

# An Agile, Intelligent and Scalable Framework for Global Software Development

Raja Asad Zaheer, Aisha Tanveer, Hafza Mehreen Fatima

**Abstract**—Global Software Development (GSD) is becoming a common norm in software industry, despite of the fact that global distribution of the teams presents special issues for effective communication and coordination of the teams. Now trends are changing and project management for distributed teams is no longer in a limbo. GSD can be effectively established using agile and project managers can use different agile techniques/tools for solving the problems associated with distributed teams. Agile methodologies like scrum and XP have been successfully used with distributed teams. We have employed exploratory research method to analyze different recent studies related to challenges of GSD and their proposed solutions. In our study, we had deep insight in six commonly faced challenges: communication and coordination, temporal differences, cultural differences, knowledge sharing/group awareness, speed and communication tools. We have established that each of these challenges cannot be neglected for distributed teams of any kind. They are interlinked and as an aggregated whole can cause the failure of projects. In this paper we have focused on creating a scalable framework for detecting and overcoming these commonly faced challenges. In the proposed solution, our objective is to suggest agile techniques/tools relevant to a particular problem faced by the organizations related to the management of distributed teams. We focused mainly on scrum and XP techniques/tools because they are widely accepted and used in the industry. Our solution identifies the problem and suggests an appropriate technique/tool to help solve the problem based on globally shared knowledgebase. We can establish a cause and effect relationship using a fishbone diagram based on the inputs provided for issues commonly faced by organizations. Based on the identified cause, suitable tool is suggested, our framework suggests a suitable tool. Hence, a scalable, extensible, self-learning, intelligent framework proposed will help implement and assess GSD to achieve maximum out of it. Globally shared knowledgebase will help new organizations to easily adapt best practices set forth by the practicing organizations.

**Keywords**—Agile project management, agile framework, distributed teams, global software development.

## 1. INTRODUCTION

TRADITIONALLY, the software development was accomplished by the co-location of teams, and was in a close physical proximity. Due to the co-location of teams, the team can easily do face to face communication and information sharing. Now the software industry tends to move their production units in decentralized manner across different geographical areas. One reason for this decentralization is

Raja Asad Zaheer, PMI-ACP is with Sofnix Pvt. Ltd. since 2007, an Australian Offshore company in Pakistan and is visiting faculty member in Riphah International University, Islamabad (e-mail: rajaasad@gmail.com).

Aisha Tanveer, MS-SE student at Riphah International University Department of Computing, Islamabad (e-mail: ash.tanver@gmail.com).

Hafza Mehreen Fatima, MS-SE student at Riphah International University Department of Computing, Islamabad (e-mail: mehreen\_fatima@yahoo.com).

access to skilled resource pool across different locations, work shift, optimization of resources and cost to develop quality software. Many large organizations have multiple software groups that reside at different countries of the world. To accomplish certain projects and goals, team with cross section expertise, knowledge, skill and perception is required. The GSD is becoming the common norm in software industry due to the potential benefits gained from it. GSD is the development of the software across different geographical locations by distributed teams. Through GSD, organizations can capture talent across the globe which is not available locally and the cost of the labor is reduced. Employing multiple teams at different location for software development requires more organized practices for management of the project. As in GSD, teams are not co-located which causes various problems, such problems should be well understood to gain the possible benefits from it. Distance can result in hiding of serious problems and withdrawal from decision making. So there is a solemn need for some effective project management methodologies to handle GSD issues. Now, software industry tends to move towards agile methodologies because there is a hidden assumption that traditional methods are not enough [13]. Agile methodologies were first used for small projects [15] but due to its success for collocated teams, it is now being used for large distributed projects and teams. Agile methodologies like scrum and XP [13], [14] have been successfully implemented in many organizations now days that were first relying on standard traditional methodologies. These agile methodologies are discussed in literature.

Due to the distributed nature of the teams in GSD many challenges are being faced by the organizations. The aim of the research is to explore these challenges and suggest an appropriate agile tool/technique as a solution. We have implied an exploratory research method to explore the challenges/problems for GSD. These challenges are discussed briefly in literature section. This research can be a good starting point for future descriptive research on this topic. We have identified that most studies in literature have discussed these problems or their solutions in isolation without establishing common grounds to actually get to the problem at hand. Significance of agile techniques/tools in solving particular problem was missing mostly. More specifically, this research attempts to identify the importance of agile tools/techniques and practices to solve the challenges faced by distributed teams. No such agile framework for GSD has been noticed so far which focus on all major challenges/problems of GSD and their solution at once. The already existed frameworks [9], [20]-[22] have narrow scope and static. We

explored that more comprehensive work is required in order to propose a new generic and extensible framework. So, to fill this gap we proposed an agile intelligent and scalable framework that will help practitioners to identify the problems and will suggest an appropriate agile tool/technique to help solve the problem.

