Enhancing the Effective Communication in Distributed Agile Environment

Arushi Walia

M. Phil. Research Scholar, Punjabi University, Patiala, India

Abstract—Agility has become today's buzzword when describing a modern software development process. Organizations combine agile approach and Distributed Software Development (DSD) in order to reap the benefits of both agile and distributed development such as to develop better quality software solutions in lesser time and cost. Agile approaches highly values communication between team members and customer collaboration to improve software development processes, even though, communication in globally distributed agile teams can be difficult. No doubt, a number of tools and services exist but these do not support the effective communication among team members which is vital for knowledge sharing in distributed agile environment for better project planning. Besides, using these tools requires proper training sessions by the team members and customers to keep them up with pace and thus proves to be costly for small and medium sized organizations. Moreover, inadequate customer involvement causes adverse effects on the success rate of distributed agile projects. In order to address these, Another Project Progress Tracking System (APPTS) is developed. As a main characteristic, it plans and organizes the software project as a series of iterations that are guided by customer's exact requirements through adequate customer involvement and effective communication among team members. The developed approach is implemented on two distributed agile software development companies which have been adversely impacted due to lack of communication and inadequate customer collaboration. The results provides distributed agile teams with an improved awareness of the actual progress and thus increases the success rate of projects by 20-35% resulting in more satisfied customers.

Keywords—Distributed Agile, Distributed Software Development (DSD), Collaboration, Communication, Knowledge Sharing, Tracking System

I. INTRODUCTION

Agility has become today’s buzzword when describing a modern software development process. During the mid nineties [5], some software engineering practitioners and consultants introduced a new group of software process methodologies called Agile Software Development to react against “Heavyweight” methods such as the Waterfall Process Model. Agile Methodologies do insist on minimal documentation and documenting only what is necessary and the ability of the process to adapt to changes. Due to these, Agile Methodologies came to known as Lightweight. Unlike other software engineering projects, it is based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams [5], [9], [35]. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change.

Agile techniques may vary in practice but they share common characteristics like iterative development and focus on interaction and communication. Agile is a practice driven and communication oriented paradigm which highlights the importance of better communication [11], [17], [21] within the development team and closer contact between developers and their customers.

In recent years, there has been an increasing interest in the integration of agile practices in Global Software Development (GSD). The idea behind this is that companies realized the competitive advantage of doing it globally as well as producing higher quality software at cost within the stipulated time plus able to manage late requirements. In Distributed Software Development (DSD) the team members may be located in different remote sites during a software development lifecycle process and thus making up a network of distant sub-teams. In some cases, these teams may be members of same organization while in others, collaboration or outsourcing involving different organizations may exist.

It is important to note that knowledge itself cannot create significant value without proper utilization. One of the main challenges in Agile Global Software Development is the information flow or knowledge transfer among the distributed team members. Nevertheless, communication is considered as one of the important elements in AGSD which allows knowledge transfer [2], [26] between team members; allow understanding of requirements from customers as well as development activities can be performed effectively and efficiently.
It has been seen that most of the distributed agile projects failed due to the inability of properly managing the communication and processes. As a part of research activity, information repository has been developed to enhance the knowledge sharing which is only possible by effective communication among team members in distributed agile environment. That is, to make them available with a central repository to share their knowledge at a common platform at much lower cost and ease of using which require no training sessions or Agile coach/ Scrum master. To practically check the developed approach, the idea has been shared with two leading software development companies to implement in their development work. The developed approach increases the success rate of projects by 20-35% by increasing the flexibility with which the development team can respond to changes in requirements, enhances customer involvement and effective communication, increases team productive time and allows showing progress frequently.

The Contributions can be summarized as:

1. Analysis of agile software development methodology has been performed. During the analysis it has been observed that the most of the research has been done on distributed agile environment.

2. Research papers on distributed agile environment have been studied thoroughly to get better idea. These papers helped to get a clear vision of what distributed agile is and where it can be best deployed.

3. Problems occurring in distributed agile environment have been studied thoroughly. The major problem occurring in their way to success is absence of effective communication, which is essential to share knowledge among distributed agile team members. Also it became hard for small and medium-sized organizations to afford such costly services and tools. To overcome the problem, information repository has been developed.

