Abstract

There are various methodologies available for developing and testing software. The method we choose depends on various factors such as the nature of project, the project schedule, and resource availability. Although all software development projects involve periodic testing, some methodologies focus on getting the input from testing early in the cycle rather than waiting for input when a working model of the system is ready. Those methodologies that require early test inclusion have couple of great conditions, furthermore incorporate tradeoffs terms of project management, schedule, client interaction, budget, and communication among team members. Agile Testing is an arrangement of timesaving techniques specifically designed to make the work of agile testing teams easier and more productive. It is an engaging procedure that produces great results and has a simple mission: Get the most ideal testing comes about with minimal measure of work an exact, targeted solution. Common challenges agile teams face and recommended solutions to handle them rapidly and effectively. These challenge and solution-based techniques do not require significant changes in your current work process. We can adopt them in increments, which empowers us to focus on one particular test and meet it head-on with a precise, targeted solution. Common challenges agile teams face and recommended solutions to handle them rapidly and effectively. This paper discusses about how testing fits and challenges into conventional/Agile approach and then discusses the test-driven development practice in Agile Methodology in detail.
I. INTRODUCTION

A Software Development Life Cycle (SDLC) adheres to important stages that are essential for developers, such as planning, analysis, design, and implementation, there are various programming improvement life cycle (SDLC) models have been made: waterfall, spiral, V-Model, rapid prototyping, incremental, Agile model. Different software development programming advancement models will focus the test effort at different points in the development procedure. Newer development models, for example Agile, utilizes test driven development and place an increased portion of the testing in the hands of the developer, before it reaches a formal group of analyzers. In a more traditional model, the greater part of the test execution occurs after the requirements have been characterized and the coding procedure has been completed. By using test-driven advancement, we can ensure that each component of an application piece is thoroughly tried ahead of schedule in the Software project life cycle. Early testing fundamentally would solve majority of the issues that will be encountered by the development team and it will empower them to monitor every one of the defects.

1.1 KEY DIFFERENCES BETWEEN TRADITIONAL AND AGILE METHODOLOGY

- Development is incremental instead of sequential. Software system is developed in incremental, rapid cycles. This results in small, incremental releases, with every release building on previous functionality. Each release is completely tested, which ensures that all issues are addressed within the next iteration.

- People and interactions are emphasized, instead of processes and tools. Customers, developers, and testers constantly interact with one another. This interaction ensures that the tester is conscious of the requirements for the features being developed throughout a selected iteration and can easily identify any discrepancy between the system and the requirements.

- Working package is the priority rather than elaborate documentation. Agile methodologies rely on face to-face communication and collaboration, with people operating in pairs. Because of the intensive communication with customers and among team members, the project does not need a detailed requirements document.

- Customer collaboration is employed, rather than contract negotiation. All agile projects include clients as a part of the team. When developers have questions on a demand, they immediately get clarification from customers.

- Responding to change is finished, rather than intensive planning. However, it suggests changing the plan to accommodate any changes in assumptions for the plan, instead of attempting to follow the first set up.

- Agile testing was totally different in many ways from traditional software testing. The biggest difference is that on an agile project, the complete development team takes responsibility for quality. This implies the full team is responsible for all software testing tasks, including acceptance test automation. When software testers
and programmers work together, the approaches to check automation will be inventive.

1.2 CHALLENGES IN TRADITIONAL METHODOLOGY

- Significant delays between once software package is written and development receives feedback
- Defects found late within this process can have major implications when changed
- Changing business needs have an effect on check cases that have already been developed.
- Communications produce risk that different groups may have different expectations of the ultimate product
- Quality suffers and lots of QA activities get ignored once testing is the last activity before a fixed release date.

II. TESTING METHODOLOGIES

At the early years of software code development, most of the users’ requirements were fairly stable, and development followed the plans without major changes. However, as software development concerned additional essential and dynamic industrial projects, new difficulties emerged in keeping with the expansion of companies. These difficulties include:

2.1 Evolving requirements:
Client requirements are dynamical due to evolving business needs or legislative problems. Most of the customers don’t have a transparent vision regarding the specifications of their requirements at the early stages. Some customers understand what their true requirements are only when they use an application that doesn’t really meet their needs. Another source of change comes from experiences gained throughout the development.

2.2 Customer involvement:
Lack of client involvement results in higher possibilities of project failure. Many companies usually don’t assign any effort for client involvement.

2.3 Deadlines and budgets:
Usually, customers don’t accept failure. On the opposite hand, companies usually provide low budgets, tight deadlines, whereas at constant time, requiring high demands, and every one of this can be owing to competition within the markets.

2.4 Miscommunications:
One cause for the misunderstanding of requirements is that the miscommunication between developers and customers. As an example, every party uses its own jargon, and this results in misunderstanding of customer’s needs.

With the existence of such issues, the OO software system development methodologies cannot satisfy the objectives of software development companies. New development methodologies have to be applied so as to befit these issues. A number of IT professionals started to began to work individually on new approaches to develop software. The results of their researches were a set of new development methodologies that have several common features. When they met in conference in, they created the so called: Agile Manifesto.
These approaches were developed based on the same rule that the most effective way to verify a system is to deliver working versions to the clients, then update it according to their notes. Agile authors designed their methodologies on four principles: First, the main objective is to develop software that satisfies the clients, through continuous delivering of working software code, and obtaining feedback from customers concerning it. The second principle is accepting changes in requirements at any development stage, so that clients would feel more comfortable with the development process. The third principle is that the cooperation between the developers and the clients (business people) on a daily basis throughout the project development. The last principle is that developing on a test-driven basis; that’s to write test prior to writing code. A test suite is run on the application when any code change. Agility briefly suggests that to strip away the maximum amount of the heaviness, commonly associated with traditional software development methodologies, as potential, so as to promote quick response to dynamical environments, changes in clients requirements, accelerate project deadlines, and the like. Agile methodologies prefer software code development over documentation. Their philosophy is to deliver several working versions of the software package briefly iterations, then update the software according to customers feedback. Applying this philosophy will help to beat the issues mentioned earlier, by welcoming changes, satisfying user requirements, quicker development, and at the end, users will have just the system they need.

