

An Efficient Model for Mobile Application Regression Test for Agile Scrum Software Development

¹Abu Wahid Md. Masud Parvez

¹Project Manager and Quality Architect, Tech Propulsion Labs, USA

Email: niloy.cit1@gmail.com, masud@techpropulsionlabs.com

Abstract – As an answer of long growing frustration of waterfall Software development life cycle concepts agile software development concept was evolved on 90's. The most popular agile methodologies include Extreme Programming (XP), Scrum, Crystal, Dynamic Systems Development Method (DSDM), Lean Development, and Feature-Driven Development (FDD). Actually scrum is rather than a full process or methodology, it is a framework. So instead of providing complete, detailed descriptions, without a lot of paper works, everything is to be done on the project. This is done because the team will know best how to solve the problem they are presented. Regression test is also a very important part of software quality control process. Regression testing means re-testing an application after its code has been modified to verify that it still functions correctly. Regression testing consists of re-running existing test cases and checking that code changes did not break any previously working functions, inadvertently introduce errors or cause earlier fixed issues to reappear. While working in agile scrum software development methodologies we realized that there is no regression complete test process model available yet which is specialized for agile scrum software development. Sprint is the one of the fundamental part of agile scrum software development. Sprint is an iteration of work during which an increment of product functionality is implemented. The length of sprint can be different. So we developed a process model to divide the whole sprint in to several phases and inject the regression test to achieve a very effective and efficient process.

Keywords – Agile software development; Scrum; Sprint; Process model; Software development life cycle; Regression test;

1. Introduction

After 1990's several software development methodologies began to become popular. All of these development methodologies had a different combination of old ideas, new ideas, and transmuted old ideas. Agile software development is a software development methodology where the processes of working are close collaboration between the programmer team and business experts; face-to-face communication, frequent delivery of new deployable business value, tight, self-organizing teams, and ways to craft the code and the team such that the inevitable requirements churn was not a crisis.[3,5,7] Now a days the mobile application development in the software industry increased and increasing in very large range. Software industry is building application for mobile including all platforms. Among them iPhone and android application development is much more than others.[1,8,9] There are different software developments methodologies are being applied for mobile application development as well. Different companies build up their test strategies for their applications. For mobile application among different test types were vital is regression test.

Regression testing is the collection of all types of software testing that seeks to uncover new software bugs in existing functional and non-functional areas of a system after bug fix, patches or configuration changes, have been made to them. Regression testing confirms that any changes which are making by the development that is walking through to the improvement. This is often found that when software is fixed, emergence of new and/or reemergence of old faults. [1,2,4] All most all software development observes this situation. The main reason it occurs because a fix gets lost through a really poor revision control practices. From experience we have observed that, a fix for a problem will be fragile. [1,8] Frequently, a fix for a problem in one area inadvertently causes a software bug in another area as the function developer have fixed that may have dependencies and that fix create the adverse situation for any one of those dependencies. Another most common reason, often when some feature is redesigned, some of the same mistakes that were made in the original implementation of the feature are made in the redesign.

2. Motivation

A major part of mobile application is entertainment based. Entertainment-related applications are flourishing as well, with mobile games, Twitter and Facebook, and photo uploads becoming fundamental elements of the new mobile lifestyle. This is reflected in the economy. Really interesting information is now annual global sales of mobile phones in units are now roughly equal to sales of laptops, with smartphones accounting for more than 10% of the total. About development, in this agile software development, the product owner or client's requirements changes often. Beside the development is done by sprint by sprint. So one sprint's developed features are always venerable for the development of the later on sprints. So this is always important to keep an eye on the quality of the application software being developed. First of all let's talk what kind of software we build in last one year of time.

Table 1. Software development in our development center on last 12 months.

Platforms	Development % of all
Web Application	35%
iPhone Application	23%
Android application	18%
WP Application	12%
Desktop Application	7%
Others Mobile Platform	5%

From the table we can see that, in our overall development in last one year time we did 35% development for web application. In sole comparison this is leading. But if we compare mobile and web application development than we will find the real scenario. If we integrate all mobile platforms then we can see we worked about 58% in our all over all development. This is representing how much mobile application market depends from software industry. So while working on these huge number of mobile application development we realized that we really need an efficient regression test model as well. But no specialized regression model was available for agile software development. So we started our work to develop one model specialized for agile software development which can be really helpful for whole software industry.

