A Survey Paper on Agile Testing

Archana Tomar¹, Priusha Narwariya²

¹²Asst. Prof., Department of Computer Science & Engineering, ITM Group of Institutions, Gwalior, India

Abstract- The trend in software development phase is agile methodology which enables and assure early to market competition between different products. In this paper we will see what agile methodology, challenges faced during testing are and how we can difference from other testing.

Index Terms- Software Development Life Cycle (SDLC), agile process, Agile Testing.

I. INTRODUCTION

In software development, agile refers to any move toward to project management that strives to principles of collaboration, flexibility, simplicity, transparency, and responsiveness to feedback throughout the total method of developing a replacement project. And agile testing usually suggests that the follow of testing software package for bugs or performance problems inside the context of associate degree agile work flow.

In the Agile process, testing and development are the two faces of software product of a coin. In the view of Agile, well-organized construction is severely constrained if developers are motivated to improve their code to a state of accuracy before passing it on to a testing team, who then strive to split it in as many ways as they can before forward their report back to the development team.

In these steps are require more time, cost and leads to internal division between developer and tester. Instead, in the terms of time and process, the Agile approaches that these two essential functions be merged. Thus bridging the illusory divide between developer and code breakers, and even minimizing the requirement for robust testing teams.

II. AGILE METHODOLOGY TYPES

1) Scrum

For completing complex projects Scrum is used in an agile framework. Scrum originally was formalized for software development projects, but it works well for large and complicated projects.

Scrum takes a highly iterative approach that focuses on defining key features and objectives prior to each sprint. It is designed to reduce risk while providing value quickly.

Scrum begins with requirements that focus how attributes should perform and tested. The team then cycle through a number of sprints to give tiny bursts of value speedily.

Scrum answered of questions from the very start for help the team work in this flexible way and avoids shifting priorities

2) Kanban

It is an agile process framework. Kanban is a technique for organize the construction of products with an importance on repetitive delivery while not overloaded the development team. The process is designed to help teams work in a group more effectively like Scrum and Kanban.

III. AGILE TESTING METHODS

1) Behavior Driven Development (BDD)

TDD extension is Behavior Driven testing. In BDD we also write tests first and then the append code. The main difference that are

- Test cases are written in simple descriptive English type grammar
- Test cases are explained as performance of software application.
- For clarify requirements examples are used

Features of BDD

- For thinking in “behavior” to Shifting from thinking in “tests”
- Cooperation work in stakeholders, Analysts and developers.
- For easy description ubiquitary language is used
- Determined by organizational value
- by utilizing natural language Test Driven Development (TDD) is Extended that non technical stakeholders can understand
- BDD can utilize for Unit test cases and it is very popular. for Unit Testing following BDD approach tools like RSpec or in .NET something like MSpec or SpecUnit is popular. On the other hand, we write BDD-style specifications regarding UI relations.

2) Acceptance Test Driven Development (ATDD)

Acknowledgment Test Driven Development (ATDD) is a training in which the entire group cooperatively discusses acknowledgment criteria, with cases, and after that distills them into an arrangement of solid acknowledgment tests before development starts.
It's the most ideal way I know to guarantee that we as a whole have the same shared comprehension of what it is we're really constructing. It's additionally the most ideal way I know to guarantee we have a shared definition of done.

Clearly I think this is an imperative agile development practice. It's one of the main part of my Agile Testing class. However some way or another I have neglected to expound on it much on this blog. Time to redress that.

3) Exploratory Testing

"Exploratory testing" – as the name proposes, is an approach to test the application by investigating it to discover what does the application do, its highlights, what it doesn't do and so on.

Exploratory testing requires no or insignificant arranging. Analyzers consistently settle on choice on his/her subsequent stage of activity. We can state that in exploratory testing, test arranging, examination, outline and test execution, are altogether done together and in a split second. It totally relies on the analyzers point of view. We as a whole do a type of an exploratory testing in our everyday life.

From time to time this testing can be more valuable than the formal testing approach and can discover decent deformities which some of the time go concealed amid formal testing. It is additionally called Adhoc testing.

In all cases, exploratory testing takes after four key standards:
1. Test Design, Parallel test arranging, and test execution.
2. Specific yet adaptable
3. Aligned toward examination of potential Chances
4. Knowledge sharing

4) Session Based Testing

Session-based testing (SBT) is another framework developed by James and Jonathan Bach for doing exploratory testing. Like visit testing it brings an organized approach, nonetheless the time based instead of context based. In the event that visit testing resembles going on a voyage through another city, session-based testing resembles free diving. You jump into the water holding your breath and can investigate just such a large amount of the ocean bottom before you have to surface again to take in some air.

The undeniable advantage of this purposeful imperative in SBT is that you compel yourself to focus on the job needing to be done and maintain a strategic distance from a wide range of waffling in your journey to discover goods (or, for our situation, programming bugs).

SBT additionally makes it conceivable to deliver important measurements, which can be introduced to senior directors and help the entire group to address any blemishes in the item and the QA procedure.

