

iDEN™ Smartphone Embedded Software Testing

Wei Hoo Chong

Global Software Group, Motorola Multimedia Malaysia

Email: hoochong@motorola.com

Abstract

iRobot is designed as a tool to perform stress test on iDEN™ Gemini Windows® Mobile Smartphone. Gemini is the first iDEN™ Smartphone from Motorola®. We will discuss the value that automation had brought to iDEN™ Phone testing which has contributed to the improvement of software quality in iDEN™ phone together with the metrics. We will also look into the advantages and disadvantage of the proposed system and some discussion of the future work.

Keywords: stress test, embedded software testing

1. Introduction

Software testing is the activity of running a series of dynamic executions of software programs that occurs after the software source code had been developed. It is performed to uncover and correct as many of the potential errors as possible before the software is delivered to the customer [1]. In Motorola®, Automation Testing is a methodology used by GSG-iSGT (Global Software Group - iDEN™ Subscriber Group Testing) to increase the testing volume, productivity and reduces test cycle-time in cell phone software testing, and makes the products become more robust before releasing to the market. In order to achieve this goal, one of the methods used is to stress the phone to its limit and, if the phones still maintain its stability.

Currently, the automation tools GSG-iSGT team using are iPTF (iDEN™ Phone Test Framework) [2][3], iHopper [4], iRobot [5], and etc. iPTF is used to execute the scripts on the test phone. It transforms the script (written in Java) into commands that are understandable by the phone, as there is a test agent in iPTF, to test the phone. However, iHopper is a stress testing tool that is able to drive the phone randomly through the UI to stress the iDEN™ UIS phones software; as iHopper is using the same iPTF test agent.

2. iRobot

In this paper, iRobot has been introduced to cater the stress testing on the Windows® CE platform device. iRobot is an embedded application/tool to stress test the iDEN™ Gemini Smartphone. The concept is that the phone is driven randomly to stress the phone software, which is almost similar to the iHopper. The recorded scripts (either one or multiple scripts) can be replayed for multiple times at anytime and at any geography location as the iRobot is resided in the subscriber unit (SU). The scripts and pattern files can be transferred and replayed on another Smartphone. Pattern File is a file that contained a set of predefined key

presses logs that can be replayed at multiple times specified by the user. There are 2 methods to obtain these key presses logs; 1) through recording 2) through the log files that were created after every session of test execution. There is a 'Record' option in the Menu for user to choose. It is able to record any key presses done by the user into a specified key presses log file and the recording can be stop when 'Stop' is selected. Besides being an input to Pattern File, the log file as shown in Figure 1 is very useful to test engineer or developer in analyzing the root cause if failure happened. Figure 2 illustrates some built-in tests such as data call, phone call, and random test (random key presses) that had been integrated into the iRobot, in a hope to stress test the Smartphone thoroughly. Data call is able to stress the phone through the numerous loadings of a specified Uniform Resource Locator, URL. Phone Call is to stress the phone through the numerous dialing of a specified phone number.



Figure 1: iRobot Log File



Figure 2: Built-in Test Tools

3. Success Story

Figure 3 illustrates the number of defects discovered by iRobot automated stress testing tool in 3 weeks. This data consists of various features tested, several of software release version and it was based on only one engineer in GSG-iSGT team with more than 10 Gemini phones. The automated test was executed 24 hours per day without any human intervention. The test engineer was only required to configure all the necessary settings in the Smartphone with test scripts required for testing. The iRobot acted like a virtual tester and logs the results to a log file by itself.

There are 2 peaks in the graph, which are 7 defects on the first day of testing, and 5 defects few days after that. This is because the coverage area of testing was different. The defects found such as "Unknown" item was not displayed in the recent call list after received a missed call from an unknown number and the most serious issue is the phone hung after stress test. In conclusion, we noticed that the Smartphone become more robust at the end of testing before the product is released to the market. Hence, the quality of the product can be maintained and improved.

On the other hand, the manual testing cycle time is reduced at least 3X to deliver the same quantity of job with automated testing with each iRobot or Smartphone.