4. The proposed approach has been implemented using PHP Programming language and MySQL on Apache WampServer. The developed approach has been tested by sharing the idea with two leading software development companies to implement in their development work.

5. It increases the success rate of projects by 20-35% by enhancing effective communication among team members for knowledge sharing and allows showing progress frequently. Thus performance has been found to be satisfactory.

II. RESEARCH SURVEY

Johanna and Shane [6] have shared their learning from leading workshops about geographically distributed agile teams. According to which the mere ability of collaboration between teams is determined by the tools and facilities available to the members that provide better means of communication which is very essential to build trust and knowledge sharing without any hesitation.

Julie and Helmut [7] have discussed some of the most common challenges distributed teams face in order to lead successfully in off-shoring within an agile model. The authors have pointed out that almost all of the other challenges can be mitigated either partially or completely by mastering the greatest challenge of all: Communication.

Michael and Stephen [11] have presented a successful case study in distributed agile project using outsourcing. Along with this team developed a customized site for team contact information, status updates, announcements etc. But the crucial area for success was the creation of a strict communication plan.

Nayan Hajratwala [15] has described the importance of task boards in agile teams. Task board acts as a communication channel for all the team members as they would be able to identify the actual status of the project.

Nina, Noor and Azlinah [17] have identified thirteen chaotic issues that adversely affect the software development process in distributed environment. Out of which the most common challenges existed in AGSD are related to communication.

Sukanya, Jigar and Elliott [22] found that information has been maintained by using living documents, well-connected communication and working software. That is, scrums, or frequent, informal and intense communication is enough to sustain information within an organization for a long period of time.

Zakaria, Amelinckx and Wilemon [26] have concluded that although many outsourced agile projects have been frequently successful but others failed in achieving their objectives and face difficulties due to global diversity, distance, lack of communication etc.

III. ANOTHER PROJECT PROGRESS TRACKING SYSTEM

The motivation behind designing APPTS, the information repository, is to enhance the knowledge sharing which is only possible by effective communication among the team members in distributed agile environment. That is, to make them available with a central repository to share their knowledge at a common platform at much lower cost and ease of using which require no training sessions or Agile coach/ Scrum master.
It has cut down the extra cost of development and training required, saved a lot of development time as it required no training sessions and has easy to use features. APPTS consists of three parts:

- Management Portal
- APPTS Server Application
- APPTS Client Application

Figure 1: Structure of Another Project Progress Tracking System

Functionality of those parts can be summarized as:

- **Management Portal**
  1) Configurations of the tool users and management of access rights
  2) Repository management
  3) Project management
  4) User management
  5) Location management
  6) Management of day book reports
  7) Manage attendance reports of users

- **APPTS Server Application**
  1) Uses notifications/ reminders/ urgent messages to inform all the related users of the project when necessary.
  2) Communication between clients/users
  3) Check for changes in project status and inform related clients.
  4) Collect information from users and generate necessary actions.
  5) Make announcements to notify all users about progress change like problems or requirements volatility etc.
  6) Uses email notifications

- **APPTS Client Application**
  1) Manage own day book reports
  2) View and analyze the project status
  3) Adds knowledge to the project progress details
  4) Add detailed information about the status change in a pop-up message attached with each status update.
  5) Generate email notifications.
  6) Post status messages about problems occurring to seek help from other team members.
  7) Managers make announcements regarding requirements change etc. on behalf of customers.
  8) Testers immediately reports about the testing bug, if any.
  9) Customers have limited access in order not to harm the progress by unnecessary interference.
  10) Customers can view and add requests for the new/modified requirements.
  11) Customers can also report the bug system if they do not found the development according to their requirements only at the end of each iteration.

A. The Proposed Approach in a step-wise manner can be summarized as:

1. Log-in to the portal with assigned user-id and password.
2. Select the location and project assigned you want to work upon.
3. View/analyze the project status from the information repository.
   a) Refined view of the project status in order to track information by using search facility:
      - category/ phase wise
      - user wise
      - through keyword
      - date wise
4. Communicate by updating/changing the project status.
   a) Detailed information about the progress change can also be sent through pop-up window attached with each status.
   b) Automatic email notification denoting every change.
   c) Update status about problems occurring to seek help from other team members.
5. A unique identification number/ticket number/reference number is attached with every status update for easy access and retrieval by users. Also different coloring scheme is assigned to each phase for easy access.
IV. RESULTS

To practically check the proposed solution the idea has been shared with two leading software development companies to implement in their development work.

