International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 10, ISSN No. 2455-2143, Pages 150-155
Published Online August - September 2016 in IJEAST (http://www.ijeast.com)

SURVEY OF TRADITIONAL V/S AGILE METHODOLOGY FROM THE PERSPECTIVE OF TESTING

Surabhi Jain
Department of Computer Science and Engineering
Krishna Institute of Engineering and Technology
Ghaziabad, India

Parita Jain
Department of Computer Science and Engineering
Krishna Institute of Engineering and Technology
Ghaziabad, India

***Abstract*-There are numbers of methodologies available that can be used for software development process and testing. The choice of the methodology depends on various factors such as the project type, resources, available cost etc. Mostly, the software development process requires periodic testing. Some methodologies require testing at very early stages of software development. These methodologies have some advantages but also have some disadvantages like project management, analysis of budget, schedule of project, customer interaction between team members and many more. Agile Testing is a technique which is time saving. This technique makes the work of testing team members more productive and easy. It produces efficient results. It mainly focuses on producing best amount of possible results with minimum amount of work. It helps us to meet the challenges and focus on obtaining the good result. The challenges faced by the agile team are analyzed and solved quickly. This paper discusses the challenges faced by a tester in Agile testing and the benefits of agile testing over traditional methodologies.**

Keywords-**Traditional methodologies, Agile methodologies, Agile Testing, Quality.**

I. INTRODUCTION

The development of the software includes different phases such as planning, analysis, designing and implementation. There are various models which have been created for Software Development Life Cycle (SDLC)such as waterfall, rapid prototyping, V-Model, spiral, incremental, Agile model but the waterfall model is mostly used in the traditional software development[8]. These models focus differently on the test effort required at different stages of development process. The new models like Agile model focus on the Test-driven approach of development. This model places an increased portion of testing on the software developing team before it reaches the final testing team. In the traditional models most of the testing is done when the requirements are gathered and the process of coding has been completed. The Test driven development ensures each feature of an application is strictly tested in the early stage of the life cycle of the project. The testing at early stages will solve many of the issues that are encountered by the development team and also help them to keep the track of the encountered defects[6].

A. Waterfall Software Development-

In the waterfall software development environment, the various phases are well-defined. Each phase have a deep review and process of authorization which is completed before it is moved forward. This is an important as it is logical and reasonable. First of all find out everything that would be needed. This process is followed by designing, coding, testing accordingly.

The main problem arises when there is requirement to make modifications and changes which were not required in the very early stages of development. This is the main drawback of this model. It is seen in the historical data that the change in a software development process is an inevitable part. So for the projects which do not require changes the waterfall model would be an appropriate choice[1].

Sometimes the upper management team blames the product team for not working thoroughly. The product team blames the developing team for the lack of order in the code. The Quality assurance team blames the management team for not having the specified tight deadlines and the cutoffs[7].

Any of the single department or team is not to be blamed because the nature of software is like this only. It always seems that it is the fault of other department. The real issue is not the persona but the principle. In other words, when the customer’s needs require changes then these changes are also incorporated in the software.

It is no wonder that the project results in the frustration. The deliveryof the software is made to the customer that lacks the important features.At the same time, it presents the fancy looking documents containing SRS, Architecture plans, Test plans. These documents do not provide any real value to the customers. The customers want efficiently working software that fulfils their requirements. They don’t need such fancy paper work[6].

B. Agile Emergence-

According to the problems mentioned above, development teams begin to shift to releases after short intervals. Firstly these iterative methods were done in small cycles. But they remained ahead of time with a fixed schedule. It had a fixed budget, a strict plan, and a predetermined set of features. So in place of one of the major release, several minor releases are scheduled.

Agile takes this concept on step ahead such that there are iterative releases while having no fixed plans. In this, each release adds some valuable bit which allows taking better decisions regarding the features that should be added next. Now the development proceeds in small cycles. This increased flexibility reduced the burden of more and more formal documentation. Now it is important to perform testing at each step in the process rather than only performing it at the end of the release[2].

II. MAJOR DIFFERENCES BETWEEN TRADITIONAL METHODOLOGY AND AGILE METHODOLOGY

- The software development process follows an incremental approach rather than a sequential approach. The software is developed in incremental cycles. It means that small test cases at every stage which are tested properly at that specific stage and faults are removed if they are identified before going to the next iteration.

- The communication between people like customers and software developers, testers etc is important rather than tools and processes. These communications allows the developer to be aware of the requirements of the customer and helps the tester to identify the discrepancies at each stage.

- The main focus is to develop the working software rather than forming a detailed documentation. The Agile methodology is based on face to face interaction and collaboration among the people that are working together. Because of the continuous communication between customers, software developers and testers there is no requirement of a detailed document[14].

