Plotter - Software Test Plan

# Team 4

# SER 216

# 1. Introduction

Team 4 will be conducting a comprehensive review of the Plotter Software (Math graphics 2.0.1). The goal of the review is to identify any remaining bugs in the software and fix them. While identifying bugs in the software, we will also be making recommendations for improvements and making changes that retain the original functionality but improve the interface and usability of the software.

**Test Plan Objectives**

* Identify remaining bugs in the software.
* Report bugs found.
* Make changes to software to fix bugs.
* Identify potential improvements in the software.
* Make changes to the software that improves usability.
* Report all changes made.

# 2. Scope

1. **Graphical Interface**

The graphical interface will be tested for performance and functionality. The interface will be reviewed and areas for improvement will be identified. Improvements will be made to the GUI that enhances the user experience while still maintaining the functionality of the software.

1. **Renderer**

The code dedicated to rendering the graphs will be reviewed for bugs and potential areas of improvement. The three different rendering methods, Cartesian 2D, Polar 2D and Cartesian 3D will all be tested for bugs and performance capabilities.

1. **Math Calculator**

The math code will be checked for bugs and potential areas of improvement. The Integral button functionality and the Show DF button will be tested. The displayed function field will be tested under varying data sets.

# 3. Test Strategy

The test strategy will be implemented in seven tests. These tests will examine the software limitations and capabilities. The testing will identify bugs and areas where improvements can be made. The final stage of the testing process will be to present the changes made to the clients and compare the finished software with the original requirements document.

1. **System Test**

System testing will focus on the behavior of the software when used under normal usage conditions. Tests will consist of known errors to check error handling as well as user scenarios designed to examine how the different components work together.

1. **Performance Test**

Performance testing will measure the system’s response time under varying conditions. Apache JMeter will be used to test that all aspects of the software are functioning within the parameters defined by the client.

1. **Automated Test**

Unit tests will be written using TestNG software. These tests will be used through the testing process to verify when changes are made to the software that previously tested code still functions normally.

1. **Stress and Volume Test**

The math code will be tested with numbers that push the boundaries of the software. All math rules will be tested to show correct error messages are thrown when math rules are broken.

1. **Recovery Test**

This testing will break the software in various ways so recovery conditions can be tested and evaluated.

1. **Beta Test**

The software will be made available to real life users to test in daily use scenarios. This will help identify issue that may not have been identified by traditional testing methods.

1. **User Acceptance Test**

Once all testing has been completed and all of the identified bugs have been fixed testing will take place with the clients. The original requirements document will be used to confirm the software meets the client’s expectations.

# 4. Environment Requirements

## Application Requirements

The Plotter application requires that the machine it is running on has Java version 1.7 or later installed. No other environmental specifications are necessary for the program to run.

## Test Environment Requirements

The Java testing framework TestNG has been selected as the testing framework that will be used for unit, functional, integration, and end-to-end tests. TestNG can be set up in an Eclipse IDE and easily facilitated from there. For performance and load testing, Apache JMeter has been selected as the preferred testing framework. JMeter comes with its own user interface where test set-up and executions can be facilitated. In order for both TestNG and JMeter tests to run successfully, the machine where the tests are hosted must have Java version 1.7 or later installed.

# 5. Test Schedule

* Get familiar with Plotter application 04/03/2017 – 04/09/2017
* Test framework analysis/selection 04/10/2017 – 04/16/2017
* Software plan and test case development 04/17/2017 – 04/23/2017
* Test implementation 04/23/2017 – 04/28/2017
* Test execution and analysis 04/28/2017 – 05/02/2017

# 6. Control Procedures

# Reviews

After each phase of the test schedule (outlined in section 5), the team will review the product of that phase together to ensure all team members are on the same page, and to identify any gaps that may exist. Furthermore, the review is also a way for the team to brainstorm on any possible additions or improvements we can make in that phase of the test schedule. Reviews are scheduled to occur on a set day and time of the week (Sunday at 12 pm). Impromptu review meetings can also be scheduled in emergency cases.

## Bug Review meetings

Bug review meetings will be held after a set of tests is run. The goal of these meetings is to review with the quality team the different defects that are discovered with each type of test, the priority and severity of the bug, and what action steps to take next. High priority bugs will be escalated immediately while low and medium priority bugs will be written up and reported out. This meeting will also help prevent duplicate defects from being written up.

## Change Request

When tests are in the process of being run, changes should not be made to the application in the testing environment apart from high priority defects. High priority defects that come out of testing (that are identified as high priority in the bug review meeting) will be escalated to a management board. There, it will be determined whether the bug in question is high enough in severity where a fix needs to be pushed out to the testing environment (which would interrupt tests and possibly cause testing to have to start over).