3. Regression tools

Most software development situations it is considered good coding practice that when a bug is located and fixed, a test that exposes the bug is recorded and regularly retested after subsequent changes to the program. Some companies may do it through manual testing procedures; some may do using programming techniques, and rest of the other do it using automated testing tools. Multiple vendors are addressing regression test challenge with a wide array of solutions. Let's talk about some top regression tools:

3.1. Hewlett Packard (HP):

Recently HP has integrated mobile application testing facility in to its recent version of Quick Test Professional

(QTP) solution. HP is also offering an third party solution, which offers an end-user experience management solution for Windows Mobile devices. Beside HP offer an excellent partnership with device anywhere which enable the software team to run the device dependency test.

3.2. Selenium

In the middle of 2012 selenium 2 has been release which is providing the android driver and iPhone driver to run the test on the android and iOS platform. Right now selenium is the most popular freeware tool for regression testing.

3.3. uTest

uTest recently has started support for mobile applications. It has of a host of software testers (more than 14,000 worldwide) who test mobile applications across a variety of operating systems and platforms. They pay the tester by the amount of bugs. Normally uTest testers execute testing scenarios and report bugs to the customer, who then the customer decide about that unexpected issue. This model is useful because human testers actually simulate the practical issue which are the user are going to see while their use on daily life.

3.4. Dexterra

Dexterra's main project for mobile application is Dexterra Concert mobile application development platform. Dexterra has two versions. One of them is the "Enterprise Edition" which is an open, standards-based development platform designed to be deployed within the enterprise and another version is the "Carrier Edition" is targeted at service providers who plan to host the solution as a service.

4. Bad Practices in Regression test

Until now regression test is one of the most misunderstood software testing concepts. Wrong regression test policies increase the risk of post-deployment and business loss. A correct regression test policy could present several recent software failures. Such an example, the Blackberry blackouts of 2007 and 2008 are an unfortunate example of production issues that should have been detected by a regression test suite in combination with other formal testing efforts. In software industry there are some main bad practices in regression test process. Let's talk about them bellow

4.1. Bad Maintenance

Most of the software companies in software industry do not have their standard regression test maintenance policies.

4.2. Lack of Update

Some companies at software industry run the same test suite for similar type of project for several years. They don't update the test plan and suite for time to time.

4.3. Test Everything

Many companies plan to do test everything in regression test phase even for a small patch upgrade. This is a really bad practice as this costs a lot.

4.4. Wrong Test Plan

Many regression test failure happen due to the development team's wrong test plan. The development team picked the wrong spot to implement the regression test. But finally defect happened in another place which was not in the range of regression test.

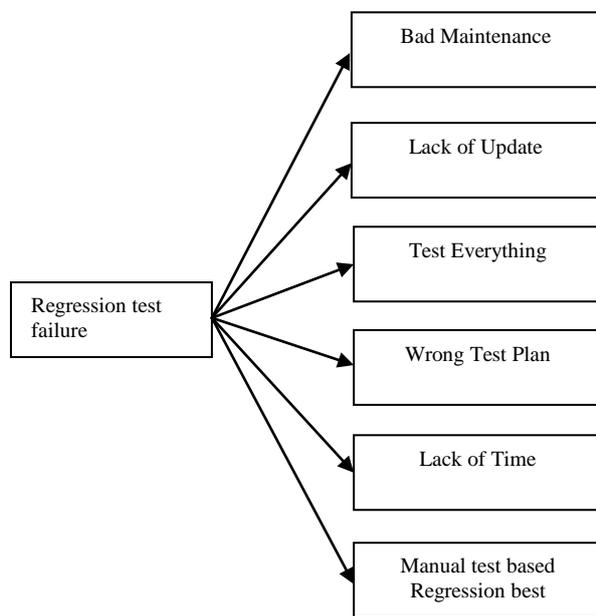


Figure 1. Bad Practices in Regression Test process

4.5. Lack of Time

This often happen that, the Product owner and the development team misjudges the correct regression time length, sometime Product owner does not offer enough time for regression test. These two make the development team cut down many scopes of regression test and finally some products fall in to trouble.

