

Agile Testing for Rapid Delivery

Sendhilraj Thangaraj¹

¹Harman International (India) Pvt. Ltd,
Bangalore, India

Abstract. In today's highly competitive world, most of the companies are facing a huge pressure in delivering a high quality product. Old traditional methods have the concept of testing the product after the completion of entire development activities. Since testing was done at the later stage of SDLC, critical bugs were found at the stage when the product was to be released to the market. This use to cost huge money to the project and the delivery was getting postponed. Now in the current fast delivering IT world to sustain in the market, product has to be delivered at a rapid phase. There are many processes and methodologies which can help the organization to deliver products in a faster way. Scrum is one of the fastest emerging methodologies that is being followed in many organizations to develop and deliver quality product to the market. This paper will discuss about a standard testing process that could be followed in a scrum based project to deliver high quality products on time.

Keywords: Agile, Scrum, Testing.

1. Introduction

In the fast delivering software world, the top most priority has to be none other than the customer satisfaction. The customer satisfaction lies on Fast delivery at a minimal cost and with little or no defects. To survive and to keep the existing business live, the companies have achieved this by following some well defined and well believed standard process. Though there are so many processes available which are really good, agile is one of the best available methods that could help the software companies at least to compete and deliver rapidly in the software market. Agile project management with Scrum is derived from the best business practices that were followed in companies like Fuji- Xerox, Honda, Canon, and Toyota [1]. The quality of the software should come from the whole family which is tightly involved in developing and testing the product and not from individual families involved in developing the product. In an Agile process, the quality is embedded into each step taken ahead towards development and deployment. Every shippable product is tested and released. Following the best agile methods could help the agile team to deliver the best quality software more often and more quickly.

2. Scrum

Projects following Scrum [2] Methodology starts with a project planning meeting where all the stake holders and the scrum team are involved. The Product Owner along with the stake holders describes about the scope of the project. The product backlog will have the user stories that need to be completed to ship a working product. The project lifecycle is a composition of certain number of sprints or iterations. Each sprint may be of 2 weeks to 1 month duration. The sprint begins with the sprint planning meeting and ends with the sprint retrospective meeting. In between, the scrum team meets at the daily scrum and discuss about its progress made against the commitment that it has promised to deliver by the end of the sprint. By the end of each sprint the team will demo what it has completed in the specific sprint. After the demo the Sprint

⁺ Corresponding author. Tel.: + 91 80 4330 6435 ; Fax: + 91 80 4095 2689
E-mail address: Sendhilraj.thangaraj@harman.com

retrospective will be conducted to discuss about the work done in previous sprint followed by the sprint planning meeting for the new sprint.

3. Building Stones of Agile Testing

A standard test process as shown in Fig 1 can be followed by any testing team to succeed in any kind of testing. It is always better to remember that any failure to follow the process is always heading you towards the road to failure. In the rest of the paper we will discuss about a standard processes that can be adapted while doing an agile testing in a project to deliver a high quality product right on time.

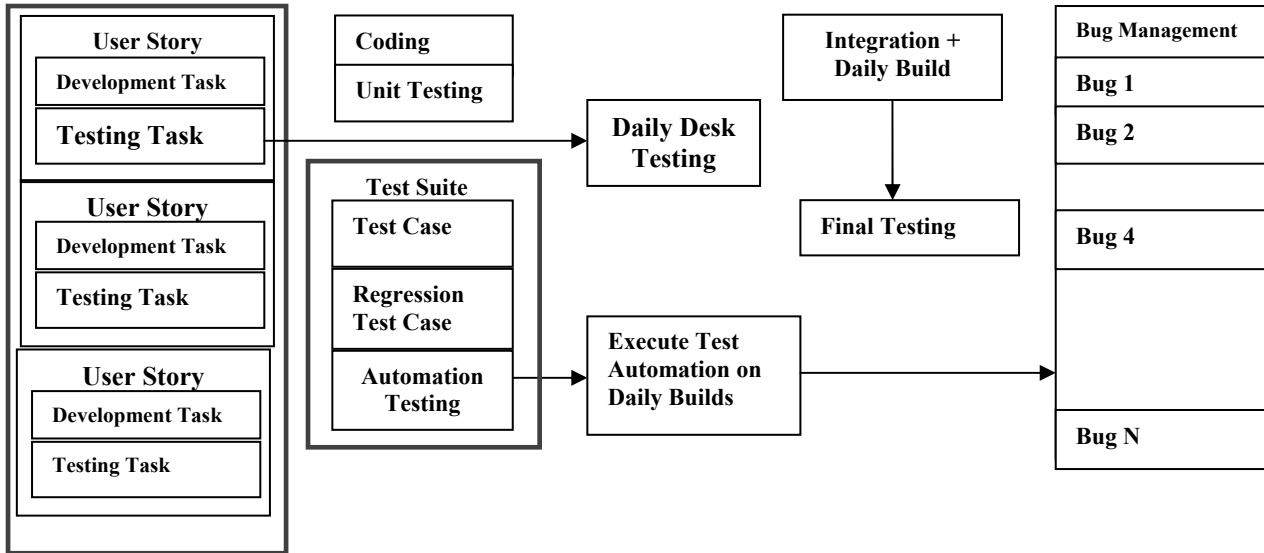


Fig. 1: Building Stones of Agile Testing.

