DigiProf Assurance Plan

Group 13

1/ Software Tools

- Automated unit testing tools/frameworks:

Embunit	• In C or C++. • Similar to Junit	• Single user license: £95 • Desktop version is free	http://www.qatestingtools.com/testing-tool/embunit#
Emma	• Purely in Java. • Supported coverage types: class, method, line, basic block. • Coverage stats are aggregated at method, class, package, and all classes levels. • Can instrument individual .class files or entire .jars. • No external library dependencies.	• Free	http://emma.sourceforge.net/ download link: https://sourceforge.net/projects/emma/files/
Junit 5	• For Java. • Simple • Test data is first tested and then inserted in the piece of code -> test-driven.	• Free	https://junit.org/junit5/ git link: https://github.com/junit-team/junit5/
TestNG	• Supports data-driven testing along with unit, functional and integration testing. • Effective with flexible test configuration. • Supports parameterized testing.	• Free	https://testng.org/doc/ git link: https://github.com/cbeust/testng

The chosen tool is TestNG due to all the mentioned features.

- Integration testing tools/frameworks:

FitNesse	• Open Source. • Provides support to Java, C#, Python, and even C++. • Allows validating the requirements with actual software implementation. • Specifications can be used as test input.	• Free	http://www.fitnesse.org/ download link: http://www.fitnesse.org/FitNesseDownload

Spock	• Compatible with different IDEs. • Can perform both assertion checking and mocking at the same time. • In Java. • Supports data-driven testing.	• Free	http://spockframework.org/ Git link: https://github.com/spockframework
JUnit	• As listed above	•	
Mockito	• Produce clean verification errors. • Easy to use. • Allows flexible verification.	• Free	https://site.mockito.org/

The tool that our team is using for integration testing is Spock because both Spock and TestNG support data-driven testing. The other features are mostly similar among the testing tools.

- Automated software testing tools for system testing and software testing in general:

Appium	• In java and can be used to test Android apps. • No need to recompile app. • Allows parallel execution of test scripts. • A small change does not require re-installation of the application. • A bit slow compared to other apps. • Easy to use.	• Free	http://appium.io/
Kobiton	• Supports real devices testing. • Can gain access cloud or local devices directly from developer workstation within IDE. • Powerful APIs.	• Free but might have to pay more. • The free version only includes 30 minites pf testing per month.	https://kobiton.com/
Ranorex	• Supports real devices testing. • Can simulate real user interaction. • Easy to use. • Allows parallel mobile testing. • Supports data-driven testing.	• Free trial	https://www.ranorex.com/prices/

Robotium	• Requires minimal knowledge of the application under test. • Fast test case execution. • Minimal time needed to write test cases.	• Free	Git link: https://github.com/RobotiumTech/robotium

Appium is chosen as the testing tool for system testing and software testing in general. What truly differentiates Appium from other tools is how it functions as a server and runs in the background.

- Validation testing: ensure that the hardware/software system meets the requirements allocated to software as identified initially. This can be seen as black box testing (testing that ignores the internal code of the system) -> This one is done manually.

- User acceptance testing: This part is for users to use the workable version of the app and test out features -> The testing is performed manually.

- Writing tests in parallel to the development of the task help enhance team collaboration. Test cases are written down and managed at all time in a doc file. Any modification to the app will result in the changing of some of the old cases.

2/ Internal deadlines:

Unit testing	11/4/20 - 12/04/20
Integration Testing	11/5/20 - 12/04/20
System Testing	11/6/20 - 12/04/20
Acceptance Testing	11/9/20 - 12/04/20

All of the testing will be performed frequently and regression testing is done by the end of every system testing.

3/ User acceptance testing:
The version 3 of the app is speculated to be useable by 12/07/2020. Therefore, the initial plan is to have the version 2 get tested on 11/16/2020 and finish within 2 days. The user acceptance testing for version 3 of the app will start on 11/23/2020 and be carried out for the following days.

First of all, the user acceptance testing scope is defined. This also includes all of the assumptions and constraints of the test. All of the test cases will be identified to make sure that they sufficiently cover most of User Acceptance Testing scenarios. The users will then be asked to use the app and provide feedback. After the testing is done, we will assess the collected data to improve future test cases and UAT workflows.

4/ Integration testing approach:

- There are 4 common approaches:

Big Bang	Testing where all or most of the units are combined together and tested at 1 go.
Top Down	Top-level units are tested first and lower level units are tested step by step after that.

Bottom Up	Same as top down but in the opposite way around.
Sandwich/ Hybrid	A combination of Top Down and Bottom Up approaches.

The team is currently following the Bottom Up approach for Unit Testing where the testing is performed in the lowest level units first and then gradually move on to other units. However, we are planning on using the Sandwich approach for our Integration testing as it allows parallel testing and appears to be a more time efficient approach. There is also a major flaw when utilizing the Bottom Up approach due to it unable to test out the working application until the last module is built. Hybrid (Sandwich) approach is carried out by firstly identifying the middle layer from which the bottom up and top down testing are done. Then the top down and bottom up testing start simultaneously until there is only the middle layer left to be tested at the end.

5/ Metrics collection tools:

The team's software tool of choice is CyVis - a software metrics collection, analysis, and visualisation tool for Java based programs. The perks of using CyVis include:

- Detailed Metrics collection at Project, Package, and Class levels.
- Metrics collection from class files. This helps exclude the need for source code.
- Metrics are shown in tables and charts for better visualization.

6/ Additional actions

As a team, we take further steps to ensure the quality of the app:

- Pair programming is preferred among members because some of it appears to be an effective way to share knowledge and makes sure that fewer coding mistakes are made.
- Code review session is regularly held with Andy Wang – our Project manager being the host as he goes through the code and provides provide feedback. The session can be seen as a place for everyone to receive criticisms on their obscure code and improve themselves.
- We have a fixed stand-up meeting schedule on every Monday. This helps avoid all the long, redundant and meaningless meetings that we would normally have at the beginning of the project. Furthermore, it allows all the team member to have more time to focus on their assigned tasks.

References:

[1] "20 Most Popular Unit Testing Tools in 2020", *Softwaretestinghelp.com*, 2020. [Online]. Available: https://www.softwaretestinghelp.com/unit-testing-tools/. [Accessed: 31- Oct- 2020].

[2] "CyVis - Software Complexity Visualiser", *Cyvis.sourceforge.net*, 2020. [Online]. Available: http://cyvis.sourceforge.net/. [Accessed: 31- Oct- 2020].

[3] "Integration Testing - SOFTWARE TESTING Fundamentals", *SOFTWARE TESTING Fundamentals*, 2020. [Online]. Available: https://softwaretestingfundamentals.com/integration-testing/. [Accessed: 31- Oct- 2020].

[4] "Top 5 Automated Testing Tools for Android: October 2018", *Bugfender.com*, 2020. [Online]. Available: https://bugfender.com/blog/best-automated-testing-tools-android/. [Accessed: 31- Oct- 2020].

[5] "Top 10 Integration Testing Tools to Write Integration Tests", *Softwaretestinghelp.com*, 2020. [Online]. Available: https://www.softwaretestinghelp.com/integration-testing-tools/. [Accessed: 31- Oct- 2020].