 5, 7, and 9 are printed to stderr in any order

Cleanup: None

2) Testing: addInteger(int) method

...

4.2 Integration Test Cases

Integration test cases can be more difficult to conduct than unit tests, depending on how well the interfaces between components were defined. You may need to get a component in a given state before a test can be conducted. Then, you may need to unwind the test to a default state before the next test can be performed; this can happen when testing a complex system backed with a database. Refer to pp. 416-420 of the book for examples. Alternate format:

Test Case ID

- Summary
- Configuration of components
- Initial Conditions
- Steps to run the test case
- Expected behavior/outcome
- Recovery steps

4.3 System or User Interface Test Cases

Show test cases that will exercise the entire system. Example:

Test cases for form “Blahblah”.

Preconditions: User has logged in and selected the “create” functionality.

#	Description	Pass	Fail
1	Click the button named “a button” without entering any data. Result: Error message appears, with button to return to form.		
2	Enter a graduation date earlier than the birthday, and click the submit button Result: Error message appears		
3	Set the inputs to X, Y, Z, and click the submit button. Result: The list of items is displayed; a new item is shown, corresponding to the data that was entered.		

4.4 Traceability Matrix (bonus 5 points)

The goal of the traceability matrix is to show how the tests relate to the other documents you created. You can use it, for example, to see how well every requirement, use case or class has been tested. The book doesn't seem to contain any example of a traceability matrix, but it's essentially a spreadsheet where each row is a test, and each column is what you want to relate the tests to (using a short ID). The most common use is a

requirements traceability matrix. If you produce one here you may earn 5 bonus points (total score still limited to a maximum of 100%). If you didn't assign unique identifiers to your requirements, you will need to do so here.

5 Appendix