CMPT 275 Quality Assurance Plan

SafeSpeed

Group Number: 4
Group Name: Two Squared
Group Members:
Linda Jia
Maryna Zarud
Xiao Ye (Jada) Jin
Yan Gu
Zhaozheng (Vincent) Han
Due: Feburary 8, 2013

Table of Contents

Table of Revisions

Table1 shows revisions of the document

Revision	Status	Publication/Revision Date	By
1.0	Created	February 6 , 2013	Jada Jin
2.0	Added all sections	February 7, 2013	Jada Jin Linda Jia Maryna Zarud Vincent Han Yan Gu
3.0	Revised section 4	February 8, 2013	Jada Jin Linda Jia

Software Testing

Unit Testing

Our team will continuously execute unit testing to find problems in our code early on and to ensure that the integration process is as smooth as possible. We will use black-box testing, in which we examine the functionality of our classes and methods. To do this we will isolate basic functions, set up independent test cases for that specific function, and write tests that include both good and bad input. As the project progresses, will use white-box testing to test the internal structures of our code for possible errors.

We will work with OCUint, the integrated unit testing function of XCode, which is based on the SenTesting framework.

Other Testing Tools and Resources

If our group finds the need for additional testing, some other possible frameworks available are as follows:

- Google Toolbox for Mac (GTM) (http://code.google.com/p/google-toolbox-for-mac/)
- GHUnit (https://github.com/gabriel/gh-unit/)

As well, the iOS Developer Library has a Unit Testing guide at:
http://developer.apple.com/library/ios/#documentation/DeveloperTools/Conceptual/UnitTesting/00-About_Unit_Testing/about.html#//apple_ref/doc/uid/TP40002143

Internal Deadlines

We have set internal deadlines for unit/system testing of the code for release. These dates are flexible and may be changed based on task priorities at any given time.

Table 2 shows the internal deadlines table, which contains a task, start date, due date, number of days, unit test due date and system test due date.

Task	Start Implementation	Due Date	Number of Days	Unit Test Due Date	System Test Due Date
Assignment 3 ➢ Version1	08/02/2013	18/02/2013	10 days	13/02/2013	15/02/2013
Assignment 4 ➢ Version2	25/02 2013	13/03/2013	16 days	06/03/2013	08/03/2013
Assignment 5 ➢ Version3	13/03/2013	25/03/2013	12 days	22/03/2013	20/03/2013

User Acceptance Testing

Date time and location: TBD (between March 25, 2013 to March 30, 2013)

Testing time will be scheduled and testing location will be decided base on our appointment with the CPC and VPD.

Target testers:

- The CPC volunteers: 2 groups, about 5 people.
- The VPD officers: 1-2 people.

During application testing, users will be provided with an iPhone and an iPod touch with the final version of the application installed.

During web based function testing a Personal Computer, iPhone, and iPod touch with the final version of the application installed will be provided. All will have internet accessibility.

Users will complete the following sessions during the acceptance testing:

Warm up session:

During the testing, no assistance and help will be given by the observers from our team. Users will follow the in-application instructions and help information to complete the testing tasks. There will be a warm up session given and one of the observers from the development team will give volunteers a short introduction to help them get the big picture of the application. After that, the volunteers can try the application for 10 – 20 minutes before then going to a real speed watching session. During this, they can take any note they feel necessary about their observations. A note sheet and pen will be provided.

Phase 1:

This phase is going to test user acceptance of car data recording function of our app. This will occur during a speed watch session. The duration of each session of phase one will be shorter than the duration of a typical speed watching session. There will be 3 to 5 such testing sessions.

There will be one group given a device with the app before the session begins to create a brand new profile with their personal information. They will follow their initial training from the warm up session to complete one speed watch session that will last 10 minutes.

Between two sessions, volunteers will be given time to reflect on their experience from the previous session and take notes about it. They may also change their recording habits to be better acquainted with the app. While volunteers are performing the testing, there will be a control group helping record the data manually using a pen and form. The manually recorded data will later be used to evaluate and compare with the app data.