This paper consists of eight sections. Section I presents the introduction, Section II describes the research method used for the study, Section III discusses the literature review, Section IV presents the proposed framework, Section V contains the brief description of the proposed framework, Section VI presents the framework discussion, Section VII is conclusion and Section VIII gives the future direction of this research.

## II. RESEARCH METHOD

The research method for this study is Exploratory Research. We explored six commonly faced challenges of GSD. We proposed a solution to overcome GSD issues. According to [17] our research type is "Proposal of Solution". We also used the evaluation criteria mentioned in [17] for assessing the quality of our study.

Based upon the background and motivation following research questions has been identified:

- RQ1: What are the most commonly faced challenges for GSD?*  
*RQ2: What are the most commonly used solutions for GSD challenges?*  
*RQ3: What type of agile frameworks/techniques already exists to overcome GSD issues?*

## III. LITERATURE REVIEW

Software development is considered as complex and knowledge intensive activity [5]. Number of people, different teams and organizations are involved in the fulfillment of common organization goals. Negative stereotypes are associated with software projects for being late, over budget and low quality. Now in this era the evolution of teams to different geographic locations [19] presents more challenges for traditional project management. Commonly discussed challenges in literature are communication and coordination, temporal differences, knowledge sharing/group awareness, cultural differences, speed and communication tools. There are more than a few articles dealing with agile methods for distributed teams.

### A. Challenges for Distributed Teams

#### 1) Communication and Coordination

When teams are co-located they spend time on communicating with team members via planned meetings and informal discussions. Frequent interaction of the co-located team members solves many problems in the software development early. Communication and coordination become problematic when it's a matter of distance [18]. Software development at every phase requires proper communication and coordination among team members. As GSD suffer from a reduced level of communication due to distribution of teams.

Team members feel difficulty to communicate and do not properly know what other team members are doing. Linguistic, cultural and temporal differences are also barriers for communication across distributed teams [6], [16]. Software projects rapidly evolve due to significant and unforeseeable changes demanded by customers, product and standards. So communication is very important in organizations with rapidly changing environments and unstable projects [1]. For communication among team a highly secure communication network mediums and tools are required. The tools help in rapid sharing of information and coordination. There is variety of commonly used communication tools discussed in literature. Most of the teams use email, video conferencing, telephones, instant messaging. Group chats and remote access and control for communication. Still these communication tools are not enough to solve all the problems. As they can also suffer from problems related to delay of important information and connection problems. The lack of communication and less effective communication can cause failure to the project beyond measures. Some of the agile project management methods are being used successfully for distributed teams. Face to face communication is one of key principals of agile [9], [13]. Distributed scrum is used to manage the distributed teams at different geographic locations. Jeff Sutherland et al. [4] discussed three commonly practiced distributed scrum models: Isolated scrum, distributed scrum of scrum and totally integrated scrum. Distributed scrum of scrum is the suggested by Scrum Alliance for distributed teams [10]. The best practices for distributed scrum that address the problem of communication are daily scrum meeting of all the team members from different locations and daily meeting of product owner team. Teleconferencing, video conferencing, instant messaging and email were used as a tool for meetings [4], [10]-[12]. Communication needs are well facilitated by scrum meetings [14] because teams are working for common sprint goal.

#### 2) Temporal Differences

Temporal differences can decrease the bandwidth of communication [6]. Large organizations can avail the opportunity of software shift work. When teams are globally distributed the extensive level of time differences in different countries causes practical problems for project team [2], [12]. Due to the time difference across different countries there are few hours in a day when teams are available at their site at the same time. This difference in time restricts the meetings in working hours of the teams, which will result in lack of communication and eventually give rise to misunderstandings. Fewer meetings due to the overlapping working hours [9] make the team to take wrong assumptions which will lead to the failure of the project. In [4] integrated scrum was provided as a solution to SirsiDynix distributed setting. For scrum meetings they fixed a time feasible for their both locations Utah and St. Petersburg. It was necessary for the team to write and distribute the answer of the three scrum known questions before the start of the meeting [4]. This activity saved the time of joint teleconferencing meeting.

