Analysis of Agile testing attributes for faster time to Market: Context of Manufacturing sector related IT projects

Tulika Chakravortya, Samyadip Chakrabortyb *, Nasina Jigeeshc

aAccenture (India) Services Pvt. Ltd., Hyderabad
bIBS Hyderabad, IFHE University, Dontanpalli, Shankerpalli Road, Hyderabad-501203, A.P
cIBS Hyderabad, IFHE University, Dontanpalli, Shankerpalli Road, Hyderabad-501203, A.P

Abstract

Agile methodologies are helpful to develop and implement systems to meet the market demands in a faster way, by helping various organizations to evaluate projects. Agile testing strategy is one of the methodologies that is suitable for implementation of IT projects. This paper focuses on IT projects related to manufacturing sector by deriving the significance of various agile project testing attributes across every phase of information systems development life cycle (SDLC) for faster time to market the products. The attributes include prioritization of features according to customer requirement, early customer involvement, incremental and iterative sprints, flexible development environment with less documentation and complexity management. The results highlight the varying significance of these attributes and identifying the key attributes in each project phase; thereby helping software professionals and project managers by providing a handy framework for focusing on phase-wise core attributes that might be essential for tackling related project issues and fulfilling the managerial priorities.

© 2014 Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).
Selection and/or peer-review under responsibility of Symbiosis Institute of Management Studies.

Keywords: Agile methodology; Testing strategy, SDLC; IT projects; Manufacturing.
1. Introduction

Agile Methodology came into existence in the late 1990's as an iterative form of software development. Software industry has over the decades become the backbone of modern day business world, which is more and more becoming customer-centric and consequently becoming highly dynamic. Referring to one of the Gartner reports conducted recently, the amount of software testing work which is being outsourced to India is now approx $4.7 billion with a growth rate of 15% annually. In this backdrop, software industry and software quality assurance gains tremendous relevance. Strategy is crucial for every organisation’s growth and development. Proper quality planning and suitable software testing strategies are the foremost requirements in modern software development and delivery of bug free softwares as per the client’s functional, technical, operational, and maintenance expectations. Moreover this industry sector being highly revenue-driven, cost containment, prioritisation of managerial objectives and focussing on key test strategy attributes become highly crucial for ensuring timely and accurate delivery.

However the industry and its client requirements are very dynamic and a major issue of software development and testing is the constantly changing customer requirements which lead to most other adverse implications. The entire project cost, time & effort gets unbalanced when customer changes the scope and requirement during the phases of the project and causes enormous amounts of rework for the software developers. However owing to the dynamic nature of the sector and it’s client requirements, managerial priorities play pivotal role in the choice of attributes and project cost implications. The factor behind this variation and set back being the lack of synchronization between the customers, managers, software developers and testers resulting in rework at each and every phase of SDLC (Systems/Software development Life Cycle). The dynamicity of the sector intensifies the need for clarity and focus on the managerial priorities and choice of supportive testing strategy attributes.

In today’s manufacturing sectors intense competitive pressure necessitates faster time to market of their new products and consequently the IT multinationals which extend their service to such high competitive global players have to extend their services at matching pace. Thus faster rolling out of the software modules with strict adherence to the time and cost schedules becomes highly challenging. In this contextual backdrop, practices like agile methodology has become the buzzword and highly sought after. In a survey conducted among some of the top software companies, just 6 years ago, it was indicated that only 5% of the software projects were developed with Agile Technique. However, at present 35-40% of the projects follows the agile technique. This is indicative of the fast pace at which this agile approach is catching up with the traditional practices in the sector. Adoption of agile systems development is increasing in every industry (Schwaber and Fichera, 2005). Manufacturing sector has also been aggressively developing suitable information systems to streamline their activities like reduction of cycle times and speed up the delivery time.