Phase 2:

This phase will test user acceptance of our data upload functions from the device and data viewing and summarizing functions using the web based sever. This session will be held after the data recording sessions.

During this phase, one representative of each volunteer group will perform the testing. They will use their iPhone or iPod touch to upload data while having internet connection, and generate a summary of session reports via the web based server.

Phase 3:

In this phase, web generated data summary will be viewed by the CPC coordinator and VPD officer for further evaluation and feedback.

During phase 1 and 2, observers form the development team will take note on the operation of application by volunteers. After phase 1 is completed, participating volunteers will be given questionnaires to fill out. After the all phases are completed, the CPC coordinator and VPD officer will be interviewed by the development team.

All of the questionnaires and interview questions will be proposed and prepared during the development process.

Integration Testing

The integration testing method we will be using is bottom-up testing so we are going to test the lowest level components first and proceed to higher level components. This testing is different from unit testing because we are integrating several units together to test a broader and more general function. Integration testing will be done in several steps:

1. Integrate User class, iPhone database to:
   a. Create a few new users;
   b. Test if user authorities are properly assigned upon log in after step a;
   c. Test if user profiles are correctly saved;
   d. Test if upon logging off, user authority is properly cleared;
   e. Test if user profile can be correctly modified and saved.

2. Integrate SpeedWatchSession class, iPhone database to:
   a. Create a few new speed watch sessions;
   b. Test if speed watch sessions are properly retrieved from database;
   c. Test if speed watch session data is correctly saved;
   d. Test if speed watch session data can be correctly modified and saved.
3. Integrate Car class, iPhone database to:
   a. Create a few new car objects;
   b. Test if cars are properly retrieved from database;
   c. Test if car data is correctly saved;
   d. Test if car data can be correctly modified and saved.
4. Integrate iPhone database, online database to:
   a. Export data saved through step 1, 2, 3 to online database;
   b. Test if data is correctly stored in online database.
5. Integrate Report class, online database to complete integration:
   a. Extract data from online database to Report class
   b. Test if users of Report class are properly assigned authority upon log in;
   c. Test if user information can be correctly modified and saved;
   d. Test if speed watch data and volunteer information can be managed and saved through Report class;
6. Perform a system test:
   a. For a specific user, perform a speed watch session;
   b. View report, graph, and map outputted on corresponding website.

Software Metrics

We will use the built-in property of XCode, Preferences to track the number of lines of code. In addition, XCode Statistician (http://www.literatureandlatte.com/freestuff/XcodeStatistician.zip) is an application for tracking the number of statements, characters, words, and classes that have been created. With this application, statistics such as the number of classes will be automatically tracked and displayed, helping us to estimate code complexity.

Additional tools that are readily available online will also be used to assist us with software metrics estimation. The list of tools is as follows:

- Clang Static Analyzer (http://clang-analyzer.llvm.org): applies algorithms and techniques on code to automatically discover bugs
- Doxygen (http://www.stack.nl/~dimitri/doxygen): documentation generation tool that can be used for generating class dependency diagrams
- HFCCA (http://code.google.com/p/headerfile-free-cyclomatic-complexity-analyzer): a tool for calculating cycolomatic complexity of Objective C code

We have estimated the complexity of our software based on the three phases. We estimate significant growth in lines of code in phase 1 and 2, and a less rapid increase in number of lines in phase 3. Complexity estimation is as follows:

Table 3 shows the project phases, number of lines of code, number of classes, and number of files.