### 3) Knowledge Sharing/ Group Awareness

Knowledge sharing/group awareness is crucial for GSD [5]. The distributed teams due to communication barriers share less knowledge and information with each other. Team members have limited knowledge about what other team members are doing at different site. This leads to lack of motivation for team members to communicate. To keep the track of changes and other team member's tasks group awareness is very important. Team members should have enough information and knowledge about the project, should know the status of project and past history of project. This awareness about project will allow the team member to make reasonable assumptions about project. Mostly distributed teams use instant messaging applications which allow the person to know whether the team member is available or not. Including one can also see how much time the team member machine has left idle. Web based and shared calendars are also used to track where the person is and to whom they are in meeting with [1]. Knowledge and information sharing is always a challenge for distributed teams. In agile coordination and collaboration depend on communication. Agile distributed teams suffer from a lack of face to face communication and interaction. One of the challenges with agile teams is that they depend on human mind to share knowledge rather than documentation. The teams share knowledge and information through shared repositories, pair programming, daily scrum, online conferences, technical forum and electronic boards. The globally distributed agile teams used system, engineering, organizational and cartographic schools for knowledge sharing [5]. In distributed teams it is difficult to communicate all the priorities and context of clients. For solving this problem and sharing of knowledge regular travelling were scheduled, skype connections, informal updates by product owner and the news gazette [8] after every iteration were planned. Pair programming is an agile software development technique in which two team member work in pair. Pair programming is most commonly used by team for knowledge sharing. It is successful for collocated teams when it comes to globally distributed teams it becomes a challenge. Distributed teams use Skype screen share, TeamViewer and VPN services for pair programming. The author in [11] presented the idea of creating the knowledge centers which will contain the frequently asked question and the technical references.

### 4) Cultural Differences

People from different culture have different values and perceptions which cause misunderstandings in project team. Verbal communications are adversely affected by cultural differences. Communication problems are often aggravated by cultural differences [16]. Sometimes the low level of fluency in the language especially on telephone conferences becomes a problem for the team to communicate [2]. Concepts and terms are interpreted differently in different countries. Cultural differences can also result in misinterpretation of action which will lead to conflicts in team members. There is a different level of knowledge, skills, understanding of problem, terms and training of team among different cultures [12]. Enhanced

communication can be used to avoid the problems caused by cultural differences inside the project. Frequent project meetings are also considered as the effective mean for resolving the cultural issues. Jeff Sutherland et al. [8] discussed the cultural differences for the PUB project. Their Indian and Dutch team varies in culture. Similarly, both teams vary in the expression of their behaviors. The most important thing to deal with these issues is a good personal relationship throughout the project. This problem was solved by seeing team members at daily stand up meetings. Scrum master also engaged the team to directly communicate with each other [8].

### 5) Speed

Speed refers to the time to market. In distributed software development different teams are working around the globe for a common project. Distributed development offers 24-hour work which reduces the time to market. If there is an effective and frequent communication and information sharing among teams, then the time to market of the project reduces. Scrum is an agile methodology which is designed in a way to add transparency and clarity to the team members of project [6]. Agile introduces frequent meetings which enhance communication. The distributed scrum, and scrum of scrum work better with distributed teams. Time boxed short releases of the working product are the major focus of the agile methodologies [7]. In agile there are continuous small iterations, continuous integration of code and sprint review. So the product produced as a result of small iterations and reviews will be just right and can be released to market.

### 6) Communication Tools

Agile distributed teams cannot rely just on sticky notes on a board and burndown chart on a wall [12]. Distributed teams due to their setting at different geographic locations use different tools and environments for effective communication and interaction among team members. The tool related problems are considered as the problematic area in GSD. Different tools are being used by distributed teams for easy flow of communication. The choice of appropriate tool for distributed team is very important. Different teams prefer different modes of communication. For some visual presentation is effective and for some listening, verbal communication and discussion is helpful. Communication tools usually do not suffer from a steep learning curve, team easily adopt them and understand them [3]. Different teams use different tools for communication and information sharing. Wikis, smart boards and single code repository are some of the tools that were used successfully for global information sharing [8]. For agile distributed teams the use of message boards and wikis instead of emails can reduce the time [6]. Most commonly teams use Internet relay chat (IRC), Skype and other chat group chat tools for daily scrum meetings [5], [8]. Some general conclusion drawn from the studies is that allow the team to choose their communication tool themselves [3] with which they are comfortable. In spite of all the electronic mediums for communication, synchronous communication has its own importance. Team should

communicate using telephones and video conferences on regular basis to counter the cultural differences.