## Defect Reporting

The quality team will write up defects as they are discovered, and they will be evaluated in bug review meetings. A defect should have the following information (this will benefit both developers in fixing the bug and quality in verifying the bug):

* Descriptive title
* Version where bug occurs
* Priority and severity (as discussed in the bug review meeting)
* Steps to reproduce (should come from the test case)
* Any supporting files that could be useful to the developer such as screenshots or an exported application file

The defect will also contain a status so that it can be tracked where the bug is in the validation process (not started, in progress, verify, and closed). When the developer begins work on the fix, it is their responsibility to update the status of the bug to in progress, and similarly put it in verify when it is ready to be re-tested. If applicable, the developer should also update the bug with the “fixed version” so the tester can be sure they are re-testing in the correct version. Once the tester has verified the fix, the bug will be moved to a closed status.

# 7. Functionality To Be Tested

The following is a list of functions that will be tested:

* Plotting a linear function
* Plotting a quadratic function
* Plotting a polynomial function
* Draw button (updates the graph)
* Zooming in
* Zooming out
* Integral Panel
  + Changing x1 and x2 values
  + Trapezium
  + Simpson
  + Gauss
* Showing/hiding degree of freedom (correct for different types of functions)
* Position indicator (bottom right corner)
* Moving around the graph
  + Click and drag
  + Using arrows
* Visualization
  + Cartesian 2D
  + Polar 2D
  + Cartesian 3D
* Changing colors (test different components change to correct colors)
* Saving and/or exporting a graph

# 8. Resources and Responsibilities

## Resources

* Connor Premuda is the Project Manager
* James Austin is the Test Lead
* Connor Premuda, James Austin, Enya Yan and Nergal Givarkes are the testers

**Responsibilities**

Our team consists of the following four members:

* **Connor Premuda** - Project Manager of this project. His responsibility is gathering the final results from the GitHub reviewing and then submitting them on BlackBoard. His primary focus is the overall success of the team in the project.
* **James Austin** - Test Lead in this team. His responsibility is scheduling weekly online meetings with them team members and keep in contact with in case immediate meeting is required. He also makes sure that the test results are acceptable based on the requirements.
* **Enya Yan** - Quality Engineer for the team. His responsibilities are to test the Plotter application based on performance, quality and efficiency. Project Manager and Test Lead also contribute in testing the application to speed up the process and to compare the final results. Additionally, Enya helps monitor testing and performance to make sure the application is running as expected.
* **Nergal Givarkes -** Also a Quality Engineer for the team. His responsibilities, similarly, are to test the Plotter application based on performance, quality and efficiency. Nergal also helps with the analysis of testing tools and frameworks necessary to cover the testing aspects needed for this application.

# 9. Deliverables

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **Responsibility** | **Completion Date** |
| Analyze the design and source code and create diagrams | Designing the State, Activity, use case and class diagrams which all team members worked on. | 04/09/2017 |
| Testing Tools Report | Selecting testing tools, deliver a report on which tools and frameworks will be used and why | 04/15/2017 |
| Software Plan Test | Gathering some information about the whole plan | 04/23/2017 |
| Document and communicate test status/coverage | Meeting weekly to discuss the plan and requirements | Weekly |
| Test Plan results | Testing the application | 04/30/2017 |
| Summary Report | Reporting the final result obtained from the test | 05/02/2017 |
| Individual Contribution | Reporting the effect of every member in the team | 05/02/2017 |

# 10. Suspension / Exit Criteria

If any defects are found which seriously impact the test progress, the QA manager may choose to suspend testing. Criteria that will justify test suspension are:

* In case team members have difficulty with how to operate testing tools.
* In case testing tools are not used correctly and cause some defects in the application.
* In case we face with activation of the testing tools.

# 

# 11. Resumption Criteria

If testing is suspended, resumption will only occur when the problem(s) that caused the suspension has been resolved. A branch will be made from the Github repository for the “hot-fix”. All team members will agree to the change and then merge the branch onto the trunk.

# 12. Dependencies

## Personnel Dependencies

Arizona State University Software Engineering students that have the necessary pre-requisites for SER 216

## Software Dependencies

The source code must be available to the testers as back up. (Black Board and a private Github repository). We also depend on the Eclipse IDE (which contains the testing framework plug-ins) for running some of our tests. Java version 1.7 or later is required for this.

## Hardware Dependencies

Mac, PC or Linux that is capable of running Eclipse (or the preferred IDE) and accessing the Internet.

## Test Data & Database

GitHub repository and Black Board site with necessary resources.

# 13. Risks

## Schedule

The four of us are busy college students with job commitments and other timely matters. Missing the schedule would essentially terminate our entire work since there would be no more opportunities present.

## Technical

Our private Github will handle the versions of the software. We will always have a “safe” previous version whenever making changes.

## Management

It is up to the individual members to find time in their schedule for their respective responsibilities.

## Personnel

As a team, we will help each other when members are stuck on certain areas. If one member fails to complete his part, the remaining members can fill his role.

## Requirements

The test plan and test schedule are based on the current Requirements Document. All team members will approve any changes.

# 14. Tools

The tester will use the tool of his comfort. The majority will be using Eclipse and it’s features such as JUnit, EclEmma, etc. Additionally, we will be taking advantage of the TestNG unit and functional testing framework, and the Apache JMeter performance testing frameworks. TestNG can be used via Eclipse, however, JMeter tests will be facilitate through it’s own user interface.