4.6. Manual test based Regression best

Many software companies in software industry still doing largely manual test than automation test. Even some products really need much more manual test than automation test too. The manual test normally cost much more time and labor to complete the regression test process.

5. Our Designed Model

In our designed model we have divided the regression test process in to two parts. One of them is Regression test in development, regression test in feature update

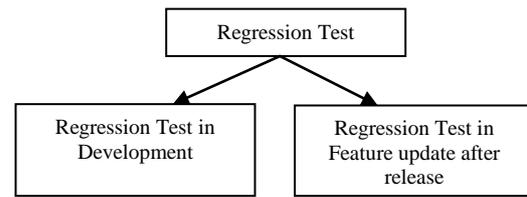


Figure 2. Regression test process division

5.1. Regression Test in Development

We will implement the regression test from the beginning of the development. Agile scrum we have sprints. Such as example for a product we are going to have N number of sprints. So each sprint we will implement the regression test in six stages. I am coming to these stages just after this point. In our designed model, we are implementing the regression test for some very vital reasons. One of the main reasons is, one sprint's development can affect any of the pervious sprint's development backlogs. Such an example while working in sprint 5 the development team can affect any backlogs which are being developed from sprints 1 to 4. Another major reason is, after finishing a sprint sometime the Product owner, wants to change any feature which can bring issue to any of the previous sprint's developed backlog.

5.2. Feature update after release

This is the case when the product already being development and also being delivered but product owner come back to the development team and wants to update any feature. Sometimes a released product comes back to development team again to fix one or more issues which were unveiled during the development period. For implementing that or those feature update the development team may go through 1 or more sprints.

6. Implementation of Regression test in Sprint

In Agile software development we have sprints. In our designed model we are implementing the regression test on each cycle. To implementing the regression test we are dividing the each sprint up to six phases. On each sprints these are six phages of regression tests would be implemented sequentially. let's discuss these six phases in sprints.

6.1. Planning phase

Planning phase is the first phase of the regression test. In sprint the first part is the sprint planning. On this Scrum master (SM) and member of the development team break down the stories and create the backlogs. And then make the estimation for each backlog implementation. After completing the sprint planning the SM and Lead

Developer will review the sprint planning. This is the regression test on planning phase. While reviewing the sprint planning they are going to focus on mainly two things. One is, make sure that all the backlogs are representation the exact specification. And another part is to make sure that all the estimation for developing backlog is as accurate as possible.

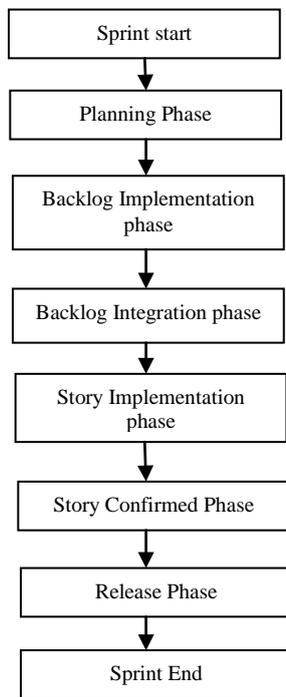


Figure 3. Regression test implementation phases on sprint

6.2. Backlog Implementation phase

Backlog implementation is the second phase for implementing the regression test. In backlog implementation phase the developer start working on each backlog and produce unit of stories. On this time developer has to implement Unit test to confirm their developed unit is working properly. To keep track of each unit’s functionality developer will implement the instrumentation. Instrumentation is the plan of regression test on this second phase. Instrumentation in the process to form of code instructions that monitor specific components in a system. As an example we can say instructions may output logging information to appear on screen. Even this can be used for management tool as well.

6.3. Backlog Integration Phase

This is the third phase in the sprint. On this phase developer integrate the small units and try to form some mini functionality. For regression, on this phase the developer will continue working with the instrumentation and implement mini functional test. This will always inform that if there any regression bugs come up during the integration.