Quality planning and management plays a vital role in dynamic software industry and adoption of appropriate testing
strategy and choice of correct software testing methodology become very crucial in today’s software delivery and quality management. While discussing about application of agile methods in an environment that may be more familiar with high ceremony project management methods, Alleman (2002) stated that agile project management methodologies are useful to develop, deploy, or acquire information technology systems in modern organizations. Even though agile methodology has been in existence for over two decades, and has been in practice in the IT domains for nearly a decade, still it lacks clarity in terms of understanding of the professionals regarding its suitability (in which type of projects and in which phase which attributes are most vital) and applicability in software testing and quality management. There happens to be no clear cut understandable framework or empirical study in the software quality assurance domain that might help the managers in identifying the most prominent key attributes of the Agile testing methodology across the five SDLC phases inline with their business priorities and project objectives. It is becoming imperative to have in-depth comprehensive understanding of specific usage scenarios and their suitability to specific software project categories and SDLC stages. Appropriate agile testing strategies and study of agile testing methodology can mitigate the issue of rework in IT projects. This offers a niche area that necessitates deeper probing. Thus this study attempts at empirically analyzing the key Agile testing attributes for faster time to Market in the niche context of manufacturing sector related IT projects. The study discusses the SDLC phases of implementation in details in the following section followed by a contemporary literature review, methodology and sampling, results, analysis and discussion and finally conclusion in separate sections.

2. Literature Review and Supportive Rationale: Building the mesh

Achieving Software Quality is a multidimensional approach comprising of functional, structural and process orientations (Chappell and Associates, 2012). Without a proper process oriented testing the right quality of software can never be obtained. Studies have shown that without testing, the value of the software developed can never be evaluated. Testing is not just checking for defects but rather, it is the measures taken for defect prevention in each and every phase of SDLC. Testing is becoming complex and without proper planning projects tend to fail. With low project funding the testing cost is usually crashed which leads to buggy software (Rao et al, 2011).

Due to project cost crashing the testing teams is always introduced late in the testing phase, testing cycles are shortened and resources are reduced causing neglected testing. Various testing activity in a project like test planning, developing test requirements, developing initial test-cases and traceability-matrix, generation of functional test-cases, etc cannot be covered only during the test execution phase, thus testers involvement is needed from the starting of the project. In software projects it is always observed that the requirements are changed every now and then based on changing client requirement specificities (Gross, 2005). Thus the major issues that come up are ‘Scope Creep’ which is the uncontrolled and unexpected changes in user expectations and requirements as a project progresses and ‘Feature Creep’ which is the uncontrolled addition of features to a system with a wrong assumption
that one small feature will add nothing to cost or schedule (Savolainen et al., 2012; Fichter, 2003). These are the major causes of project failure or rework. It can be rationally argued and understood that actually, most of the estimates of time and cost are done at the beginning of the software development during early planning phase much before the actual development starts and hence when problems are identified and requirements change all the plans and estimates fail leading to loss of revenue. In Today’s market condition when technology and scope are continuously changing, market competition is high and customers have their own outlook to go for any product and choices. “Staying in business involves not only getting software out and reducing defects but tracking continually moving user and marketplace demands” (Cockburn, 2002). Under the given circumstances, thus the traditional SDLC approach, where the requirements are analyzed and fixed only once in the ‘Requirement Analysis’ phase and gone ahead with the implementation of the planned features, opens up risky propositions necessitating agile approaches aimed at faster timing to market of quality customer specific software.

While discussing the competing models, the traditional V-Model is effective in many cases as it includes formal software project management, requirements analysis, architecture, design, computer programming, and testing (Beck, 1999), however its implementation at times leads to several rework issues in dynamically requirement changing projects. Changes are hard to be introduced in mid release as all the planning is done in the plan phase itself leading to massive rework if changes are introduced in later phases (Berki et al, 2006, Siakas and Siakas, 2009). Due to all these problems most of the software developed completely fail due to unreliability in design, mismatch of build and requirements. This creates the necessary ambience for most software project managers to conceptualize and adapt a much adaptive and iterative agile approach since nowadays fast and in-time delivery of software to the market is the foremost need and major challenge of software industry is inability to deliver software within the expected time, cost and within requirements (Boehm, 1981).