Phase	Number of Lines	Number of Classes	Number of Files
1	4000	9	18
2	8500	10	20
3	11000	10	20
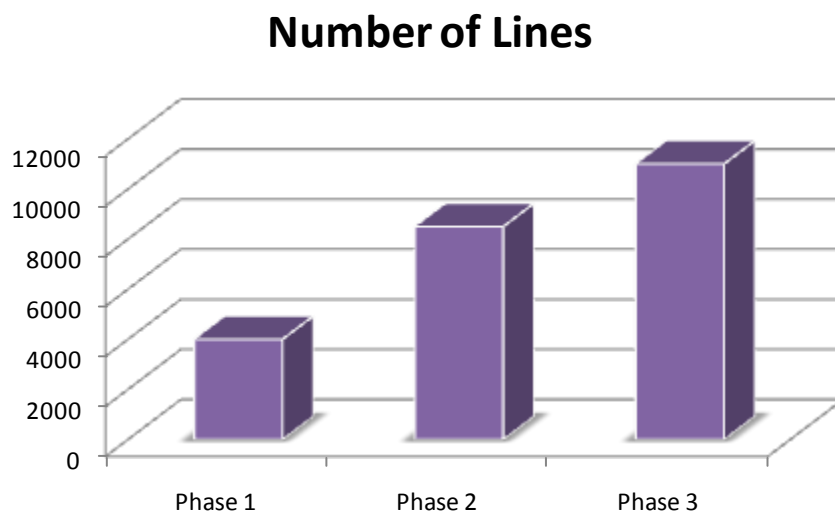
Number of Lines

Figure 1 shows a bar-chart comparison of the number of lines of code across the three phases.
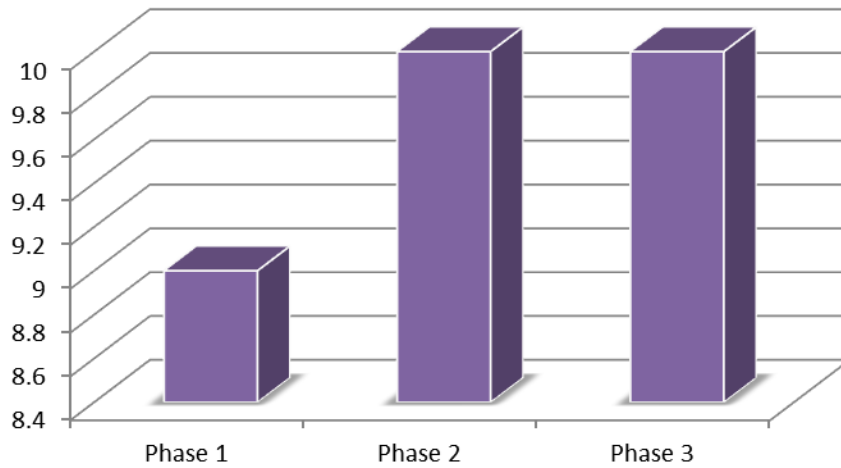
Number of Classes

Figure 2 shows a bar-chart comparison of the number of classes across the three phases[1].
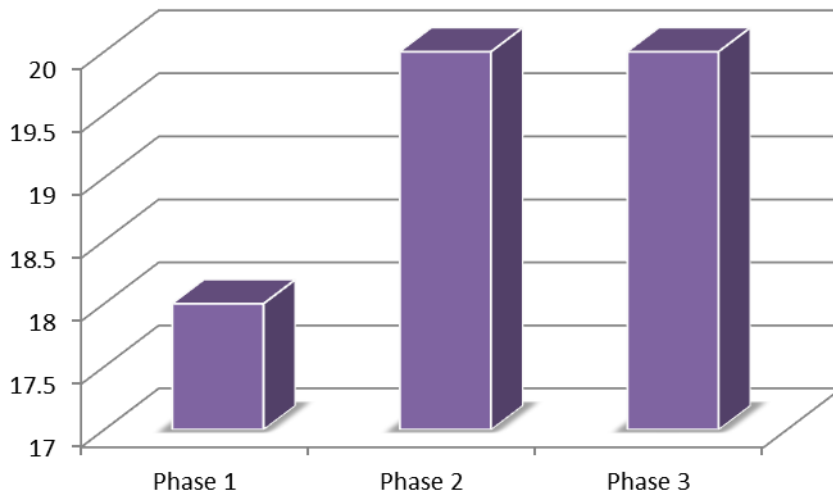
Number of Files

Figure 3 shows a bar-chart comparison of the number of files across the three phases[1].