IV. FRAMEWORK

We presented an agile intelligent framework to overcome the GSD issues. We explored that there is a paucity of such frameworks in literature. This framework provides a new way of exploring not only GSD issues but also helps us in analyzing the problems existing in our project management methodologies as well. The framework comprises of three phases pre-processing inputs, processing body and post processing output. Pre-processing inputs are existing GSD

issues and solutions. Processing body shows the actual working of the framework and GSD issue resolution. Post processing output contains the solution of GSD issues in the form agile tools/techniques stored in a globally shared knowledgebase repository.

Our proposed framework analyzes GSD issues organizations are facing and suggests an appropriate solution to resolve such issues. The solutions will be stored in a Globally Shared Knowledgebase which can be used for future reference if such type of issues occurs again. Fig. 1 shows the framework.

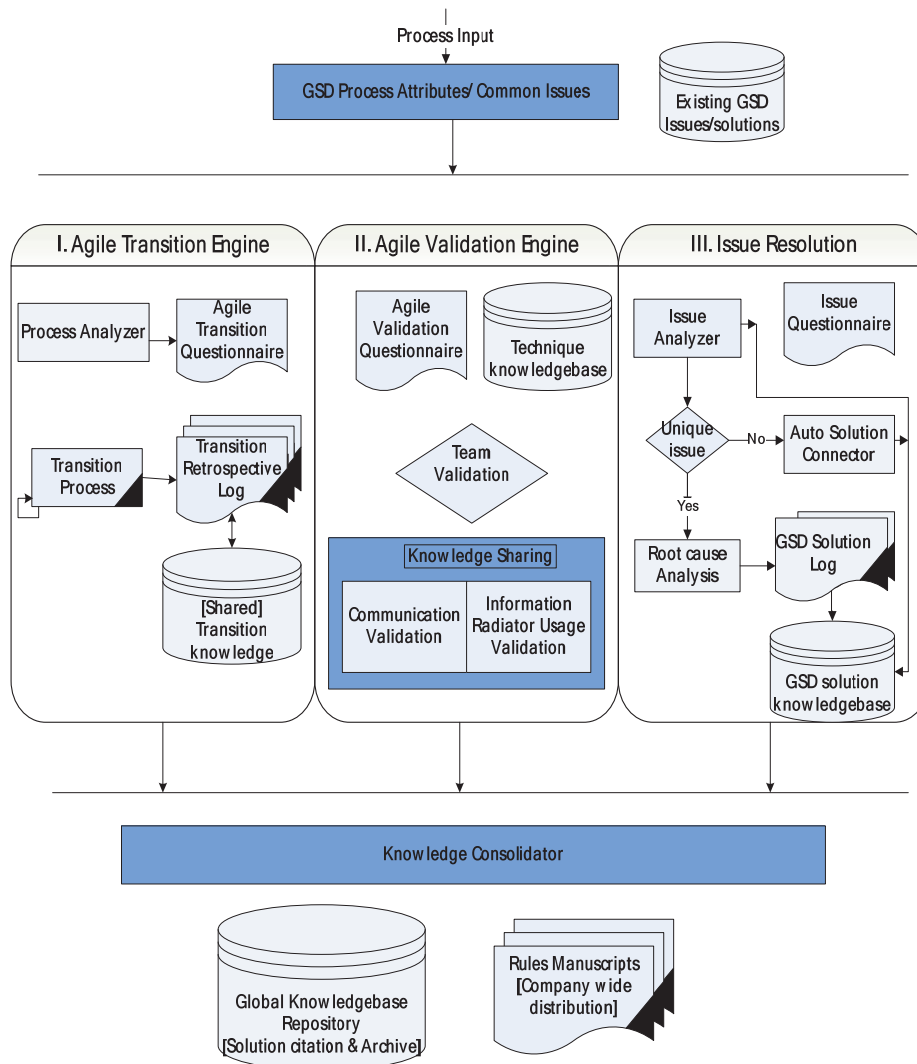


Fig. 1An Agile Intelligent Framework for GSD

V. FRAMEWORK DESCRIPTION

A. Pre-processing Inputs

- 1) Existing GSD Solutions/Issues

This database contains GSD issues which organization is

currently facing and the type of solutions they are using for resolving such issues.

- 2) GSD Process Attributes/Common Issues

Important attributes related to GSD and common issues faced will form the initial input to the framework. This module

also acts as a wrapper function to feed these attributes to the DB following a proper schema.