In the early 1990s the Scrum Methodology was first used for software projects delivery which brought spectacular difference from the traditional approach resulting in new era of scrum based methodology, which dramatically enhanced software projects’ success rate. In Agile methodology the entire feature is delivered in small 3-4 weeks cycles and can capture the market much ahead of its competitors. ‘Agile’ means responsive and flexible. Agile approach to software engineering are characterized by early customer involvement, flexible development processes, and iterative product releases (Beck, 1999; Highsmith, 2002; MacCormack et al., 2001). Due to the flexible Development process of Agile Model they can readily respond to change and uncertain market condition (Highsmith, 2002).

Chow and Cao (2008) studied the critical success factors of Agile Software Projects taking four categories as: Quality, Scope, Time, and Cost and identified that in order to achieve a successful project the critical factors are: Delivery Strategy, Agile Software Engineering Techniques, and Team Capability. Liu et al. (2000) stated that
executives and managers of Fortune 1,000 firms representing more than 16 major industries may benefit from the knowledge that agile methods may be linked to better software project performance.

Agile concentrates on only the functional aspect and working of software but technical processes are missing. All the phases are not well defined and well documented. The phases are not separated with proper sign-off documents and deliverables are rather informal. Due to its short sprint time developer cannot document everything and later it is difficult to trace back by others working on it. Cohn and Ford (2003) studied the problems in transitioning organizations to agile processes while Larman (2004) discusses in detail mistakes and misunderstandings happened in agile projects. A research by Boehm and Turner (2005) emphasizes on management challenges in implementing agile projects, whereas a study by Nerur et al. (2005) covers problems not only in management aspect but also in people, process, and technology dimensions of migrating to agile projects. Stettina and Heijstek (2011) conducted a quantitative survey of five agile factors- Shared Leadership, Team Orientation, Redundancy, Learning and Autonomy and found that ‘the organizational and individual levels of autonomy and redundancy are the dimensions with the lowest scores’. David F.Rico (2004) studied the links between better software project performance & early customer involvement, flexible development processes, and iterative product releases which are the aspects of Agile Methodology.

Studies on distributed Agile methodology indicates lot of risks and challenges in terms of information sharing challenges and highlights high importance on efficient information sharing. (Kontio et.al. 2004). In small-scale distributed agile software development projects, volatile requirements and inefficient communication can cause severe problems (Korkala and Abrahamsson, 2007). Agile methodologies preach the process of face-to-face communication, as it is not a heavy documentation centric approach. Pure face to face communication is not possible in distributed agile process. So this challenge can be handled by developing a system for the task based capture and proactive distribution of recurrent information(Holz and Maurer, 2003).

Understanding of SDLC phases is pre-requisite in the current study context. SDLC or software development lifecycle is broadly segregated over 6 vital phases: plan, analyze, design, build, test-execution and deployment. Out of these the first five stages are essentially the stages where the concept of software testing and software quality assurance, not only becomes a priority but becomes imperative.

*Plan phase* is the initial phase of the project where all the required planning and conceptualizations, related strategies and project outline are prepared; during *analysis phase* the requirements are analysed and high level strategies and estimates are prepared; in the *design phase* the designing of the process and sub-processes are done, whereas in the *build phase* the software is built, code is developed and functionalities are created. The 5th phase is
the test Execution phase, when the software is tested and quality is analysed, defects (if detected) are rectified and perfection is brought into the software for accurate performance as per the requirement. The final stage of SDLC is the Deployment phase. In this final phase, the developed and tested software is deployed on the customer side. These phases are irrespective of the types of software projects i.e. present across major categories of software projects: Development Project, Maintenance Project as well as Up-gradation or Migration Project (Others). All these major categories are considered in the current study.

