Software Testing and Debugging

Milestone 2 - Lessons

Introduction

In this report we will detail the things we gained from developing tests and structure for the Android application MinimalTODO. For an android application that did not have any unit tests there were many challenges surrounding the infrastructure and using several tools for the first time.

Using Mocks, Fakes, JUnit and Simulating Behavior:

One of the learnings in this whole project was to understand and utilize the use of mock objects, fake objects and then controlling the behavior of the mocks to test for all possible cases for failure and testing developer behavior. For example, if there is a part of the code where we are handling exceptions, we could ensure that the exception is triggered and handled appropriately.

Using the Android Framework and Gradle:

Another positive from this project was understanding how to trigger and build test cases for the entire project. Also, understanding scripting and automation made us realize how powerful test automation can become.

Coverage alone is not enough:

Another lesson that was reinforced in our head is that code/test coverage alone does not mean anything and cannot uncover any bugs.

We cannot test badly written Code:

We learnt that tests are not a cover for badly written code. In this project we had to change several classes to get them to be test compatible. For example, using dependency injection and construction rather than creating new objects inside methods for example, which can become hard to examine and test.

Stress Testing:

In stress we checked the ability of a program or device to maintain a certain level of effectiveness under any conditions. This process measures the frequency of errors or system crashes. With this testing, we learned about the Synchronization and timing bugs, Priority problems, Memory leaks, Data loss & corruption etc. and we potentially identified the breaking points, which will allow us to correct them before they become expensive issues in later on. For this app, we have done stress testing for 100 Processors and 100 loops. It takes around 115mins to complete the entire activity.

```
./StressTestApp.sh -NumBackGroundProcesses 10 -NumberLoop 10
```

Monkey Testing:

The Monkey is a command line tool that runs on your emulator or device and generates random streams of user events. Here, we have used Monkey command to stress out the applications in a random yet repeatable manner. To run monkey testing, we launched the Monkey command on our development machine. Because the Monkey runs in the emulator/device, we had to launch the application and it should be connected to the device before run monkey command from a shell and below is the command for our app:

```
$ adb shell monkey -p com.avjindersinghsekhon.minimaltodo -v 500
```