Quality Assurance Plan

Group A - Team Red Panda



Peter Grypma

Chris Hobbs

Hizkia Mertadiwangsa

David Wu

Table of Contents

[Quality Assurance Plan 1](#_Toc339031895)

[Software Tools for Unit Testing 2](#_Toc339031896)

[Internal deadlines 2](#_Toc339031897)

[Acceptance Testing of Version 3 3](#_Toc339031898)

[Integration Testing 3](#_Toc339031899)

[Code Line and File records and other testing and validation methods 4](#_Toc339031900)

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Status** | **Revision Date** | **By** |
| **1.0** | **Created** | **Oct. 17** | **Peter Grypma** |
| **2.0** | **Edited, revised** | **Oct. 19** | **Peter Grypma** |
| **3.0** | **Updated version, edited acceptance testing** | **Oct. 26** | **Peter Grypma** |
| **4.0** | **Updating to be to date with version 1 of project** | **Oct. 31** | **Peter Grypma** |
| **5.0** | **Made it to be QA Plan for Version 1, for homework 3 submission** | **Nov. 2, 2012** | **Peter Grypma** |

# **Software Tools for Unit Testing**

**The software we are using for unit testing will be the xcode integrated development environment on the Mountain Lion operating system for Macintosh computers. As well, with the xcode environment there is an iPhone simulator which will help us to see the results of what we are doing and the iPhone simulator will help with debugging. We generate our test cases by hitting play on xcode and the iPhone simulator will display the results. Then, from the iPhone simulator, we can go through test cases to make sure the program is doing what we thought it would do.**

## **Internal deadlines**

-Deadline for unit testing of the first prototype is by the latest October 31, as the first prototype is due November 2.

-Deadline for doing proof of concept for both apps would be on October 17, which is the using a dummy, simple wordlist. After having the games working, we would implement the games using a wordlist from a location on a server.

-Deadline for the second prototype, which is being able to login and pull a wordlist from a specified source, will be on November 14, so that we have time before the deadline of November 16.

-We would use the same user interface, but update the settings so that a login feature is implemented so that it can pull a user specific wordlist. Testing would be to test whether we can pull a wordlist to the device, and then test to integrate that wordlist into our existing program.

-Deadline for all features implemented will be November 29, giving us enough time to test everything in the final product for the final due date of December 3.

For all of these deadlines, we will be testing against our requirements to see if they are met. This is the verification part of testing, and will be done before submitting each version of our product.

## **Acceptance Testing of Version 1**

**The deadline for our version 1 is October 31 at midnight, so that we can do user testing, which will take place in the Computer Lab. To do this, we will go through all possible things that you can do in our application that we have implemented for version 1, outlined in the following:**

* When game is finished with all correct answers, then games should work correctly by going to end game screen and checking the words you answered correctly
* Make sure typing wrong answers does not yield a check mark beside a word, and the game will only finish to end of game screen when all answers are correct
* When pressing “menu” button in the game, make sure the “return” button brings you back to the game with previous game data saved
* The words in *Word Jumbles* are scrambled, the words in *Missing Letters* have missing letters
* “Main Menu” button brings the user back to the main menu
* The text of the games is clear and the keyboard is not overlapping with other buttons or text boxes
* “Instructions ” button brings the user to the instruction page
* “Settings” button brings the user to the setting page

## **Integration Testing**

**Our integration testing will be done by incremental integration testing. This is a kind of bottom up approach, so we will do unit testing first for functions and classes. Then, the ones that need to be combined will be combined and tested together, until eventually we have all of our code working together and integrated.**

**In addition to integration testing with our functions and classes, integration testing will include testing the graphical user interface to make sure it displays everything properly. Then we will test what happens when the dummy wordlist is incorporated into the graphical user interface. Through all of this process, each of the functions will be tested as we go, and at the end then the entire GUI will be compiled and working. Once we have the GUI working with a dummy wordlist (dummy being a short list of our own creation), then we will test how the logic and GUI interface of our game will work when we pull a wordlist and store it on the device. This testing is different than unit testing as the unit testing is testing functions and classes as we go (for example in the GUI) but this testing how each of the parts of the entire system will interact with each other. The logic of the game will go from integrating a dummy wordlist to integrating a pulled wordlist from a server location, and this will be the main difference between the self contained version of our game and the version of the game where a user specific wordlist is pulled when username and password are entered.**

## ****Code Line and File records and other testing and validation methods****

**As we go through the project, xcode will document the number of lines of code and the number of classes and files (these are easily countable for a small project like this). We can graph this data and keep it saved in Microsoft Excel. We can show how the size of the project changes over time using graphs in Microsoft Excel, such as lines of code we have per week for example.**

**Other things we are doing to ensure the quality of our project is to ask classmates whether the user interface is friendly and if the game is fun and useful enough at certain points. This ensures that we as the developers won’t be the ones doing all of the testing, but we use people who are not familiar with our project to test our product. This gives a different perspective on the usefulness and usability for the common person, so that we remain focused on the end goal, which is developing an app useful for the common user. Doing this also helps us keep our app simple and effective. As well, the validation stage is when we make sure what we built is what was required to be built as the final product, and this will take place with the submission of version 3 with Norbert Renert, who is the maker of the SynPhony literacy system.**