Analysis of Open Source Software Development From Agile Perspective

Harish Rohil¹, Manisha Syan²
Assistant Professor ¹, Research Scholar ²
Department of Computer Science and Applications CDLU, Sirsa ¹, ²
harishrohil@gmail.com ¹, manishasyan@gmail.com ²

ABSTRACT: Open Source Software Development is becoming a fundamental area in Information Technology Companies. It is a known fact that traditional methods have certain limitations on changing technology for software development. After years of been practiced in the industry the Agile software development process possesses standard characteristics of a process paradigm. It is rare to observe studies on Agile practice used in IT industry with its impact to student learning. This paper describes findings of such study conducted in a Company environment. The study was conducted to make a survey about how the companies are following open source development with sample size of 10. The results indicate that companies are using both agile and traditional ways of software development.

I. INTRODUCTION

Open source software development is the process by which open source software (or similar software whose source code is publicly available) is developed. These are software products “available with its source code and under an open source license to study, change, and improve its design”. Examples of popular open source software products are Mozilla Firefox, Google Chromium, Android and the OpenOffice.org Suite. In the past, the open source software development method has been much unstructured, because no clear development tools, phases, etc., had been defined like with development methods such as Dynamic Systems Development Method. Instead, every project had its own phases. However, more recently there has been much better progress, coordination, and communication within the open source community. With Globalization in software industry Agile Practices comes into act and provides huge benefits and changes in approach of software development worldwide. We have analyzed the growth, issues, methodologies and vision of open source software development implementation of Agile Practices.

A. AGILE SOFTWARE DEVELOPMENT METHODS

Documentation

For documentation, the agile viewpoint is that there should only be the bare minimum of it [13]. Maintaining extensive documentation is a primary requirement only for setting up the next game, as the updated documentation is only needed to make it possible for new developers to understand the system. In the gamer’s perspective, setting up the next game is only a secondary goal. If the primary goal is not reached, the game ends in any case. Therefore, there should only a sufficient amount of documentation.

Simple Design

To keep the code easy to change, agile methods favour a simple design. The idea is to develop the simplest solution that could possibly work for the current feature. Since the future is uncertain, it is useless to try to design the system as it is at the end of the project. If and when the design solution is not good enough anymore, it is refactored to meet the new requirements. Refactoring must also be used to keep the software structure as simple as it can be and to remove duplicated code. This idea of designing through refactoring keeps the code clear and allows developers to design only what is required to implement the next feature. Without the extensive unit test cases that need to be built, the developers might be unwilling to perform the required refactorings to keep the code simple.

Metaphor

Architecture is not a term used very often with the XP methodology, in its place, XP uses an all-encompassing story called the metaphor that describes and controls software development projects. The metaphor describes a comprehensible story shared amongst all stakeholders involved in a project. XP discards architectural documentation such as meticulous system design specifications seen in more common traditional approaches.

On-site Customer

The on-site customer practice of XP aims to encourage rapid feedback and decision making, with face-to-face communication between the customer and development team. ‘A real customer must sit with the team, available to answer questions, resolve deputes, and set small-scale priorities. By “real customer” we mean someone who will really use the system when it is in production’. ‘An on-site customer is rapidly aware of exactly how fast development proceeds, and the planning game makes them aware of how it is likely to proceed in future. If this rate is not sufficient to meet deadlines, customers are prone to fall back on old habits and demand increased development speed by various flawed practices’.

II. LITERATURE REVIEW

In [1], it has been discussed that Open Source Software Development is increasing day by day due to the mass acceptance of open source software’s by different communities worldwide. It was a discussion about
conventional software development and open source software development.

In [2], differences between the two styles of development have been discussed. In closed source software development, the programmers are often spending a lot of time dealing with various issues. In open source software development issues are solved by integrating the users of the software in the development process, or even letting these users’ builds the system themselves.

In [3], it has been discussed that it is not feasible to run an open source project following a more traditional software development method like the waterfall model, because in these traditional methods it is not allowed to go back to a previous phase. It was concluded that tools are free and support casual use, more members of the development team will be able to access and contribute to artifacts in all phases of development.

In [4], it has been discussed that besides requirements, often volunteer staff is attracted to help develop the software product based on the early releases of the software. It proposed a definition and a classification of agile software development methods. It analyzed ten software methods that characterized being “agile” against the defined criteria. It compared those methods and highlighted their similarities and differences. It was concluded that the software development is:

- Incremental (small software releases with rapid cycles),
- Cooperative (customer and developers working constantly together with close communication).
- Straight forward & adaptive.

In [5], it has been discussed “rapid prototyping, incremental and evolutionary development, spiral lifecycle, rapid application development, and, recently, extreme programming and the agile software process can be equally applied to proprietary and open source software”. Open source development agile methodologies are Extreme Programming and Internet-Speed Development.

In [6], provides numerous proficient arguments for using open-source software in the commercial world, such as low costs, no vendor lock-in, and reliability. Open source projects are guided by technology instead of business needs, with technical excellence often as the primary goal. This publication described the ease and benefits of open source software development in industry and to customers.

According to [7], open-source software can be freely used, modified and distributed provided that certain restrictions are observed. In this research author described the importance of open source software in development industry and also about different versions and model for open source development. It was concluded on the basis of subjective evaluation that mature open-source solutions and agile development methods can work well together.