The development projects are those projects that are started from the scratch based on detailed customer provided requirements set and are often afflicted with operational and managerial issues like lack of proper scope definition of the software, lack of goal alignment, improper and inaccurate estimates, poor communication among the developers, testers and customers, lack of understanding about project complexity, etc. Maintenance projects are those software projects which are already developed and further modifications are planned for betterment, correction of existing faults or enhancement of capabilities to the software which already exist, deletion of some obsolete modules and further optimization of the software. However uncontrolled changes and lack of knowledge about the team pose important threats for such projects. Up-gradation or migration projects referred to as others in the current study are the software projects in which the existing software is replaced with some new technology or newer advanced version of the same technology and product. The major issues in these projects are compatibility problems, synchronization issues, data issues and fail back strategy.

This study focuses on one of the major software project success features as identified by the project managers through the focussed group interview: Faster time to market. The detailed process is given in the methodology segment. The discussion led to the finalization and unanimous agreement from the focussed group regarding the priority of Faster time to market aspect in software testing with respect to the manufacturing sector IT projects. Further the agile features that were highlighted by the FGD participants were subjected to Q-sorting by 4 different experienced professionals from the sector and finally the Agile methodology for this study context has been portrayed using 5 parameters, namely: Prioritization of the features according to the customer requirement (AG1); Early Customer Involvement (AG2); Incremental and iterative (Sprint) ready for release in 3-4 weeks (AG3); Flexible development environment with less documentation (AG4); Complexity Management in Agile projects (AG5).

3. Methodology and Sampling

The study consists of a two stage methodology. The first is of qualitative research approach. The study aims at understanding the foremost managerial priority in the context of manufacturing sector IT projects’ software testing and quality assurance perspective. The study further narrows down on the principle attributes of the Agile methodology which are relevant decisive factors in the context of the manufacturing sector IT projects. Qualitative
research is often characterized by the use of multiple methods, which is often referred to as triangulation. Literature indicates that different methods can be used in qualitative research such as participant observation, analysis of texts and documents, interviews, recording and transcribing (Dooley 2001). The principle logic behind all these actions is generation of in-depth understanding of the phenomenon. It is evident from the extant literature that no such scales exist to quantify such an understanding and no prior studies have also been carried out. So, in order to create the framework we treaded the mixed path of qualitative and quantitative approach where we started with the focussed group discussions (FGDs) with around 15 project managers in three sets of five each from three different IT multinational organizations in India. The outcome of the FGD clearly established that one of the most important and relevant managerial priorities is “Faster time to Market” for the software in the context of manufacturing sector IT projects and majority of them impressed upon the relevance of Agile methodologies along the SDLC for superior outcome. This happened to mark the beginning focal point of our study and as a consequence, we also came up with certain key attributes which they suggested to be vital for achieving such priorities in the parlance of Agile Testing methodologies.

As a second stage of our triangulation and qualitative study, for reconfirming the validity and relevance Agile methodology in the current study context, we arranged separate one-to-one unstructured interviews with 12 team leads across 5 IT organizations, since they were directly in touch and abreast with every minute detail of the practised methodologies. The unstructured interviews were started based on the knowledge from extant literature regarding the agile methodology attributes and aim was to validate them based on the responses by the professionals. From this second set of FGDs we obtained 6 major attributes for Agile. Finally as part of the standard practice in qualitative study to achieve reliability, in the third stage we further subjected the identified attributes for Q-sorting procedure with distinct sets of senior testing professionals. The Q-sorting was carried out for three times with two professionals each time rating the validity and appropriateness of the attributes. The process went for three iterations, till the inter-rater reliability of more than 90% was achieved. This led to the finalization of the attributes and the questionnaire instrument which finally had 5 agile attributes. The sixth attribute which was deleted was “Client/Customer involvement in inside team decision making”. This deletion was based on the Q-sort feedbacks in consultation with the professionals. Many professionals shared disparate levels of knowledge and difference of opinion regarding its relevance and also its level of possible correlation and scope definition with the second attribute “Early customer involvement”.