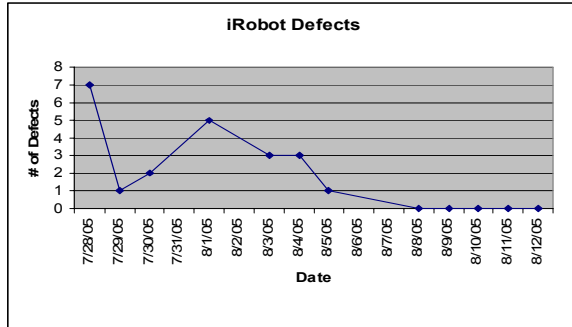


Figure 3: Number of defects discovered by iRobot. Baseline is Zero defect (as low defects as possible is better)

We assume that each engineer only works 8 hours per day consider as one Staff day and each iRobot will execute for 24 hours per day. As a result, we are able to reduce the cycle times up to 3X. Please refer to the equation (1) below.

$$\text{Cycle Time} = \frac{(R * I_N)}{S_D} \quad (1)$$

Where

R – Number of hours iRobot executes

I_N – Number of Smartphone used

S_D – One staff day in hours

With the less cycle time the productivity will increase. Since the number of test cases vary from cycle to cycle, the productivity is very important to determine the testing cycle time in a test cycle. This is shown by the equation (2) below.

$$\text{Productivity } \alpha = \frac{TC}{\text{Cycle Time}} \quad (2)$$

The productivity is inversely proportional to cycle time spent on testing and proportional to number of test cases assigned, TC. This is because of the recordings need only to be made once each time there is a new phone setting. Besides, users do not have to keep an eye over each test run. They can check the results in the log file after each execution. Thus they will have the time to focus on other tasks while the stress test is running on the phone.

4. Advantages of iRobot

Besides the advantages above mentioned, iRobot enables

- Reusable: The scripts recorded or key-presses logs can be reused in the future as well as on another Smartphone, thus saving the time of having to record a same script over and over again.
- Reliable: Eliminating human error because tests perform precisely the same operations each time they are run.
- Repeatable: A set of tests can be run repeatedly.
- Programmable: You can program sophisticated tests that bring out hidden information from the released phone software.

- Flexible: No special attention is needed too, and users can leave the scripts running without having to keep an eye over it as the result of each test will be logged in a log file. Users need only to refer back to the log file which contains easy-to-read yet detailed information of every test run.
- Saving Cost: as less resource are needed to operate the test.
- No programming knowledge is required from the users as the tool can be operated easily through a few key-presses.

In short, this tool serves a lot in time yet it allows means to stress the phone to its maximum.

5. Disadvantages of iRobot

There is only one main setback in iRobot as it only supports Smartphone for the time being.

6. Future Works

iRobot still can be enhanced to have more features, such as, to provide a list of script files for user to select to Playback. Besides, the iRobot can be enhanced to have the same capability as iPTF which is able to verify the functionality in the released phone software. Thus, besides the stress testing, manual testing is still needed in the System Test cycle to fully test the phone capability and functionality as iPTF is not supports Smartphone.

7. Conclusion

This paper studies automated testing tool, named iRobot, in stressing the Smartphone. By doing this, we are able to determine the limit of the Smartphone. Thus we are able to further improve it if the limit is unsatisfactory before the actual release of the product to the customer. In additional, iRobot provides seamless stress testing environment as it is an embedded testing tool. Thus, the testing can be carrying out anytime, at anywhere. In short, the tool enables us to improve on the quality of the phone software, and increasing the efficiency of the testing.

8. References

- [1] W H C Bassetti, W. E Lewis, "Software Testing and Continuous Quality Improvement", 2004, pp. 289-290.
- [2] Wei Hoo Chong, Market-Based Resource Sharing Job Scheduler, *Proceedings of the International Conference on Information Technology and Multimedia*, ICIMU'05, Malaysia, 2005.
- [3] Wei Hoo Chong, iDENTM Phoens Automated Testing in P2P, icns'06, ISBN 0-7695-2622-5, 2006.
- [4] iDEN Subscriber Test Group, iHopper User Guide, version 00.01.00, Motorola Malaysia Software Center, 2005.
- [5] iDEN Subscriber Test Group, Design Document for iRobot, version 0.0.1, Motorola Malaysia Software Center, 2004.