### *B. Processing Body*

#### 1) Agile Transition Engine

Agile transition engine is a module to be used by organizations not implementing agile. The main purpose of this module is to enable transition of organization's processes in an iterative and incremental manner. This module induces basic changes in current processes of the organizations implementing GSD to ensure agility. An organization experiencing Agile Transition Engine can effectively comply with the core values of agile.

##### *i. Process Analyzer*

Process Analyzer helps gather information about current processes being used by the organization including but not limited to team organization/structure, transparency, process data sharing, leanness, time boxed events/activities etc. Process Analyzer uses standard schema for storing process attributes acquired from transition questionnaire.

##### *ii. Agile Transition Questionnaire*

It is an XML based configuration questionnaire containing standard data/queries used to gather important insights of the processes being used by the candidate organizations. It contains different sets and question paths that help analyzer to query and validate the answers put forth by the relevant organization. Currently, it is at a very basic level with fewer paths and less annotations hence requires expert handling and answers. So, for few initial organizations we intend to get manual inputs gathered directly. But as the framework learns at later stages, we intend to improve it gradually to incorporate tricky questions, paths and better annotations to help both analyzer and amateur user to determine inputs in an effective and efficient manner.

##### *iii. Transition Process*

It is a core activity of transition engine module. It is an iterative process that helps feed and log changes to the organization's process to make it agile compliant. It acts on the data fed by process analyzer to propose/validate required changes. It uses transition retrospective log to judge ongoing changes. In the start, it is also fed manually from retrospective log but in future we intend to implement propositions to help it automatically judge the state of the organization through retrospective log to advice future course with forecasting ability.

##### *iv. Transition Retrospective Log*

Transition retrospective log records all the activities involved in the agile transition engine. It helps us to judge the transition process and helps us observe continuous improvements actions in this process in future.

##### *v. Shared Transition Knowledge*

It is a centralized knowledgebase containing successful transition data having dos and don'ts from a transition

retrospective log. The extent of knowledge fed is on the discretion of the particular organization running transition activities and is shared globally. This data can be used for furthering research in the transition domain and infer issue resolution with the help of ontologies. To enable organizations sharing knowledge, we need to introduce some perks or recognition that is yet to be researched on. But we strongly believe that like many other open source collaborative projects taking place, this knowledge sharing activity can also become a norm in the near future.

#### 2) Agile Validation Engine

Agile validation engine is a module to be used by the organization implementing agile. The purpose of this module is to validate current agile process the organization is following to ensure they are following agile with all major core principals.

##### *i. Agile Validation Questionnaire*

It is an XML based configuration questionnaire containing standard data/queries used to gather important insights of the agile processes being used by the candidate organizations. It helps us to validate the current agile process used by organization against core principals of agile to help identify an organization is implementing valid agile process or not.

##### *ii. Technique Knowledgebase*

Technique knowledgebase is a database containing the knowledge of all the agile techniques used by an organization whether it is scrum or XP etc. It helps agile validation engine to analyze and validate the techniques using the shared knowledgebase. Currently, our focus is on Scrum and XP, as they are the major agile techniques used so far.

##### *iii. Team Validation*

Team validation helps to validate the structure and nature of the team because it is an important aspect and one of the root causes of GSD issues. Team validation analyzes characteristics of the team concerning cross functional, self organizing, motivated and dedicated individuals with proper agile roles and responsibilities. This Validation can generate standard errors and warnings with the help of data imported from technique knowledgebase in this context to ensure team composition as per that particular agile technique's standards.

##### *iv. Knowledge Sharing*

Knowledge sharing contains two activities communication validation and information radiator validation. Communication validation validates how the organization's communication takes place, are there any meetings, workshops/seminars and some collaborative mediums etc.

Information radiator validation validates the information radiators an organization is using for the purpose of communication and knowledge sharing. We believe that properly used information radiator can help resolve many underlying problems related to communication and transparency as well. So, it should be managed properly.

### 3) Issue Resolution

Issue Resolution is a module which is used for resolving any issues that are creating impediments for GSD.

#### *i. Issue Questionnaire*

It is an XML based configuration questionnaire containing standard data/queries used to gather important insights about the issues being faced by the candidate organizations. It helps the issue analyzer to investigate the nature of the issue which will further help in issue resolution. Initially issue questionnaire contains common/basic questions related to the commonly faced issues but later on it will be updated upon the exploration of new unique issues.

#### *ii. Issue Analyzer*

The issue analyzer analyzes the current issues in order to reach the root cause of that particular issue/problem, fish bone diagram is created to get to the basics of problems. An appropriate solution in the form of agile tool/techniques is suggested to the particular issue.