Table 2. Six phase regression test implementation

Phase	Examining subject and Output	Regression implementation	Reponsible
Planning	Sprint Plan and Produce backlogs	Review Sprint plan and Backlog Specification	SM , Developer
Backlog Implementation	Backlog / Unit	Instrumentation	Developer
Backlog Integration	Combination of backlog for story	Mini Functional Test set	Tester
Story Implementation	Completed story	System Test set	Tester
Story confirmed	A story confirmed build	UAT test	Tester
Release	A build with sprint’s all backlog	Early developed backlogs	SM

6.4. Story Implementation Phase

On this phase the developer start integrating all the backlogs of a story to complete the story. On this phase the regression test opportune is agreed by present most software development companies. On this phase the regression test can be implemented in two ways. One way automation implementation and another one is manual implementation or combination of both. The responsible person for this regression test would the quality control engineer or tester.

6.4.1. Automation Implementation

On this process the quality control engineer or tester will write the automation scripts and run them by the automation tools. This process will always give a run time update about the story implementation phase.

6.4.1. Manual Implementation

According to this process the tester will test by the manual process. This is the popular way for small companies. But this cost times which are not always available for the project but allows testing all human test cases. So using the automation implementation way is the recommended way.

6.5. Story confirmed Phase

On the phase the stories are being formed already and developer are not integrating story unit with other stories. This is an important sector for regression test. As while integrating with storied there often may have some dependencies between stories. So while working in the integration of the storied after each integration the tester will check the dependencies to confirm the quality status. The most important part of this regression test is the performance and the dependencies. After phase five the tester will submit a regression test report to SM for reviewing in phase six.

6.6. Release Phase

This is the last phase of regression test in a sprint. On this sprint the SM will confirm that all the regression tests were being implemented in the all previous phases. SM will integrate the regression test result with the release note and attach with the build.

7. Practical Implementation

We took couple of projects. These two projects were almost similar type of projects. Both are online magazine and on android platform. Let's name them as Project X and Project Y. We implemented Project X in regular regression model and Project Y in our design regression process model. Then we compared these two projects in the following way.

Table 3.Project X and Project Y implementation comparison result.

Factor	Project X	Project Y
Regression process	Traditional way	Our designed model
Plat form	Android	Android
Development length	6 months	6 months
Number of regression bugs	A	A + 25%
Process understandability	B	B + 15%
Bug fix Promptly rate	C	C+25%
Average bug life cycle length	D	D - 10%
Time Cost	E	E – 15%
Process failure	F	F – 45%

From the table we can observe that, Project Y and Project X both projects length were 6 months and was implemented in android platform. After implementing both projects, we found that we got 25% more regression bugs in Project Y than project X, the member who joined the team they informed that Project Y's process is 15% more easy to understand and implement than project X's process, the regression bugs where fix 25% more promptly than project X, the average bug life cycle of Project Y was 10% shorter than the project X's bug, Project Y took 15% less time for regression bug process than project Y. Process failure means an unexpected situation created in the development cycle where team does not know how to solve that problem. Project Y got 45% less failure state comparing to the project X. We examine the both Projects and realize that our designed regression process model working much better in all the six criteria.

8. Conclusions

Agile scrum software development is now one of the most effective and popular software development methodologies. We are working in agile scrum software development and while working we was continuously meeting issues in our previous projects and then we developed this model which is specialized for agile scrum software development. This model almost solves our all major issues. In future it would be great if someone work on regression test process model in distributed collaborative development environment which is specialized for agile scrum software development.

Acknowledgements

We would like to thank the whole Tech Propulsion Labs team for their help to implement and examine our model.

