TEST PLAN (TP)
FOR

Phunctional UML Editor
(pUML)

Version 2.0
May 8, 2012

Prepared for:
Dr. Clint Jeffery

Prepared by:
Josh Armstrong
Zach Curtis
Brian Bowles
Logan Evans
Jeremy Klas
Nathan Krussel
Maxine Major
Morgan Weir
David Wells

University of Idaho
Moscow, ID 83844-1010

CS384 TPD
RECORD OF CHANGES (Change History)

Change Number	**Date completed**	**Location of change (e.g., page or figure #)**	**A M D**	**Brief description of change**	**Approved by (initials)**	**Date approved**
01	03/27/2012	Test Plan	A	Document Created	MM	03/27/2012
02	04/09/2012	Test Plan	A	Test cases added	MM	04/09/2012
03	05/08/2012	Test Plan	M	Final Revision	MM	05/08/2012

A - ADDED M - MODIFIED D – DELETED

pUML
TABLE OF CONTENTS

1 TEST PLAN IDENTIFIER

This document is a stand-alone document and has no identification numbers other than the revision number.

2 REFERENCES

- Systems and Software Requirements Specification (SSRS) ver. 1.0
- System and Software Design Description (SSDD) ver. 1.0

3 INTRODUCTION

The purpose of this Test Plan is to ensure the integrity of the pUML software through a well-defined series of tests. The testing outlined in this plan will be applied to each component of the pUML software. Errors will be well-documented and each test will require a follow-up so all changes or recommended improvements to both functionality and features may be applied to this software.

The testing required will include manual testing, unit testing, and a combination of both and/or other specialized tests as necessary for each test item. results for each test item will be logged. action taken on each test item will be logged The purpose for this test is to ensure that all errors are found and handled.

4 TEST ITEMS

Items to be tested include:

- Installers
- Multiplatform portability (Windows and Linux)
- Functions and parameters
- Excessive code complexity
- Large program components, i.e. Window and Canvas class

5 SOFTWARE RISK ISSUES

Software risk areas include extremely complex functions and modifications made on components with a history of failure. These functions and components are itemized as follows:

- QPaint. This introduces a complex hierarchy of classes and functions. To ensure this is adequately functional, manual testing will be required.
- Compilation issues.
- Restoring object classes. This tends to be fragile.

6 FEATURES TO BE TESTED

Features to be tested include all objects, connectors, and associated functionality, Open/Save/Restore functionality, and ensuring that all diagram types load properly.

6.1 Black Box Testing

All black box tests will be conducted by students in the Computer Science department, and will be performed from a user perspective on the GUI functionality only.

Name of Test	**Tests**	**Fulfils SSRS Req.**
testinstall	pUML installer successfully installs and uninstalls	2.2.1
testlaunch	pUML launches correctly	2.2.1
testmainwindow	All main window options	2.2.2
testobjects	All object behavior	2.2.3
testconnectors testselfconnector	All connector behavior	2.2.4
testsave testsaveas	Launch function All open functions All save functions Diagram integrity Tab functionality	2.2.1 2.2.5 2.2.6 2.2.7 2.2.8

6.2 Integration Testing

The integration test is being indirectly tested as a combination of the user-tested black box tests on the GUI, and the unit tests performed on the pUML software.

6.3 Unit Testing

The unit tests will be conducted as part of a QT unit test project named **pUMLUnitTest.**

7 FEATURES NOT TO BE TESTED

- We will not test invalid Save file names. QT already has implemented measures to prevent invalid file names.
- We will not test what the right-click function does in areas we have not assigned right-click features.

8 APPROACH

We will be utilizing QT's built in unit testing features by designing a project to test all units. Black box testing will be conducted by Computer Science students per text files outlining all steps to complete the test. Failed tests will be logged as issues in Google code and will be resolved manually.

9 ITEM PASS/FAIL CRITERIA

If any test fails in any aspect, then entire test will have been assumed to have failed. A test will not be considered to pass until all steps of that test have passed successfully. If a test continues to fail and seems unresolvable given time and resources, the test may be rewritten to ensure a satisfactory pass grade may occur.

10 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

Features and desired functionality may be suspended if it is assumed that it cannot be reasonably functional within the remaining time left to complete this project. Features to be suspended at this time will not be resumed.

11 TEST DELIVERABLES

- Test plan document
- All black box test cases with extraneous observations
- All unit test runs
- Relevant error logs and problem reports

12 REMAINING TEST TASKS

There are no remaining test tasks at this time, since further testing will not continue after the end of Spring semester 2012.

13 ENVIRONMENTAL NEEDS

There are no environmental needs associated with the pUML project.

14 STAFFING AND TRAINING NEEDS

pUML users are assumed to be familiar with UML and have a need to perform UML diagramming. As such, basic training will not be necessary. A user guide will be provided with the pUML software to address any additional concerns.

15 RESPONSIBILITIES

The test plan documentation is being handled by Maxine Major. The unit test is being developed primarily by Josh Armstrong. The other members of the group have contributed to the development of the content to be tested and have decided which features may forgo testing. All group members have contributed to the Alpha phase of black box testing, to ensure that major bugs are resolved prior to giving tests to third party testers.

16 SCHEDULE

The black box tests and the unit test have been modified several times during the last month of the semester, and testing will occur until the project completion deadline. All testing will be considered complete at that time.

17 APPROVALS

Dr. Clint Jeffery is the only authorized individual to approve this project as complete.