- The feedback from the customers is required. The Agile methods are based on customer collaboration rather than contract negotiations. The Agile methods treat its customers as a part of their team. Whenever the developers have any questions or any confusion about the software the immediately clarify it with their customer.
- The change is an important state which is employed whenever it is required to deliver a quality project.
- There are many differences between Agile Testing and Traditional Testing. The major difference is that in Agile Testing both software developers and software testers work together. Both of the developing and testing teams takes the responsibility to deliver a quality software[12].

III. IMPORTANCE OF TESTING IN AGILE

- To succeed in agile, the testing in the software development process should be the central pillar. To proceed, the wasted time for quality assurance should be cut out from the last schedule; the delivery date should be move forward tossing out the expensive documentation. However, this is proved to be a failure.
- When executed appropriately, Agile is not at all disorganized. It is allowing the developers to run wild. But actually the truth is just opposite of this.
- According to the definition, the programmers create the unit tests before writing the code. It’s just that is a different type of order.
- Firstly, it focuses on coding rather than paper work. In the traditional type of project every team is responsible to provide quality. But in the agile projects, every team writes the tests. These teams include all the software developers, the business analysts, and also the customer. Here everyone provides various project advantages.
- It lowers the burden on a single project tester. This proves majorly important role because of the unrelated reasons the project tester may be replaced during the various release cycles.
- Secondly, the test awareness of the developers increases because the developers are also responsible for writing the test cases for each different feature so that they can quickly work as per the change in requirements.
- Finally, the testability is designed to test the specifications and the code. It means whenever it is required, the developers should be able to add features according to the specifications. The coordination is usually difficult and even neglected when the developer team and the tester team work separately[5].

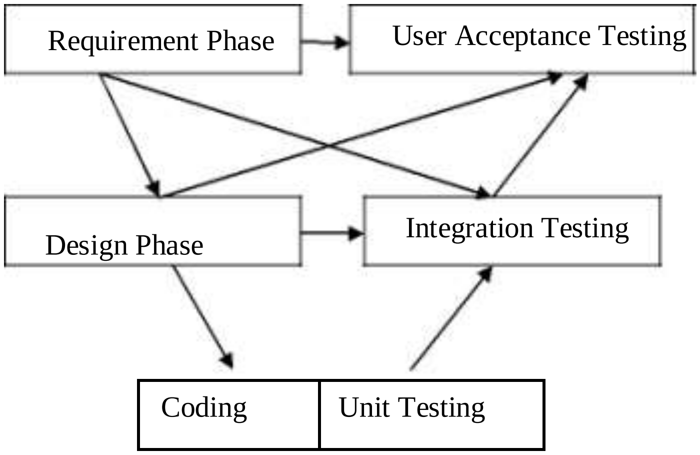

Fig. 1. Verification Vs Validation in Traditional Testing

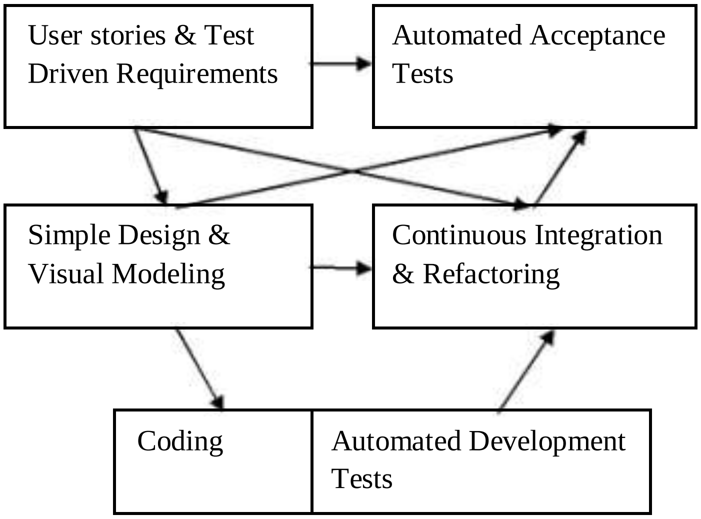
User stories & Test Driven Requirements
Automated Acceptance Tests
Simple Design & Visual Modeling
Continuous Integration & Refactoring
Coding
Automated Development Tests

Fig. 2. Verification Vs Validation in Agile Testing

IV. CHALLENGES IN TRADITIONAL METHODOLOGIES

- There are sufficient number of delays between when the software is being written and the feedback of the development.
- The defects that are found at later stages have significant effect on the software when changed.
- When the business requirements changes then it will affect the test cases that have already been developed.
- The quality of the software suffers and many quality assurance activities are left out in testing[6].

V. ADVANTAGES OF AGILE TESTING

- The feedback from the developers allows the testers to ask the required questions at the right time.
- The dependencies and testing challenges are identified at early stages.
- It accepts change as the healthy and requires part of the software development process.
- The collaborations of the team allow the members of the team to work for a single and collective goal.
- Quality is of utmost importance and the quality assurance is must[2].