In [8], it is described some qualitative reflections and observations on the nature of open source software and most popular and important claims associated with the open source approach and further explained the concepts and intuitions that are really peculiar to open source, and to distinguish them from features and aspects that can be equally applied to or found in proprietary software. It was concluded that most claims associated to open source and the related development process do apply also to proprietary software.

In [9], described the implications of structural quality and benefits of structural quality analysis of the code delivered by open source style development. Overall, it is concluded that an open source process with the following features:

- The definition of a programming standard to be respected by the project participants at the launch of the project.
- The statically source code analysis in the stage before release content definition, to measure the code developed and verify conformance to the rules imposed.
- The utilization of the measurement results in the configuration of the new release.

In [10], described the success of open source in price against the software in market, in other word explains the total cost of ownership of open source really lower than that of proprietary software. It was concluded that Open source offers cheaper price for direct cost. The total cost of ownership of open source based applications can be higher or lower than proprietary software.

In [11], it is presented experiences of using pair programming (PP) extensively in an industrial project. The developers considered that pair programming improved quality and knowledge transfer, and was better suited for complex tasks than for easy tasks.

In [12], described the social network view for development and adoption of open source software. In this the number of developers has adopted the open source approach. These projects were modeled as a collaborative network, with developers as nodes and joint membership in projects as links between the nodes.

III. RESEARCH METHODOLOGY

The research is conducted with the different employees of different companies. The survey is done to see whether the companies are following agile practices or traditional practices for software development. To study the nature of Open source development strategies being adopted by the Software companies with degree of the agility of the Software Delivery so that we can analyze the adoption of open source software.

Objectives:

To analyze the nature of open source software development in Indian industry and to analyze whether
the companies are following agile practices or traditional practices for software development.

3.1) Research Design: Research design provides a blueprint for a study. Descriptive and exploratory research design will be followed.

3.2) Sampling Design: For the purpose of data collection, the whole Open source Industry will be divided into different clusters. A well structured questionnaire is prepared to obtain responses from target developers.

Sample Size: 10
Sample units: Software development team.
Sampling Technique: Random Sampling/Cluster Sampling.

IV. EXPERIMENTAL RESULTS

Structured questionnaire are used to collect data from different information technology companies. The questionnaire is prepared by using the seven point likert scale. Interviews of respondents are also conducted for the collection of information regarding Open source adoption in prospective of Agile techniques.

Questionnaire:
To achieve the set of objectives, survey questionnaire are presented to the respondents with the criteria to assess the knowledge sharing techniques in organizations. Questionnaires are filled by the various employees from various departments of the companies.

The questionnaire is based upon elements of existing Open source sharing structure. It is divided into four parts:
1. General information:- This part of questionnaire consists four questions to collect information regarding potential open source sharing policies.
2. Current situation:- This part of questionnaire consists nineteen questions about the use of current methodologies.
3. Gaps and Desiderata:- This part of questionnaire consists three questions about to identify gaps in existing techniques.
4. Future lines of research:- this part of questionnaire consists six questions about the adaption of new techniques.

Test of hypothesis:
For checking whether companies are using traditional approach or distributed agile approach hypothesis is set. In this study chi-square test is used to check the acceptance of hypothesis.

The hypothesis used in this study is given below:

Since a p-value is greater than the conventionally accepted significance level of 0.5 (i.e. \( p < 0.05 \)) we will reject the null hypothesis. In other words, Companies are working under distributed Agile approach is in use.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Agile</th>
<th>Traditional</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company 3</td>
<td>40</td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td>Company 4</td>
<td>38</td>
<td>20</td>
<td>58</td>
</tr>
<tr>
<td>Total</td>
<td>78</td>
<td>40</td>
<td>118</td>
</tr>
</tbody>
</table>

Chi-Square calculations:
\[ \chi^2 = 98 \left( \frac{22 \times 18 - 20 \times 38}{40 \times 58 \times 60} \right) \]
\[ \chi^2 = 0.5834528/5289600 \]
\[ \chi^2 = 1.103 \]

Df= (2-1)*(2-1)= 1

Alpha level of significance: 0.5

The corresponding probability is below 0.10 probability levels. That means that the p-value is above 0.5. Since a p-value is lesser than the conventionally accepted significance level of 0.5 (i.e. \( p > 0.05 \)) we fail to reject the null hypothesis.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Agile</th>
<th>Traditional</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company 5</td>
<td>38</td>
<td>0</td>
<td>38</td>
</tr>
<tr>
<td>Company 6</td>
<td>40</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>Total</td>
<td>78</td>
<td>10</td>
<td>88</td>
</tr>
</tbody>
</table>

Chi-Square calculations:
\[ \chi^2 = 88 \left( \frac{38 \times 10 - 40 \times 0}{38 \times 50 \times 78} \right) \]
\[ \chi^2 = 153 \left( \frac{144400}{1482000} \right) \]
\[ \chi^2 = 8.574 \]

Df= (2-1)*(2-1)= 1

Alpha level of significance: 0.5

We now have our chi square statistic \( (\chi^2 = 8.574) \), our predetermined alpha level of significance (0.5), and our
### V. CONCLUSION

In our research we have analyzed the growth, issues, methodologies and vision of open source software development with implementation of Agile Practices. A survey was conducted to see whether the companies are using traditional or agile practices of software development. It is concluded from the results that companies are using both agile and traditional practices. The result shows there is equality between both the ways of open source software development.

### VI. REFERENCES


13. Antony R. Grinyer, “Investigating the adoption of agile software development methodologies in organisations”, ISSN 1744-1986