3.1. Setting up Early Metrics in Testing

Metrics are the steps that can be used to build a very strong testing team. Metrics are build based on inspection and analysis of the products and processes of a software project [3].The very basic thing that should be kept in mind while building up any kind of new metrics is try build things from very simple metrics. In testing the very simple metrics like counting the number of bugs logged by the testing team, Number of defects reported by the client, Lines Of code, defect detection ration based on Application Under Test (AUT) can be implemented. In an Agile testing project metrics can be created out of Number of User stories Tested, estimation of each user story, Burn down charts and team velocity. The graphs that can be created out of this data can be surely used by the senior management to forecast any kind of future projects.

3.2. Daily Development and Daily Testing

As soon as the team starts working on the sprint the developer will be busy in developing the code for the user story and the tester will be busy in developing the test case for the user stories. Sometimes the user story may take more than a day to complete .In such case it would be better if the tester can do some kind of testing called Daily Desk Testing as shown in Fig 2. In the initial stage the developer won't like such kind of testing saying that it will hinder his development. This kind of testing will not be more than five to ten minutes. Such testing would make the developer to know the way he is developing the software that will be integrated to existing daily build. Also such kind of testing will also reduce the number of bugs that the developer will uncover during the unit testing.

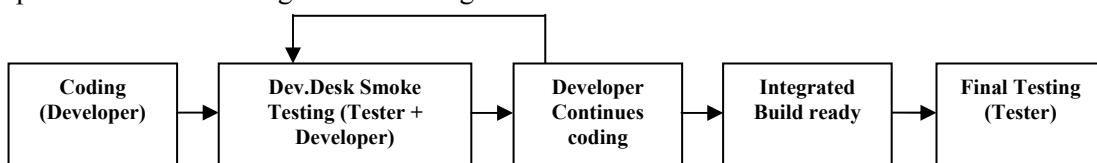


Fig. 2: Daily Desk Testing.

3.3. Test Case Prioritization for Regression Testing

Testing efficiency is an important aspect of the testing phase with very hard expectation riding back on the quality of the software. There are common situation where the test team usually run out of time to test the product. The Product owner and the management will be behind the test team to make it to test more stuffs in a short span of time. In such scenario, the test team has to pick of some test cases to do the testing. Picking up the correct test cases is not an easy task for the testers. Test Case prioritization is one of the methods that the agile team can adopt during the regression testing. During the regular sprint the test team member should prioritize the test case that he is developing for the regression testing. It is inefficient to re execute every test cases for every single change occurring. Test case prioritization techniques help to increase the effectiveness of testing in such a way that the most important test cases are executed at the beginning [4]. It is also well known that the execution of prioritized test cases increases the defect detection ratio. Similar kind of prioritization can be used for executing the regular regression testing and for running the daily automated testing too. One potential goal of test case prioritization is to increase a test suite's rate of fault detection that how quickly that test suite tends to detect faults during the testing process.

3.4. Acceptance Testing

The most important question that arises in the mind of the tester is about the acceptance criteria for a user story. When a story is pulled into a sprint planning meeting, it is the responsibility of the testers to ask questions to the product owner about the user story so that they can always arrive at the acceptance criteria for the user story. Also it is hard for the product owner to write the acceptance criteria for all the user stories. It will be tester's responsibility to help the product owner to arrive at proper acceptance criteria for a user story [5].By having a well defined acceptance tests for all the user stories the product owner gains confidence about the application being developed.

3.5. Unit Test Cases Review

In most of the software projects developers write the unit test case and a peer review is done by another developer. This kind of review is done in a developers view. A small Change in the Unit Test case review process as shown in Fig 3 can yield a better result. It's true that the tester's way of thinking is different and it will be really nice if the unit test case is reviewed by a tester instead of the developer. During this review the developer can explain about the way he has written the unit testing and the tester can suggest his way of thinking to test the same. After this review the developer will provide more unit testing coverage than he normally does. It's always better that if the developer and the tester involve themselves in a very short discussion about the unit testing plan. This kind of discussion will reduce the review time at the later stage. Most of the agile projects have been benefited by following Test Driven Development (TDD) [6].

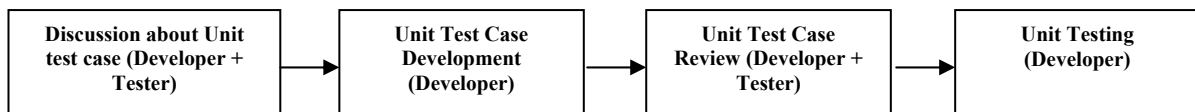


Fig. 3: Unit Test Case Review.