VI. DESCRIPTION OF AGILEMETHODOLOGY USING TEST DRIVEN DEVELOPMENT

Test driven development is the main part of the Extreme Programming (XP). The Extreme programming uses a feedback approach such that it develop test cases before it develop code[3]. The reverse of the traditional development process is called as the Test driven development. The Test driven development begins with a set of unclear requirements and also on the feedback loop between the customers and the developers. The input requirements are provided by the customers. The customer is a part of the core team and provides the feedback whenever required. This method shows that requirements are essential during the development cycle of a project. The process of writing the test cases before functional code ensures that all the requirements are addressed. Also the required functionalities are involved. In the traditional development method, it requires a defined architectural design at the beginning of the development phase but it is not necessary in the Test driven method of development. The Test driven development focuses on solving the smaller problem first and the requirements become clear so that it canstart the software development cycle of the project[6].

Advantages of Test-Driven Development-

- Test driven development focus on low cohesion and high coupling in the code as the functionality is developed in very small steps. Each of the functionality should be self-sufficient to test the data in isolation.
- The test suite contains the documentation for the specifications such as functional specifications for the final system.

- The automated tests are used by the system such that it significantly minimizes the time taken to re-test the existing functionality for adding a new functionality to the system.
- If a test case fails then it is clear that some tasks should be performed to solve the problem. Clearly the success is achieved when the test cases no longer fails. This shows the confidence of the system that it efficiently meets the customers' requirements.
- Test-driven development helps in ensuring that source code is significantly unit tested. But it is still needed to take traditional testing techniques in consideration, such as functional testing, system integration testing and user acceptance testing. Most of these testing can be done at the early stages of the project. Also in Extreme Programming, the stakeholder(s) specifies the acceptance tests for the user's task.The specifications it provided either before or side by side when the code is being written. It provides stakeholders with the confidence that the system will meet their specifications and requirements[11].

VII. TRADITIONAL TESTING PROCESS

- Gather customer requirements and proceed for the review.
- Analysis of these requirements to develop test conditions and test cases.
- Write the test procedures.
- Execute the tests.
- Now re-execute some of the tests to fix the bugs or to add new functionalities[10].
- Reach the acceptable risk, the fixed deadline.
- Software is released.

VIII. AGILE TESTING PROCESS

- The customer gives the business requirements which are reviewed by the Engineering team. To plan the stages of development, the quality assurance team or the testing team is also involved in reviewing the customer requirements.
- The user stories are written by the engineering team and also perform the analysis of various issues at different stages like the designing and the implementation stage. These are reviewed and updated by the customer regularly. The Testing team followsup regularly at each stage till an integrated documentation is prepared. This ensures that the engineering team, the Testing team and the customer are at the same stage always. This further ensures the full test coverage.
- When the engineering team performs the implementation, the testing team performs the preparation of test cases and planning of test strategies[5]. All of this is documented and given to the engineering team and also to the customer. This helps in full test coverage and in removing the redundant test cases.
- The development team implements the code and the testing team tests the code simultaneously. This helps in identifying the defects at the early stages. The developer fixes these defects at that particular stage before moving forward for further development. This process is continued until the whole code is implemented. Now, the testing team emphasis on important test items like integration, usability and integration testing[9].

IX. AGILE TESTING APPROACHES

- Formally the development team and the testing team were not distinguished. The developers work in pairs. A developer can also play multiple roles such as a QA and the QA can work as a business analyst. One person writes the functionality of the module and the other develops the test cases[15].
- There is more communication and interaction between the members of the team as compared to the traditional method of development. The teams have a daily 30 minute small stand up meeting to help to resolve problems by asking the queries. The teams also have a weekly review meeting to track the progress on each specified iterative cycle.
- The development of the project started with no formal designed document. The specifications were same as user stories which were agreed by the team members. The team members do the planning about how to complete the task and how many iteration should be assigned for each task in the weekly review meeting.
- Every task is divided into several small tasks. Each of the stories and their related task are written on small cards which serve as the small design document foe the application block.
- There are no formal test plans. For feature development, the testing was mainly dependent on the tasks. The developing team gets the immediate feedback from the testing team. The testing team creates the quick starting samples for the development team to give a view of real life use of the application block.
- The quick starting samples are useful in code testing in the form of functional and integration testing. If there is any variation found at any stage, it is immediately

reported and fixed on the basis of case-by-case. The code that is modified is tested again so that it can be handed over for further processing.