#### *iii. Auto Solution Connector*

The issue analyzer analyzes the issue if the issue/problem is unique. If issue is unique then fishbone diagram is created and the solution is suggested. If the issue/problem is not unique and already exists in GSD solution knowledgebase then auto solution connector connects to the existing GSD Solution knowledgebase.

#### *iv. GSD Solution Log*

The GSD solution log contains all the solutions suggested by issue resolution module. It helps to judge how effective our solutions are in resolving a particular issue.

#### *v. GSD Solution Knowledgebase*

GSD solution knowledgebase is a database containing all the solutions to GSD issues. It will be updated when a new solution to the GSD issue is analyzed by the Issue analyzer. It is shared globally which helps an organization to get to the solution instantly when a particular type of issue occurs again.

### *C. Post-Processing Output*

#### 1) Knowledge Consolidator

The knowledge consolidator groups together all knowledge bases from three phases of the framework: Agile Transition Engine, Agile validation Engine, Issue Resolution. In this way, a shared Global Knowledgebase Repository is created containing all the information and knowledge of the issues. It is extensible and will be updated when a new problem arises.

#### 2) Rules Manuscripts (Company Wide Distribution)

These manuscripts are manually created based on the outcomes of knowledge consolidation. They contain the rules/manifest followed in the organization concerning agile implementation and GSD. It is the responsibility of Scrum masters to maintain these manuscripts and distribute them in their organization as required. We need to closely monitor this exercise to see its effectiveness and add/discard information

accordingly in these manuscripts based on our needs. Currently, we donot have any specific template for them but we intend to collaboratively design the templates of these manuscripts by gathering information from agile practitioners in future.

#### 3) Global Knowledgebase Repository [Solution Citation and Archive]

Global knowledgebase repository is a database containing the knowledge of three modules of framework. It contains the agile transition knowledge, agile validation knowledge and GSD solutions knowledge. It is fed by the knowledge consolidator. Due to its global nature this knowledgebase repository fosters effective knowledge sharing among different organizations. It contains the common issue/attributes and their best possible solutions suggested by the Issue resolution. Knowledgebase repository is globally shared and can help organizations get the solutions of their issues at first hand. It is extensible and will be updated when a new issue arises. Knowledgebase repository also contains solution citation and archive.

## VI. DISCUSSION

We used the framework proposed in [20] for developing a sample questionnaire for our framework. The sample questionnaire is developed in XML for only one issue of GSD that is Communication. It will help the issue analyzer to understand the root causes of communication issues. The sample questionnaire is presented in Fig. 2.

Other frameworks proposed in [9], [20]-[22] are generic, have narrow scope, static or for one particular issue, but our framework is generic, extensible and scalable it has ability to adopt other frameworks' features. Due to the shared global knowledgebase repository it provides us the facility to add the existing solutions in our knowledgebase repository. So if an organization is facing any GSD issue it can first check the global knowledgebase repository to get the solution from existing one. If the solution is implemented by the organization for particular issue that solution will be cited and added into the solution citation for later reference. For unique issues if solution doesnot exist in the knowledgebase then the issue resolution will help in finding the new solution. In this way our framework can give the access to the already existing solution and the new ones also.

## VII. CONCLUSION

GSD due to the distributed nature of the team offers more challenges and difficulties than the traditional project management. The objective of this research is to explore the commonly faced challenges by GSD to ascertain the best used agile techniques for managing GSD. In this article we have proposed an agile intelligent and scalable framework based on one of the most widely used and acceptable Agile methodology. This framework is beneficial for both agile and non agile organization. It comprises of three modules Agile transition engine, Agile validation engine and issue resolution. Agile transition engine is concerned with the transition of an

organization to agile in an iterative and incremental manner. If an organization is already implementing agile then agile validation engine helps to validate the current agile processes/techniques followed by the candidate organization. Issue resolution will help analyze the issue through root cause