### III. BENEFITS OF AGILE TESTING

- On-going feedback to developers permits testers to ask the right questions at the right time.
- Early identification of dependent, technical or testing challenges and roadblocks.
- Embraces modification as a healthy and real part of software development.
- Team collaboration helps everybody work together toward a common goal.
- Quality comes initial as a result of final acceptance criteria are established before the work starting.

### IV. KEY CHALLENGES FOR TESTER IN AGILE PROJECT

#### 4.1 Scrum Master

A Scrum Master is a team leader and assistant who help the team members to follow agile practices so they will meet their commitments. The responsibilities of a scrum master are as follows: To enable close co-operation between all roles and functions, To take away any blocks, To shield the team from any disturbances, To work with the organization to trace the progress and processes of the corporate. To make sure that Agile Inspect & Adapt processes are leveraged properly which includes: Daily stand-ups, Planned conferences, Demo, Review, Retrospective Meetings, and To facilitate team meetings and decision-making method.

#### 4.2 Product Owner

A Product Owner is the one who drives the product from business perspective. The responsibilities or a Product Owner are as follows: To outline the requirements and
prioritize their values, To confirm the release date and contents, To take an active role in iteration planning and release planning meetings, To make sure that team is functioning on the most valued requirement, To represent the voice of the client, To accept the user stories that meet the definition of done and defined acceptance criteria.

4.3 Cross-functional Team
Every agile team ought to be a self-sufficient team with 5 to 9 team members and an average expertise starting from of 6 to 10 years. Typically, an agile team includes of 3 to 4 developers, 1 tester, 1 technical lead, 1 product owner and 1 scrum master. Product Owner and Scrum master are considered to be a part of Team Interface, whereas alternative members are part of Technical Interface.

V. PROCESSES FOLLOWED IN TRADITIONAL TESTING
Firstly receive requirements document from the client then proceed to review then eventually can get requirements document that's considered base lined or signed-off. After this Analyses these requirements to create test conditions and test cases then Write test procedures and wait for a bit of software to miraculously appear in test environment. Then now start executing tests. Now begin re-executing a number of these tests as you now start iterating through new builds that are released to fix bugs or they may even include new functionality. Then reach the appropriate risk, enough testing point (or the fastened stabile point) and the software is released

Table -1: Agile Development Manifesto

<table>
<thead>
<tr>
<th>Individuals And Interactions</th>
<th>Processes And Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Software Product</td>
<td>Comprehensive Documentation</td>
</tr>
<tr>
<td>Customer Collaboration</td>
<td>Contract Negotiation</td>
</tr>
<tr>
<td>Responding To Change</td>
<td>Following A Plan</td>
</tr>
</tbody>
</table>

According to agile manifesto shown in table, agile software development approach permits requirements changes and it means that changes is created conjointly in check cases. That method needs higher communication between developers, testers and finish users so as to beat the issues and build a lot of versatile and optimized solutions. The individual people and their interactions are the most necessary part in the project, i.e. the collaboration between the team members is responsible for higher learning atmosphere where new members may learn things from the senior and most experienced team members. Since agile development needs multifunctional groups that follow the principles of iterative and incremental developing practices, the testing method ought to be efficient and it needs creating tests early and often. There should be clear definition what the results of testing ought to be at the end of each sprint. It means that tests have to be done before the implementation of the project functionalities in every sprint.

The key issue of successful testing is close collaboration between end users, developers and testers. Testers should be a part of the development team and every activities must be parallelized as much as possible. As an example, while testers are working on test cases,
developers are coding the user stories. Because the agile development paradigm says that every process should be done fast, there's no such a lot time for testing, so the test data preparation should be done in the planning phase in the development process. Better approach is these take a look at information to be ready at the side of the stakeholders so as to satisfy the functional and performance requirements.

- **How an Agile Team Plans its Work?**

  An Agile team works in iterations to deliver user stories where every iteration is of 10 to 15 days. Each user story is planned supported its backlog prioritization and size. The team uses its capability – how many hours are available with team to work on tasks – to make a decision how much scope they have to plan.

### VI. CONCLUSION

In this paper, we described the various testing approaches to software system development through traditional and agile methodologies. Furthermore, we initially criticized on both traditional and agile methodologies followed by the comparison. Agile methodologies came into existence after the requirement for a lightweight to do software code development so as to accommodate changing requirements environment. Agile methodologies give some practices that facilitate communication between the developer and therefore the client, and undergo develop-deliver-feedback cycles, to possess additional specific view of the requirements, and be prepared for any amendment at any time. The most aim of agile methodologies is to deliver what’s needed when it is needed. Further, we discussed on benefits, analysis on take a look at driven development & tester challenges in both traditional and agile methodology. The requirement for business to reply rapidly to the environment in an innovative, cost effective and economical means is compelling the utilization of agile methods to developing software system. The future of agile methodologies appears terribly dominant. In general, there are some aspects of software development project that will benefit from an agile testing approach and others can benefit from a more predictive traditional testing approach.

### VII. REFERENCES


To Cite This Article