Thus the final questionnaire instrument asked same set of 5 questions representing those 5 selected attributes using a 5-point Likert scale across the first 5 established phases of the software development life cycle (SDLC). The sixth SDLC stage (deployment) in many cases remaining outside the scope of agile methodologies have not been included for the study purpose. The five questions asked across all the SDLC phases for capturing the perception of the professionals regarding the importance of these attributes towards fulfilling the managerial priority of “Faster Time
to Market” are representing the five Agile parameters:

- Prioritization of the features according to the customer requirement
- Early Customer involvement
- Incremental and iterative (Sprint) ready for release in 3-4 weeks
- Flexible development environment with less documentation
- Complexity Management in Agile projects

In the second stage of quantitative research, we administered the questionnaire as a survey instrument to 162 manufacturing and services related software testing project professionals from across five IT MNC in Hyderabad (all of which came in the fortune 500 list) who had used and knowledge of using both the Agile and V-model methodologies during their career. 54 valid and completed filled-in questionnaires were received after a 10 day reminder, finally giving a consolidated response rate of 33% (which is pretty acceptable in case of survey based studies). The questionnaire had demographic data based on which the male to female ratio of the respondents, experience profile of the respondents and current working project type of the respondents were obtained as well. One-way Anovatechnique is helpful to verify the similarity of the variances among the groups. With the help of SPSS software package, one-way ANOVA is applied on the feedback data to examine the varying importance assigned by the respondents to each of the five agile attributes chosen for the present study in each phase of testing.As part of ANOVA, Levene’s test for homogeneity of variances is used to assess whether the population variances for the various groups are significantly different from each other. In continuation, one-way ANOVA table is also helpful to check whether there are statistically significant differences between the group means. If the assumption of homogeneity of variances is not met, Robust Tests of Equality of Means can be used to check whether there are statistically significant differences between group means. In either case, to measure which groups differed from others, multiple comparisons table is useful. Tukey post-hoc test is one commonly used procedure of multiple comparisons and the same has been applied in the study to compare all pairs of group means.

Out of the 54 IT software engineers who are working on IT quality assurance, 29 respondents had experience above 5 years, 18 respondents had 0-2 years of experience and the remaining 7 had below 2 years of hands on experience in agile software testing methodology. All the respondents are from Indian IT multinational companies and working in manufacturing sector IT projects. Their responses are considered as the representative responses as per the priorities expected from the softwares pertaining to the manufacturing sector. As per the type of project the distribution was 21 respondents worked in development projects, 19 in maintenance projects and 14 in migration and upgradation type projects (considered as others) in Table 1. The gender-wise response figures is representative as the male to female ratio in the Indian software industry and especially software testing and quality management domain is close to 2:1.
Table 1: Demographics of the respondents

<table>
<thead>
<tr>
<th>Experience</th>
<th>No of respondents</th>
<th>Gender</th>
<th>No of respondents</th>
<th>Project type</th>
<th>No of respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-2 years</td>
<td>07</td>
<td>Male</td>
<td>35</td>
<td>Development</td>
<td>21</td>
</tr>
<tr>
<td>&gt;2 to 5 years</td>
<td>18</td>
<td>Female</td>
<td>19</td>
<td>Maintenance</td>
<td>19</td>
</tr>
<tr>
<td>Above 5 years</td>
<td>29</td>
<td>others</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>54</td>
<td></td>
<td>54</td>
<td></td>
<td>54</td>
</tr>
</tbody>
</table>

Based on the extant literature on one hand and the FGDs and interviews of the professionals on the other, the five agile parameters and the rationale behind selecting them are as follows:

- **Prioritization of the features according to the customer requirement (AG1):** Requirement prioritization is the key feature of agile process. This process brings value to the deliverables and quickly incorporates changing requirements. (Racheva et al, 2010). Agile believes in value based delivery rather than process based and the easiest way to develop high-value software is to implement the highest priority requirements first. This prioritization is based on client priorities. Therefore, the requirements that the client perceives as having the highest business value for the project are included in the first iteration; the remaining features are further reassessed and prioritised for inclusion in future iterations. In order to maximize value in agile process, customer is closely involved in the analysis of each requirement functionality, business value, importance, story-size and resource availability. In this feature of Agile, prioritizing changes can be difficult where there are multiple stakeholders or customers. Different people have different priorities; they work differently and see the world differently so prioritization process may be a conflict.

- **Early Customer Involvement (AG2):** One of the major causes of the software failure is the communication gap between the customers and the software development team. Lack of user involvement traditionally has been one of the major project failure issues. Even when delivered on time and on budget, a project can fail if it doesn't meet user needs or expectations. In Agile-Methodology, this kind of rework due to misunderstanding can be avoided as the customer works closely with the team and has daily communication with the team about work and any kind of impediments. This Early customer involvement in agile model helps in reducing rework; manage time and performance of the project (Beck, 1999; MacCormack et al., 2001; Highsmith, 2002). Customer working so closely with the development team improves communication dramatically and decisions are made more quickly because the customer is readily accessible. The only problem of early customer involvement within the team is that it can be difficult to keep the interest of customers who are involved in the process.
Incremental and iterative (Sprint) ready for release in 3-4 weeks (AG3): Scrum process follows the incremental development process (Rico, 2004) in which the product is broken down into a set of manageable and understandable chunks. Most of the software development organizations are forced to deal with requirements that tend to evolve very quickly and become obsolete even before project completion (Boehm, 2000). In Scrum process all the functionally related requirements are covered in one sprint of 3-4 weeks. In each of the sprint all the phase like- Plan, Analyze, Design, Build & Test are covered and each of the functionalities is in the release ready state without any backlog and defect. This incremental delivery system shortens the time to market and may result in higher revenue, as each completed backlog represents a new release of the product. In addition, reviewing each sprint before moving to the next means that testing is conducted throughout the process, which allows teams to change the scope or direction of the project at any point.

Flexible development environment with less documentation (AG4): Agile Model believes in flexible and Value-driven approach. Agile software processes prefer face-to-face communication over written specifications. Agile development does not document requirements in detail upfront. In this process as the sprint time is small and entire requirements are to be delivered in 3-4 weeks window thus agile model has an approach that developing extensive documentation and models is counterproductive since most of the time in traditional setup is wasted through oral meetings and communication (Ramesh et al., 2010).

Complexity Management in Agile projects (AG5): Complexity of software can be defined in terms of the number of modules in a project; Number of interactions among the modules or linkages between the modules and the level of differentiation between the modules. In case of complex projects managing agile may be complex and managing the project’s operational performance i.e. dealing with cost containment, delivery flexibility and manoeuvre and quality assurance.

4. Hypothesis, Results and Discussion

All the five agile test attributes were analysed together based on the collected feedback on their importance in each of the five phases of software testing. For this purpose, testing of following null hypotheses was taken up:

- \( H_1 \): All the five Agile attributes are equally important in Test Planning Phase;
- \( H_2 \): All the five Agile attributes are equally important in Test Analysis Phase;
- \( H_3 \): All the five Agile attributes are equally important in Test Design Phase;
- \( H_4 \): All the five Agile attributes are equally important in Test Building Phase; and
- \( H_5 \): All the five Agile attributes are equally important in Test Execution Phase.
From the one-way ANOVA, the feedback given by the respondents on the importance of each of the five agile test attributes have been analysed for each of the five phases of the software testing. Along with descriptive statistics, the homogeneity of variances was tested and F-values derived. In addition, Pos-Hoc (Tukey HSD) tests were conducted to have multiple comparisons among all the five agile test attributes. From the feedback of the 54 respondents, the mean values of the importance of each of the five agile test attributes were computed for each of the five phases. All these values are given in Table 2.

