Agile Communication Model for Distributed Software Development

S. Bhalerao  
AITR, Indore, India  
Email: bhalerao.shilpa@gmail.com

M. Ingle  
IIST, Indore, India  
Email: maya_ingle@rediffmail.com

Abstract - Global software development generated a new era in software development by eliminating boundary of development team. Distributed software development has its own challenges and implementation of agile software development in it further brings the challenges due to their contradictory practices. In this paper we addressed the issues of DSD and proposed agile communication model for DSD. This model uses the concept of pair programming in intra and inter-group of development team for reducing the communication issues in DSD thereby mitigating the risk and delays of software.

II. CHALLENGES

Agile software development process advocates face-to-face communications, close collaboration, close interaction between customer and team [GT08]. Implementation of ASDP in DSD emerges in to a lot of challenges due to its contradictory practices. These challenges are discussed as follows:

Communication

Agile manifesto states that individual interaction over process and tools is more important in ASDP to manage volatile requirements of project. On the contrary, distributed team suffers from the problems of communication and coordination widely. Further, verbal or face-to-face communication practice of ASDP generates the complexity in DSD.

Design Knowledge

Design knowledge refers to the understanding of functional specification, design standards, coding standards, transformation of knowledge from person to person. For example, architecture of the software is designed at one place but implemented at different places. It has been reported that design rationale of the software is generally misunderstood by distributed teams due to improper knowledge transition, therefore affecting the overall software quality. Knowledge sharing and resource sharing become more challenging in case of chief architect and software engineer when work at different places.

Other issues related to ASDP in distributed environment such as cultural differences, project management activities, frequent requirements changes at various sites, organization size etc. may sometimes create the hurdles [LDO03].

III. AGILE COMMUNICATION MODEL

In this paper, we propose an agile communication model for implementing ASDP in DSD. This model uses the concept of pair programming in intra and inter-group of development team for reducing the communication issues in DSD thereby mitigating the risk and delays of software. Section 2 discusses the challenges in DSD environment. Proposed model is introduced in Section 3 with practices in details. Concluding remarks are presented in Section 4.
illustrate model, let us assume the two teams working on same project in distributed environment with single customer, unavailable to devote much time with both the teams. Figure 1 represents one scenario of intra and inter pair programming model. Team A (with team members in red color) and Team B (with team members in blue color) work on single project but are dispersed in two cities as shown in Fig. 1. Here, the customer is represented in green color, PC stands for Proxy Customer, LR A represents Local Repository of team A and LR B is used for Local Repository of team B.

With this structure of the model, we have introduced some practices that resolve the above discussed severe problems and are as follows:

**Initiating with Active Communication Media**

A brainstorming meeting may be conducted in which developers will discuss with the customer using active communication media such as video conferencing, face-to-face etc. Thus, it becomes possible for the whole team to get aware of the project and the associated vital factors. Such type of brainstorming is beneficial in identifying project attributes such as uncertainty, estimation of cost, size, duration, risk etc.

**Introducing Proxy Customer**

Due to unavailability of the customer on both sites, experienced person in each team may play the role of customer. In Fig.1, Proxy Customer (PC) is represented in green color. Team members of both the teams may be in contact with PC by using instant messaging, blogging, emails, chatting. PC is also responsible in gathering information and feedback from customers. This process makes both customer and team representatives aware of latest updates of software development.

**Suitable Decomposition**

This practice suggests the decomposition of a project by using a criterion of low coupling and high cohesion tasks. It will minimize communication between team members located at different places.

1. **Pair Programming**

It is suggested that each team should follow pair programming for fastening the development process. 1. pair programming refer to intra and inter pair programming in teams. Pairs may be formed in such a manner that every member of team A is paired with the member of team B in rotation during each phase of development. A sequence of pairing among both the teams must be followed as shown in Fig. 2. Let us consider that team A consists of 3 pairs and team B also involves 3 pairs in it. Firstly, the intra-group paired sequencing has to be performed in team A and team B. Secondly, we suggest the inter-grouping the team members. Thus, an intra-group sequencing is maintained by shifting one member of pair P2 to P1 in the structure of team A, one member from P3 to P2 of team A. Also, the sequencing of pairs is performed within team B. The inter-grouping of members will take place when one member of pair P1 from team B shifts to the P3 pair of team B. In this manner, whole team is aware of progress of the project in terms of functionality as well as of technicality. Thus, principles of agility may be introduced in Agile Development of software in DSD.
We may maintain the shared repository locally at each team site. Globally, one shared repository must be available for keeping both the teams updated regarding the current project activities. This repository must be updated on daily basis. It is observed that this shared repository may be useful to reduce the conflicts during requirement gathering and analysis, design etc.

**Use of Internet Enabled Tools**

Our Agile model in distributed environment recommends the use of internet enabled tools for managing the following issues:

i) **Requirement Management**: Requirement management tools maintain requirements in the form of stories or functional specification or UML. These requirements also maintain the dependencies, responsibility and change histories of a particular requirement.

ii) **Source Code Management**: The source code management tools keep track of various versions of source code, designed documents and other files required by the team during the software development.

iii) **Configuration Management**: It maintains the change reports, bug reports, new feature requests and progress report of the software.

iv) **Test Management**: Test management tools manage test plans, test cases and test results.

v) **Project Management**: Project schedules, task breakdowns work assignments and progress of the project are maintained using these tools.

**Agreement on Tools and Technology**

It has been observed that with availability of broadband internet connectivity, networking issues have been almost resolved whereas technical issues in case of distributed environment may be resolved by using planning poker game. In this game, team A and team B will participate as a whole so as to take collaborative decisions on tools and technology used for developing software.

**Effective Communication**

Communication is a major hurdle in DSD for reducing the delivery time of the project. Further, ASDP prefers active communication and individual communication throughout the development of the software. ASDP incorporates two types of communication namely; communication with customer and communication among the team members. Our model ensures the communication among the team members by using intra-pair programming. At the same time, communication among the members of different team will increase due to inter-pair programming practices. On the other hand, communication with customer may be effective and improved with use of proxy customer.

**Solution to design and development tool**

It has been observed that the broadband internet connectivity has almost resolved networking issues whereas all practices proposed in model is a step towards resolving the technical issues using pair programming.

**Knowledge Transfer**

Inter and intra team pair programming proposed in model and concept of proxy customer are source of transferring knowledge from person to person.

**Risk Mitigation**

Distributed software development may possess the risk of ill-defined requirements. It may be reduced after the use of this model due to concept of proxy customer.

**Whole Team**

Agile software development process emphasizes on whole team as to make aware about status of the project and discuss the problems and plans for future developments. Proposed model recommends certain practices (i.e. pair programming, daily meeting of team with proxy customer, maintaining blogs and wiki) which are the source of information and provide the feeling of virtual teams. Collaborative decisions leads to virtual team with feeling of ownership.

**Building Trust**

Model emphasizes on inter-pair programming to encourage team member to meet personally with each other at least once. Thus, this practice provides facility to understand each other and develop trust and confidence among the team members.

**Keep Remote Team Involvement**

Practices specified in model ensure the involvement of remote developers by continuous interacting with
planning poker, Wiki, blogs etc. I² pair programming model addresses maximum issues related to implementation of ASDP in DSD as well as issues of individual DSD environment.

Thus, proposed model is a step towards implementing agile practices in DSD for getting benefits of both strategies. However, model needs verification in case of large team.

REFERENCES