3.6. Test Automation in Sprints

The main challenge in developing scripts for automation is to develop code while the product is being developed. Often teams write code in one sprint and automate the same during the next sprint. By doing so, most of the automation value is lost. [7].It's really costly to imagine a Regression testing phase in a software project without test automation. All smart management deploys test automation in to their project at the very beginning stage. The bitter truth is that the ROI during the initial phase is not so much, but if the test cases are developed following a well defined process, it is a well known truth among the software development companies that the Test automation will help the testing team during the regression phase.

The process of running test automation sprints along with the regular manual testing into the sprint has to come from the top level management. The reason for saying top level management is that the management always ROI immediately and ROI from a well planned Test Automation takes some time. There should be

very clear plan and direction from the top level to guide the automation in a proper direction. To ensure continuous development of automation scripts, it is always better to maintain a separate test automation developers within a project. With daily build being a part of the development teams output, test automation scripts should be used daily on a separate test pc to test the daily builds as shown in Fig 4. Since automation scripts are developed for the GUI that is being developed parallel it always happens that sometimes the script fails due to the change is the GUI. In such cases the existing scripts needs some kind of rework to make it run again. It's always better to have a very generic framework so that it will be easy to incorporate changes into the existing scripts.

One of the key areas that should be paid more attention during the test automation is the way the code is written. It is always better to enforce some coding standards whilst developing the scripts. Instead of doing a manual code review, several static analysis tools like Parasoft's webkings can be used to verify the coding standards.

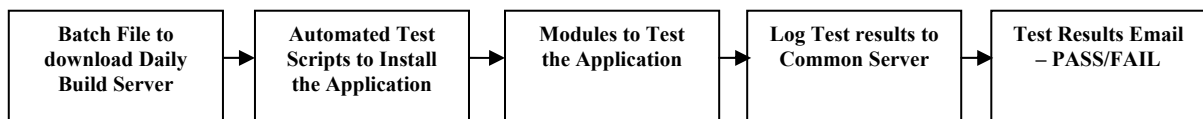


Fig. 4: Daily Test Automation Process.

The key advantage of this kind of automation is that, both the software and the Firmware will be tested before the tester begins the testing activity for the day. Apart from regular testing of daily builds and the daily firmware, automation can also be used to measure the performance of the software.

3.7. Reviewing and Improving Agile Testing Projects

The retrospective meeting is the heart beat of the scrum. This is the meeting where the scrum team looks back and will do a post mortem on its own work to improve further. In an Agile testing project, by the end of the sprint it is common to have a sprint retrospective meeting. The team discuss about the good work they did and the best practices that can be carried forward too. After talking about what went well, it is critical for the Scrum Team to discuss what did not go well during the Sprint. The scrum team needs to discuss about the issues faced during the entire sprint or project and find some solution to solve those problems. Identifying the solution will surely help the team to avoid such mistakes in the future sprint. The retrospective meeting is the place where all the team members are expected to be more open and communicative. Since it is a meeting where things are getting reviewed, team members should strongly view this meeting as a place where things are discussed to improve their own and teams capability further. [8]. Also the team members are encouraged to present new ideas to the team. Team members also appreciate each other's help that they received during the iteration. Similar kind of meeting really helps to increase the team's energy level.

4. Conclusions

The techniques described in this paper can be adapted by organizations trying to implement agile testing. Still these are just methodologies and it can be modified based on the customer needs and the project requirements. The real challenge in implementing an agile testing process lies not only in the hands of the scrum master; it also heavily depends upon the management interest to adapt changes in the process level. The common areas that needs attention while introducing agile testing process are

- Maintaining Developers: Testers ratio.
- Selecting the Correct Automation tool for the entire project.
- Conducting two separate sprints for manual testing and the Automation testing.
- Creating a Metrics for agile testing.
- Improving testing team performance through constant feedback

5. Acknowledgments

I would like to express my appreciation to all my friends for their valuable feedback about agile testing and to my wife for her great patience and constant encouragement that she gave during this paper work.

6. References

- [1] M. Cohn. *Succeeding with Agile Software Development using Scrum*. Addison-Wesley, 2009.
- [2] L. Rising and N. S. Janoff. The Scrum Software Development Process for Small Teams. *IEEE Software*, Aug 2000 :26-32.
- [3] J. Watkins. *Testing IT: An Off-the-Shelf Software Testing Process*. Cambridge University Press, 2011.
- [4] P. R. Srivastava. Test Case Prioritization. *Journal of Theoretical And Applied Information Technology*, 2008 JATIT: 178-181.
- [5] I. G. Stamelos and P. Sfetsos: *Agile SW development QA*. Idea Group Inc (IGI), 2007.
- [6] J. C. Sanchez, L. Williams, and E. M. Maximilien. On the Sustained Use of a Test-Driven Development Practice at IBM. *AGILE 2007*, Aug 2007:5-14.
- [7] Takeuchi Hiroataka and Nonaka Ikujiro: The New Product Development Game. *Harvard Business Review*, 1986.
- [8] E. Woodward, S. Surdek and M. Ganis. *Practical guide to distributed Scrum*. IBM Press, 2010.