Table 2: Mean values of the importance assigned by respondents on Agile test attributes

<table>
<thead>
<tr>
<th>Agile attribute</th>
<th>Feedback of respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Planning phase</td>
</tr>
<tr>
<td>1) Prioritization of the features according to the customer requirement</td>
<td>4.4630</td>
</tr>
<tr>
<td>3) Incremental and iterative cycles (Sprint) ready for release in 3-4 weeks</td>
<td>4.5185</td>
</tr>
<tr>
<td>4) Flexible development environment with less documentation’ is important for ‘faster time to market</td>
<td>3.5370</td>
</tr>
</tbody>
</table>

Mean values of the importance and impact of challenging issues of Human resources on quality, cost and time of IT projects

Based on the results of means of the feedback collected on all the five agile test attributes phase-wisegiven in Table 2, five graphs were developed and compared with each other, as shown in the figure, Fig. 1.
Comparison of Feedback of Agile Test Attributes among five different phases

Levene statistic along with P-value is derived for all the five phases. It is found that the Levene statistic has a P-value of 0 which is less than 0.05 (5%) in all the five cases. Hence it can be interpreted that the assumption of homogeneity is not met, meaning that there is statistically significant difference among the five agile test attributes in each test phase. The same observation has been established from the Significance of Robust Tests of Equality of Means and also ANOVA. All these results are included in Table 3 along with brief discussion and interpretation of results.

Table 3. Statistically significant differences among the five Agile attributes in each phase of Software testing

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Levene Statistic and P-value</th>
<th>Robust Tests of Equality of Means Significance; and ANOVA F-value</th>
<th>Observed significant differences from Tukey’s Post-Hoc</th>
<th>Most Significantly differing attribute(s)*</th>
<th>Interpretation</th>
</tr>
</thead>
</table>
1) \( H_1 \): All the five Agile attributes are equally important in Test Planning Phase

| (Significance) | test | 4\(^{th}\) and 5\(^{th}\) attributes differ from other three attributes and also differ from each other | 1) There is statistically significant difference in the mean importance given to the agile test attributes; thereby rejecting the hypothesis.

Hence, all the five Agile attributes are not equally important in Test Planning Phase.

2) First three attributes are found very important in Test Planning phase.

2) \( H_2 \): All the five Agile attributes are equally important in Test Analysis Phase

| (Significance) | test | 3\(^{rd}\), 4\(^{th}\)and 5\(^{th}\)attributes are differing significantly from the first two. | 1) There is statistically significant difference in the mean importance given to the agile attributes; thereby rejecting the hypothesis.

Hence, all the five Agile attributes are not equally important in Test Analysis Phase.

2) 1\(^{st}\) and 2\(^{nd}\)attributes are found very important in Test Analysis phase.
### 3) H₃: All the five Agile attributes are equally important in Test Design Phase

<table>
<thead>
<tr>
<th>Test Design Phase</th>
<th>Mean Importance</th>
<th>Variance</th>
<th>Significance</th>
<th>Attribute Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>27.043;0.000</td>
<td>0.000;</td>
<td>33.764 (0.000)</td>
<td>3rd, 4th and 5th attributes differing significantly from the first two.</td>
</tr>
</tbody>
</table>

(1) There is statistically significant difference in the mean importance given to the agile attributes; thereby rejecting the hypothesis.

Hence, all the five Agile attributes are not equally important in Test Design Phase.

(2) 1st and 2nd attributes are found very important in Test Design phase.