analysis and suggest an appropriate solution in the form of an agile technique/tool. Along with that the framework establishes the knowledge sharing repositories both locally and globally to help practitioners/organizations to get to the solution instantly when a particular type of issue occurs again.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <GSDIssues>
3  <title> Issue Questionnaire </title>
4  <issue type="Communication">
5  <mainquestion> What type of communication issue you are facing? </mainquestion>
6  <mainanswer id="1" ans="Asynchronous">
7  <question que="Synchronized_work_hours">
8  <subquestion id="1" subque="Are you adjusting working hours between distributed sites to support distributed scrum meetings?">
9  <answer ans="Yes"><!--More Questions and answers nesting can be added here--></answer>
10 </subquestion>
11 <subquestion id="2" subque="Are you allowing distributed team members to attend meetings from home (e-g by phone)">
12 <answer ans="Yes"><!--More Questions and answers nesting can be added here--></answer>
13 </subquestion>
14 </question>
15 <question que="Reduce_scrum_meetings_length">
16 <subquestion id="1" subque="Are there any strict time boxed short meetings? (e-g 30 minutes planning meeting rather than longer)">
17 <answer ans="Yes"><!--More Questions and answers nesting can be added here--></answer>
18 </subquestion>
19 <subquestion id="2" subque="Are you doing any prior asynchronous work (e-g backlog preparation) before attending meetings?">
20 <answer ans="Yes"><!--More Questions and answers nesting can be added here--></answer>
21 </subquestion>
22 </question>
23 </mainanswer>
24 <mainanswer id="2" ans="Lack_of_group_awareness">
25 <question que="Team_Gathering">

```

Fig. 2 Sample Issue Questionnaire

## VIII. FUTURE WORK

In future we intend to continually improve the framework by getting feedback from various stakeholders using a collaborative approach. Also, we will be exploring the components of the framework further to include more implementation details. A survey will be conducted to seed the knowledgebase apart from the literature review we included as solutions to the common issues in GSD. As our framework is scalable, so the purpose of this survey will be to provide a thorough improved framework by analyzing problems and their solutions in a more practical manner. Components of Framework will be open sourced to give practitioners an opportunity to implement and researchers to further enhance it with improvements.

## ACKNOWLEDGMENT

We would like to express our gratitude to Riphah International University for providing us a platform for conducting this research. We are really thankful to Dr. Saad Naeem Zafar, Mr. Abdul Rauf and Mr. Salman Ahmad for their constant motivation and suggestions throughout the research.

## REFERENCES

- [1] Herbsleb, James D., and Audris Mockus. "An empirical study of speed and communication in globally distributed software development." *Software Engineering, IEEE Transactions on* 29, no. 6 (2003): 481-494.
- [2] Komi-Sirviö, Seija, and Maarit Tihinen. "Lessons learned by participants of distributed software development." *Knowledge and Process Management* 12, no. 2 (2005): 108-122.

- [3] Thissen, M. Rita, Jean M. Page, Madhavi C. Bharathi, and Toyia L. Austin. "Communication tools for distributed software development teams." In *Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce*, pp. 28-35. ACM, 2007.
- [4] Sutherland, Jeff, Anton Viktorov, Jack Blount, and Nikolai Puntikov. "Distributed scrum: Agile project management with outsourced development teams." In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, pp. 274a-274a. IEEE, 2007.
- [5] Razzak, Mohammad Abdur, and Rizwan Ahmed. "Knowledge sharing in distributed agile projects: Techniques, strategies and challenges." In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pp. 1431-1440. IEEE, 2014.
- [6] Phalnikar, Rashmi, V. S. Deshpande, and S. D. Joshi. "Applying agile principles for distributed software development." In *Advanced Computer Control, 2009. ICACC'09. International Conference on*, pp. 535-539. IEEE, 2009.
- [7] Sureshchandra, Kalpana, and Jagadish Shrinivasavadhani. "Adopting agile in distributed development." In *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on*, pp. 217-221. IEEE, 2008.
- [8] Sutherland, Jeff, Guido Schoonheim, Eelco Rustenburg, and Maurits Rijk. "Fully distributed scrum: The secret sauce for hyper productive offshored development teams." In *Agile, 2008. AGILE'08. Conference*, pp. 339-344. IEEE, 2008.
- [9] Nuevo, Eva del, Mario Piattini, and Francisco J. Pino. "Scrum-based methodology for distributed software development." In *Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on*, pp. 66-74. IEEE, 2011.
- [10] Lee, Seiyong, and Hwan-Seung Yong. "Distributed agile: project management in a global environment." *Empirical Software Engineering* 15, no. 2 (2010): 204-217.
- [11] Shrivastava, Supriya Vasudeva. "Distributed agile software development: A review." *arXiv preprint arXiv:1006.1955* (2010).
- [12] Smits, Hubert, and Guy Pshigoda. "Implementing scrum in a distributed software development organization." In *Agile Conference (AGILE), 2007*, pp. 371-375. IEEE, 2007.