As one company, ‘Rom Group’ at Patiala faced numerous hurdles while developing the solution for their customers. The reasons behind the delay comprised of lack of customer coordination, no idea about distributed agile practices so the developers have not been able to cop up with new requirements arrived late in the project.

Another case is CSCS company, Delhi. The distributed developers in the team knew that they have been working on Agile practices but they had no tracking system to track the current status of the project. Lack of necessary communication caused the great amount of replicated work, team members were not aware about what was happening in the project on the other side.

Due to this lack of knowledge sharing capabilities the delay occurs in project completion.

A. Case Study:

CSCS Software Company was having difficulties while working on the project assigned by company heads that is Attendance Management System using Agile practices in Distributed environment. They faced numerous problems while working on the project. Resulting which the success rate of the project started declining at very high pace.

A1): Problems occurring with the project before implementing APPTS approach:
1. All team members reside at distant locations; this causes a big barrier in sharing ideas.
2. One team member doesn’t know the exact status of work assigned to another team member.
3. Lack of proper communication between team members, which results in mismanagement of the tasks assigned.
4. Inefficient team member can hamper performance of other (efficient) team members.
5. Replication of work arises.
6. No common space for team contact, information sharing, status updates, announcements, discussions and documents.

A2): Problems solved by implementing APPTS approach:
1. Projects/modules can be assigned to any team member. Monitoring can also be done on the same.
2. Every team member can update all others about the status of the work assigned.
3. Provides a common platform for discussing/monitoring the task assigned.
4. Managers can track performance of every employee through day book. Analysis can also be done on the same.
5. Manages attendance of all employees.
6. Location of employees working at different geographic locations can be verified.

7. No need to use emails for discussing and updating information on a task. Portal automatically communicates all the necessary communications among all team members.

8. A reference number (ticket) is assigned to every task. Every user completes its job on a task with respect to reference number only.

9. Manager can monitor the work done by any team member at any time.

### Table I

<table>
<thead>
<tr>
<th>Factors</th>
<th>Performance Before using APPTS</th>
<th>Performance After using APPTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company A: Rom Group</td>
<td>Compan y B: CSCS</td>
<td>Company A: Rom Group</td>
</tr>
<tr>
<td>Requirements Volatility</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Team Work</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Business Value</td>
<td>Low (1-2 iterations)</td>
<td>Medium (2-3 iterations)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Daily Brief</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>No. of cycles</td>
<td>1-2 iterations</td>
<td>2-3 iterations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cycle Length</td>
<td>3-4 months approx.</td>
<td>2-3 months approx.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reflection (Feedback &amp;</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Morale)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Knowledge Sharing</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Risk Reduction</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Customer Collaboration</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Customer Satisfaction</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Keeps History</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

Figure 4: Progress status check along with the attached pop-up window with each status update using APPTS.

Figure 5: Manager can check the progress of any employee from history
V. CONCLUSION

No doubt a number of web applications, tools, approaches, information radiators exists but all these provide services at a very high cost and which can be utilized with heavy frameworks or plug-ins. Also, using these tools etc. requires proper training sessions by the team members and the customers to keep up with the pace and thus doubles the cost of development and also a time consuming process. In spite of all this, they don’t solve the major problem occurring in their way to success that is effective communication which is essential to share knowledge among distributed agile team members. So it became hard for small and medium-sized organizations to afford such costly services and tools. As a result they more prefer to adopt other means and services which can be easily afforded by them. But this has not produced desired results rather decreased the morale of team members with unsatisfied customers. Another Project Progress Tracking System, APPTS has solved this problem as it increases the flexibility with which the development team can respond to changes in requirements, enhances customer involvement and communication, increases team productive time and allows showing progress frequently. It increases the success rate of projects by 20-35% as seen in results obtained above. The future work involves adding functionalities as per the requirements of companies in specific domains.

REFERENCES

[7] Julie Calboutin and Helmut Holst, Whitepaper- Distributed Agile : Pointless or possible?, SQS- the world’s leading specialist in software quality, August 2013


[27] Lee G., Xia W., 2010, Toward agile : An integrated analysis of quantitative and qualitative field data on software development agility. MIS Quarterly 34, Pages 87-114.