- The unit testing is still usually done by the developers but there is more focus on automated testing at every unit level.
- There is more focus on the test driven development in the extreme programming. Extreme programming refers to the method of writing the test cases before writing the code[3].
- The integration testing and the system testing are done together. In an integrated environment, features can be tested as they are developed so that there is a daily build up and the continuous integration. According to the waterfall model, the testing is done by the professional testers at the end. But here each feature is to be tested as it is developed. Testing is not done after the completion of the project.
- At the end of every sprint when all the features are developed, tested and integrated, the regression testing is to be done on the software before releasing it. Regression testing is short as it is automated. The test driven development must not deviate from the expected results after testing.
- A stabilization sprint by an independent test team makes sure that everything must be okay before the release of the software. It can also be avoided if there is no need of it. But if it is required it should be of shorter duration and removing any defects before the launch of the software[15].

X. CONCLUSION

In this paper, the emergence and importance of agile, the various testing approaches for software development byboth traditional and agile methodologies are discussed. Furthermore, the criticism is followed by comparison of both traditional and agile methodologies. Further, the benefits are discussed along with the description of the test driven development. The traditional testing processes and the agile testing processes are also discussed followed by the agile testing approaches. There is a need for business to respond quickly and efficiently to provide the environment which is cost effective and focus on the usage of agile methods for developing the software. The agile methodology seems very dominant in the future. These are various aspects of software development process that can easily benefit from the agile testing approach. Also, the future scope is stated that an independent test team should be included at the end in the Agile model of software development to ensure the delivery of a highest quality software product to the customer.

XI. REFERENCE

[1] Georgios Papadopoulos “Moving from traditional to agile software development methodologies also on large, distributed projects” in *International Conference on Strategic Innovative Marketing, IC-SIM 2014, September 1-4, 2014,Madrid, Spain.*

[2] Shubhangi A. Deshpande, Anand A. Deshpande, Manisha V. Marathe and G.V. Garege, “Improving Software Quality with Agile Testing” in 2010 *International Journal of Computer Applications (0975 - 8887)Volume 1 – No. 2.*

[3] David Talby, Orit Hazzan, Yael Dubinsky and Arie Keren “*Agile Software Testing in a Large-Scale Project” in July/August 2006 IEEE SOFTWARE”.*

[4] Chhavi Malhotra and Anuradha Chug “Agile Testing in Scrum-A Survey” in Volume 3, Issue 3, March 2013 ISSN: 2277 128X*International Journal of Advanced Research in Computer Science and Software Engineering.*

[5] Allen Parrish, Randy Smith, David Hale, and Joanne Hale “A Field Study of Developer Pairs: Productivity Impacts and Implications” in IEEE SOFTWAREPublished by the *IEEE Computer Society 0740-7459/04/$20.00©2004IEEE.*

[6] Balaji Sundramurthy, Ronald S. Cordova and M. Sundara Rajan “Traditional Vs Agile Methodology: An Analysis on Challenges faced in Testing Perspective” in *Intl. Conf. on Advances In Computing, Electronics and Electrical Technology - CEET 2014.*

[7] Ming Huo, June Verner, Liming Zhu and Muhammad Ali Babar “Software Quality and Agile Methods” in *28th Annual International Computer Software and Applications Conference (COMPSAC’04).*

[8] Marian STOICA, Marinela MIRCEA, Bogdan “Software Development: Agile vs. Traditional” in*Informatica Economică vol. 17, no. 4/2013.*

[9] Kaushal Pathak and Anju Saha “Review of Agile Software Development Methodologies” in Volume 3, Issue 2, February 2013 ISSN: 2277 128X *International Journal of Advanced Research in Computer Science and Software Engineering.*

[10] P.B.Selvapriya “Different Software Testing Strategies and Techniques” in International Journal of Science and Modern *Engineering (IJISME) ISSN: 2319-6386, Volume-2, Issue-1, December 2013.*

[11] Shaweta Kumar and Sanjeev Bansal "Comparative Study of Test Driven Development with Traditional Techniques" in International Journal of Soft

Computing and Engineering *(IJSCE) ISSN: 2231-2307, Volume-3, Issue-1, March 2013.*

[12] Nils Brede Moe, VenuGopal Balijepally, Sridhar Nerur and Torgeir Dingsøyr "A decade of agile methodologies: Towards explaining agile software development" in *The Journal of Systems and Software 85 (2012) 1213– 1221.*

[13] Mario Špundak "Mixed agile/traditional project management methodology – reality or illusion?" in

Procedia - Social and Behavioral Sciences 119 (2014) 939 – 948 ScienceDirect27th IPMA World Congress.

[14] Maarit Laanti, Outi Salo and Pekka Abrahamsson "Agile methods rapidly replacing traditional methods at

Nokia: A survey of opinions on agile transformation" in *ScienceDirect Information and Software Technology 53 (2011) 276–290.*

[15] Anil Agarwal, NK Garg and Avirag Jain "Quality Assurance for Product Development using Agile" in

2014 International Conference on Reliability, Optimization and Information Technology-ICROIT 2014, India, Feb 6-8 2014.