- [13] Paasivaara, Maria, Sandra Durasiewicz, and Casper Lassenius. "Distributed agile development: Using Scrum in a large project." In

- Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on, pp. 87-95. IEEE, 2008.
- [14] Cristal, Mauricio, Daniel Wildt, and Rafael Prikladnicki. "Usage of Scrum practices within a global company." In Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on, pp. 222-226. IEEE, 2008.
- [15] Razavi, Abbas Moshref, and Rabiah Ahmad. "Agile development in large and distributed environments: A systematic literature review on organizational, managerial and cultural aspects." In Software Engineering Conference (MySEC), 2014 8th Malaysian, pp. 216-221. IEEE, 2014.
- [16] Marquardt, M. J., & Horvath, L. (2014). Global teams: How top multinationals span boundaries and cultures with high-speed teamwork. Nicholas Brealey Publishing.
- [17] Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements Engineering, 11(1), 102-107.
- [18] Herbsleb, J. D. (2007, May). Global software engineering: The future of socio-technical coordination. In 2007 Future of Software Engineering (pp. 188-198). IEEE Computer Society.
- [19] da Silva, F. Q., Costa, C., Franca, A. C. C., & Prikladnicki, R. (2010, August). Challenges and solutions in distributed software development project management: a systematic literature review. In Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on (pp. 87-96). IEEE.
- [20] Hossain, E., Babar, M. A., Paik, H. Y., & Verner, J. (2009, December). Risk identification and mitigation processes for using scrum in global software development: A conceptual framework. In Software Engineering Conference, 2009. APSEC'09. Asia-Pacific (pp. 457-464). IEEE.
- [21] Tariq, A., & Khan, A. A. (2012, October). Framework supporting team and project activities in Global Software Development (GSD). In Emerging Technologies (ICET), 2012 International Conference on (pp. 1-6). IEEE.
- [22] da Silva, F. Q., Costa, C., Franca, A. C. C., & Prikladnicki, R. (2010, August). Challenges and solutions in distributed software development project management: a systematic literature review. In Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on (pp. 87-96). IEEE.

research on one of the Formal specification language Behavioral Trees in 2015. She is currently a member of research group on Formal Methods in Riphah International University and she is conducting her MS research in the same area with International collaboration.

**Hafza Mehreen Fatima** is a MS Software Engineering student at Riphah International University, Islamabad Pakistan. She graduated as a PEC registered Software Engineer from Fatima Jinnah Women University, Rawalpindi Pakistan in 2013. She did internship at Network Operation Center PTCL Headoffice Islamabad in 2012. Her main research interests are in Agile software development and Requirements engineering. She is currently conducting her MS research in agile software development.

**Raja Asad Zaheer** (PMI-ACP) holds distinction in MS (Software Systems & Engineering) with a degree earned in 2009 from a reputed institution of Pakistan named as Muhammad Ali Jinnah University, Islamabad.

Asad is a PMI (Project Management Institute, USA) Member in a good standing and is Agile Certified Practitioner (ACP) from the same organization.

He is designated as operations manager in an Australian company named ReadHowYouWant headquartered in Sydney, Australia. Looking after its offshore operations in Islamabad, Pakistan with more than 50 full time in-house resources and virtually managed outsourced resources in Philippines, Malaysia, Bangladesh and India. He is also CEO of its sister concern Sofnix since Jan, 2014. Prior to this assignment, till Oct 2013 he worked in Sofnix since March, 2007 where he started off as a Sr. Software Engineer and progressed his career to a designation of Development manager. Following his passion for teaching and bridging industry-university gap, he often delivers lectures as visiting faculty member at graduate and post graduate levels.

He has diverse interests in research ranging from distributed & semantic computing to security and data grids. He remained research associate and coordinator for Centre of Distributed and Semantic Computing (M.A.J.U) from 2004-2006. Some of his publications include ("Intelligent Cache Management for Data Grids", ACWS Frontiers 2005, NewCastle Australia), ("SCAM: Semantic Caching Architecture for Efficient Content Matching over Data Grid", AWIC 2007, France), ("SECURITY IN LARGE SCALE DISTRIBUTED SYSTEMS (COMPUTATIONAL GRIDS): CRITIQUE AND SOLUTIONS", HONET 2005, Pakistan).

**Aisha Tanveer** is a MS Software Engineering student at Riphah International University, Islamabad Pakistan. She graduated as a PEC registered Software Engineer from Fatima Jinnah Women University, Rawalpindi Pakistan in 2013. She did internship at Network Operation Center PTCL Headoffice Islamabad in 2012. She worked as a Jr. Software developer in 360 Technologies, Islamabad in 2013. Her main research interests are in Formal Methods, Agile software development and Requirements engineering. She did