---

[1] The number of classes and files do not change between phase 2 and 3. This is because in phase 2 we will implement all classes, but not all the methods in the classes. Phase 3 will implement the rest of the methods.

Appendix A – Team Meeting Minutes

Group #4 – Two Squared
Purpose: Working on Assignment 2
January 25, 2013 at 1:30 PM
Chair: Maryna Zarud
Attendee: Jada Jin, Linda Jia, Maryna Zarud, Vincent Han, Yan Gu
Absent: None

Topic	Discussion	Action/Decision	Person responsible /Due date
General Assignment 2 discussion	• looked over the assignment and discussed some of the details, and things that might need further research	-	-
App Features	• discussed the specific app features we want to include in our documentation	• everyone needs to create a list of features	All are responsible, due January 28th.
App UI	• discussed what kind of pages, buttons, and features the app will have, as well as ways to ensure that it is fast	• everyone needs to come up with a potential UI design for the next meeting	All are responsible, due January 28th.

Group #4 – Two Squared
Purpose: Working on Requirements Document
January 28, 2013 at 3:00 PM
Chair: Maryna Zarud
Attendee: Jada Jin, Linda Jia, Maryna Zarud, Vincent Han, Yan Gu
Absent: None

Topic	Discussion	Action/Decision	Person responsible /Due date
App Features	• looked over everyone's list of potential features and created a final, cumulative, list	-	-
Splitting up Requirements Document	• discussed how to split up the work of the requirements document	• Yan Gu – putting it all together • Maryna Zarud and Jada Jin – Example Tutorials •Linda Jia – Features and Non-Functional requirements • Vincent Han – Technical Specifications	Everyone must finish their part by February 1st.

Group #4 – Two Squared
Purpose: Working on Design Document
February 1, 2013 at 1:30 PM
Chair: Maryna Zarud
Attendee: Jada Jin, Linda Jia, Maryna Zarud, Vincent Han, Yan Gu
Absent: None

Topic	Discussion	Action/Decision	Person responsible /Due date
Split up the diagrams of the Design Document	• discussed which diagrams we want to create to represent our project best	**Yan Gu** – putting together document **Jada Jin and Linda Jia** – class and database diagram **Maryna Zarud** – state diagram **Vincent Han** – CASE diagram	Everyone is responsible for their part for February 4th.

Group #4 – Two Squared
Purpose: Working on Assignment 2
February 4, 2013 at 2:30 PM
Chair: Maryna Zarud
Attendee: Jada Jin, Linda Jia, Maryna Zarud, Vincent Han, Yan Gu
Absent: None

Topic	Discussion	Action/Decision	Person responsible /Due date
Design Document Diagrams	• discussed the CASE, class, database, and state diagrams everyone prepared, and made the necessary corrections	• most diagrams stayed the same, some of the wording was changed, and a 'Hep Page' was added to the app	-
QA Document	• split up the work of the QA document by the numbered tasks on the assignment description (1-5)	**1-**Maryna Zarud **2 and putting document together-**Jada Jin **3-**Vincent Han **4-**Yan Gu **5-**Linda Jia	Everyone is responsible to do research and complete most of their part for February 6th.

Group #4 – Two Squared
Purpose: Progress meeting for Assignment 2
February 6, 2013 at 1:30 PM
Chair: Maryna Zarud
Attendee: Jada Jin, Linda Jia, Maryna Zarud, Vincent Han
Absent: Yan Gu

Topic	Discussion	Action/Decision	Person responsible /Due date
QA Document	• discussed the research everyone did about testing tools and the finalization of the QA document	-	Jada Jin must put together document for February 8th.
Assignment 3	• discussed possible coding practices, how to split up the work, and when the coding will begin	• Everyone should set up a coding environment and get acquainted with XCode and Objective-C	Complete this by February 11th, which is the next meeting.

The Meeting Minutes can also be found at **http://www.sfu.ca/~jadaj/Assignments.html**