References

- [1] Hurdugaci, Victor; Zaidman, Andy (2012), "Aiding Software Developers to Maintain Developer Tests" Software Maintenance and Reengineering (CSMR), Page(s): 11 – 20. IEEE conference publication
- [2] Bohme, M. (2012), "Software regression as change of input partitioning" Software Engineering (ICSE), 2012 34th International Conference on Digital Object Identifier: 10.1109/ ICSE.2012.6227046 , Page(s): 1523 – 1526
- [3] Kun Wu; Chunrong Fang; Zhenyu Chen; Zhihong Zhao,(2012) "Test case prioritization incorporating ordered sequence of program elements Automation of Software Test (AST)", 7th International Workshopon Digital ObjectIdentifier: 10.1109/IWAST.2012.6228980_ Page(s): 124 - 130
- [4] Jin Chen; Mengxiang Lin; Kai Yu; Bing Shao,(2012) "When a GUI Regression Test Failed, What Should be Blamed?" Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference on Digital Object Identifier: 10.1109/ICST.2012.127, Page(s): 467 - 470
- [5] Gu Hongying; Yang Cheng (2011), "A customizable agile software Quality Assurance model", Information Science and Service Science (NISS), 5th international conference, Page(s): 382 – 387
- [6] Waguespack, L.J.; Schiano, W.T.(2012), "SCRUM Project Architecture and Thriving Systems Theory", System Science (HICSS), Page(s): 4943 – 4951.
- [7] Marinescu, P.D.; Cadar, C. , (2012) "make test-zesti: A symbolic execution solution for

- improving regression testing”, Software Engineering (ICSE), 2012 34th International Conference on Digital Object Identifier: 10.1109/ICSE.2012.6227146 , Page(s): 716 - 726
- [8] Salo, O.; Abrahamsson, P, (2005), “Integrating agile software development and software process improvement: a longitudinal case study”, Empirical Software Engineering, Digital Object Identifier: 10.1109/ISESE.2005.1541828
- [9] Szabo, Csaba; Samuelis, Ladislav; Ivanovic, Mirjana; Fesic, Tomas, (2012) “Database refactoring and regression testing of Android mobile applications”, Intelligent Systems and Informatics (SISY), 2012 IEEE 10th Jubilee International Symposium on Digital Object Identifier: 10.1109/SISY.2012.6339502 , Page(s): 135 – 139
- [10] Hu Guang-yong, (2012), “Study and practice of import Scrum agile software development”, Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on Digital Object Identifier: 10.1109/ICCSN.2011.6013698 , Page(s): 217 - 220
- [11] Yudong Zhang, Lenan Wu, Crop Classification by forward neural network with adaptive chaotic particle swarm optimization, Sensors, 11(5) (2011) 4721-4743
- [12] Yudong Zhang, Lenan Wu, Bankruptcy Prediction by Genetic Ant Colony Algorithm, Advanced Materials Research, 186 (2011) 459-463.
- [13] Yudong Zhang, Lenan Wu, A Hybrid TS-PSO Optimization Algorithm, Journal of Convergence Information Technology, 6(5) (2011) 169-174.
- [14] Yudong Zhang, Shuihua Wang, Yuankai Huo, Lenan Wu, Feature Extraction of Brain MRI by Stationary Wavelet Transform and its Applications, Journal of Biological Systems, 18(s) (2010) 115-132.
- Yudong Zhang, Lenan Wu, Cluster Analysis by Variance Ratio Criterion and PSOSQP Algorithm, Advances in Life Science and its Applications, 1(2) (2012) 29-35.
- [15] Yudong Zhang, Lenan Wu, An Mr Brain Images Classifier via Principal Component Analysis and Kernel Support Vector Machine, Progress In Electromagnetics Research, 130 (2012) 369-388.
- [16] Yudong Zhang, Lenan Wu, Shuihua Wang, UCAV path planning based on FSCABC, Information-An International Interdisciplinary Journal, 14(3) (2011) 687-692.
- [17] Yudong Zhang, Shuihua Wang, Lenan Wu, Yuankai Huo, Artificial Immune System for Protein folding model, Journal of Convergence Information Technology, 6(1) (2011) 55-61.

Vita



Abu Wahid Md. Masud Parvez received his Graduation degree at **Computer Science and Information Technology** from **Islamic University of technology (IUT)**. Masud parvez is currently working as **Project Manager & Quality Architect in Tech propulsion labs (USA)**, currently he is posted at Asia branch of the company. Previously he was working as **Research Engineer** in Electronics Research and development center, Walton, and some other multinational companies.