### 4) H₄: All the five Agile attributes are equally important in Test Building Phase

<table>
<thead>
<tr>
<th>Test Building Phase</th>
<th>Mean Importance</th>
<th>Variance</th>
<th>Significance</th>
<th>Attribute Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>26.688;0.000</td>
<td>0.000;</td>
<td>12.567 (0.000)</td>
<td>1st and 2nd attributes differ from 3rd and 5th but not between themselves; and 3rd and 5th differing from 4th but not between themselves.</td>
</tr>
</tbody>
</table>

1) There is statistically significant difference in the mean importance given to the agile attributes; thereby rejecting the hypothesis.

Hence, all the five Agile attributes are not equally important in Test Building Phase.

(2) 1st, 2nd and 4th attributes are found very important in Test Building phase.

### 5) H₅: All the five Agile attributes are equally important in Test Execution Phase

<table>
<thead>
<tr>
<th>Test Execution Phase</th>
<th>Mean Importance</th>
<th>Variance</th>
<th>Significance</th>
<th>Attribute Differences</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5.886;0.000</td>
<td>0.000;</td>
<td>8.141 (0.000)</td>
<td>3rd attribute</td>
</tr>
</tbody>
</table>

There are significant differences among the pairs of attributes: 1st - 3rd and 3rd - 5th attribute

1) There is statistically significant difference in the mean importance given to the agile attributes. Hence the hypothesis is rejected;
Hence, all the five Agile attributes are not equally important in Test Execution Phase.

2) 1st and 5th attributes are found very important in Test Execution phase.

*: The attribute that differs significantly from all the remaining attributes.

Results indicate that in the test planning phase, the prioritization of customer requirements, early customer involvement and incremental iterative cycles are most vital. In the test analysis and design phases, both prioritization of customer requirements and early customer involvement attributed vital importance. For the test building phase, the issues of prioritization of customer requirements, early customer involvement and flexible development environment with less documentation are very important. In the test execution phase, importance has been attributed to the issues of prioritization of customer requirements and complexity management.

Comparing all the above results, it can be stated that the issue of prioritization of customer requirements is most vital and common across all SDLC phases of agile testing. Next to this issue, the issue of early customer involvement has found vital importance across the first four stages. Therefore, it can be ascertained from the feedback collected, that the first two agile attributes, namely, prioritization of customer requirements and early customer involvement attributed vital importance, whereas the fifth attribute, complexity management happens to be the most significantly differing agile attribute. However the issue of complexity management attained importance in the execution phase. These findings could provide a handy framework for the project team members and managers in agile projects to prioritize and handle the agile testing issues in each of the SDLC phase.

5. Conclusion and Managerial Implication

This study is a step ahead in the field of software testing and delivery methods and provides a framework towards selection of the phase-wise key attributes. Prioritization of the features according to the customer requirement has been identified as the most important Agile test attribute across all the five SDLC phases and this has very high implication for the managers in terms of project planning and co-ordination for the managers. So, the project managers and team leads need to keep the provisions for planning the project stages keeping in mind the main customer requirements. If for a project the customer requirement is user-friendliness of the software, then no matter how robust the application is the final delivery and acceptance from the customer end will suffer if the customers objective is not met. The second attribute of early customer involvement has also registered very high importance across the first four stages and this seems to be a logical extension as well in the agile domain. The second attribute can be thought of a measure to reduce rework and prevent the project from going off the track from the primary
project objectives and requirement.

Incremental and iterative releases have gained importance in the planning phase while flexible development environment with less documentation has become vital in the build phase and complexity management has been proved to be most important in the test execution phase. These findings appear completely in sync and very significant in terms of managerial implications because during the planning phase itself the managers need to ensure that provisions for incremental changes are maintained and the flexibility in terms of documentation directly reduces the time to delivery of the completed software module after testing. However the last attribute have rightly found major importance in the final execution phase because unless the complexity aspect is controlled the implementation and the delivery of the module will suffer leading to delay in time to market.

Thus this study provides a handy framework based on empirical findings regarding the relevance and importance of the agile attributes in software projects related to manufacturing sector and will help managers and professionals to make a much more rational and streamlined planning of their respective projects.

References