IV. Testing in Agile Methodology

Agile strategy take after test driven approach, upgrades are made in every cycle and are tested, products are adjusted and refreshed to meet the requirements. Testing in Agile Methodology for every dash.
Agile Methodology Challenges and Solutions

In the Testing Difficulties which are by and large faced for the duration of testing in agile methodology:

1. Frameworks are still under developments.
2. Framework prerequisites, framework under test and test situations may change much of the time.
3. Testing is done rigoursly and over and over, and regression is its greatest part. Brief span is provided for testing, it ought to be fast and proficient.
4. Testing and development stages are short and covering so sync between both is required.

Inaccessible/Underdevelopment System: Stubs/Mocks will be made to imitate conduct of unavailable or over development systems.

• WebUI:
  Selenium will be utilized to record procedures on webUI and those scripts will be sent out to TestNG to make integration test cases and suites.

• Data conditions:
  Data dependencies problems with test cases will be recuperated by isolating test data from the test cases, test data will be put away in different sheets so it can be changed without affecting test cases.

• Environment conditions:
  Condition/target URL may change after emphases and has major impact on test case implementation, it can be isolated from target test cases by putting away and retrieving it from outer files.

• Integration Testing:
  Reconciliation testing will be finished utilizing TestNG, test cases and test suites will be made for every situation furthermore, there will be a controller to execute those test cases, controller will have the capacity to summon test cases identified with different framework.

• Regression Suite:
  Regression test suites will be made after every cycle/run and will be executed in straightaway emphases.

• Tracking changes in the framework and necessities:
  We can make change following framework to track changes in UI, application, necessities and its effect on the test cases.

Comparison between SDLC Approaches

<table>
<thead>
<tr>
<th>SDLC approach</th>
<th>Waterfall Model</th>
<th>V Model</th>
<th>Agile Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Waterfall is discharge driven, with a characterized basic way and sequence for delivery</td>
<td>1. It takes verification and validation approach</td>
<td>1. agile depends on short iterative delivery cycles typically 2 a month in term</td>
<td></td>
</tr>
<tr>
<td>2. Assessments depend on the work required to meet the necessities</td>
<td>2. Phase containment</td>
<td>2. Assessments are done in view of the measure of work the group can achieve in a set period of time</td>
<td></td>
</tr>
<tr>
<td>3. Requires obviously characterized necessities forthright</td>
<td>3. The testing gets &quot;squished&quot; on the grounds that coding takes longer than expected, and on the grounds that groups get into a codeand-fix cycle toward the end.</td>
<td>3. Prerequisites are relied upon to develop and changes are grasped</td>
<td></td>
</tr>
<tr>
<td>4. Achievement is estimated by the IT association</td>
<td>4. It is point of view development.into a codeand-fix cycle at the end.</td>
<td>4. Achievement is estimated by business value delivered</td>
<td></td>
</tr>
<tr>
<td>5. It is perspective advancement</td>
<td>5. It is perspective development.</td>
<td>5. Group can see and feel improvement</td>
<td></td>
</tr>
</tbody>
</table>

V. Future Scope

Although Agile has officially made noteworthy advances into the software development lifecycle, there’s as yet far to go, particularly among testing team.

Going ahead, more across the board selection and more prominent development of Agile procedures will expect analyzers to go past test creation and execution and start to center around code delivery and integration as well.
In the meantime, analyzers should sharpen their automation abilities, turn out to be more engaged with the whole software development process and keep on developing a community oriented association with designers. At last, these progressions will likewise require testers to wind up specialists at development and product usage in order to give more all encompassing testing systems and take against the part of "quality champions."

Later on, three key fundamentals will turn out to be especially essential for testers working in Agile situations:

1) **Communication**

   Agile requires a highly collaboration amongst testing and development field, and that cooperation makes correspondence a best need for testers. Also, in our current reality where quality turns into everyone responsibility, testers will progress toward becoming “quality champions” that fill in as internal specialists, which will put their capacity to plainly convey testing needs and thinking under the spotlight.

2) **Skill Diversity**

   In an Agile situation, everything can change on a dime, and that expects testers to be versatile. Some portion of this flexibility is having a different range of abilities with the goal that testers can change course as required. For example, useful testers need to extend their abilities past manual scripted execution. This various range of abilities will be an absolute necessity as various dashes require distinctive kinds of testing to be executed in a short time span.

3) **Business Mindset**

   Finally, Agile is totally based on customer centric approach which will ensure customers receive as much value, quickly and as early as possible. For delivering this value Testers perform a big role, but it needs them to take on a business mindset therefore that they can identify with customer desires, wants and concerns and build up their testing techniques strategies accordingly.

VI. **Conclusion**

Agile method is one of the trending software development techniques.

- Testing in agile process faces a few difficulties and risks.
- Risks and difficulties can be settled utilizing test automation technique.
- Success rate in agile technique is extremely higher if contrasted and other software development approaches e.g. waterfall model.
- Value created from the product or module is obvious from early test cycles.
- Risks decrease radically with the cycles.

**REFERENCES**