TEST PLAN (TP)
FOR

Phunctional UML Editor
(pUML)

Version 2.0
April 9, 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	3/27/2012	Test Plan	A	Document Created	MM	3/27/2012
02	4/9/2012	Test Plan	A	Test cases added	MM	4/9/2012

A - ADDED M - MODIFIED D – DELETED

pUML
TABLE OF CONTENTS

Section Page

1 TEST PLAN IDENTIFIER

This is test plan No. 007.

The pUML project team is not associated with an established company at this time, and this will be the only unique numerical identification number for any of our products.

2 REFERENCES

- Systems and Software Requirements Specification (SSRS) ver. 0.0
- System and Software Design Description (SSDD) ver. 0.0

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
- Legality
- 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.

Name of Test	Tests	Fulfils SSRS Req.
testinstall	pUML installer successfully installs and uninstalls	
testlaunch	pUML launches correctly	
testsave	All save functions	
testopen	All open functions	
testmainwindow	All main window options	
testobjects	All object behavior	
testconnectors	All connector behavior	

6.2 Integration Testing

The integration test will be conducted as a unit test, testing all signals and slots and connections within pUML.

Name of Test	Tests	Fulfils SSRS Req.
connectCanvasWithDocumentTest	connectCanvasWithDocumentTest	

6.3 Unit Testing

The unit tests will be conducted as part of a QT unit test project. This project will test all units.

Name of Test	Tests	Fulfils SSRS Req.
pUMLUnitTest	all pUML functions	

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
- The integration test runs
- 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 occur at the end of this semester.

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

Who is in charge? This issue includes all areas of the plan. Here are some examples:

- *Selecting features to be tested and not tested.*
- *Ensuring all required elements are in place for testing.*

Specific testing responsibilities are still TBD.

16 SCHEDULE

All black box tests, unit tests, and the integration test will be conducted on a weekly basis, as soon as test creation has been completed.

17 PLANNING RISKS AND CONTINGENCIES

[Insert text here.]

18 APPROVALS

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