

## Effectiveness of Agile Practices in Global Software Development

Ritu Jain<sup>1\*</sup> and Ugrasen Suman<sup>2</sup>

<sup>1</sup>*Research Scholar, School of Computer Science & IT, Devi Ahilya University, Indore, M.P., India*

<sup>2</sup>*Professor, School of Computer Science & IT, Devi Ahilya University, Indore, M.P., India*

<sup>1</sup>*ritujain\_doc@rediffmail.com, <sup>2</sup>ugrasen123@yahoo.com*

### Abstract

*Global software development (GSD) has become a customary way of developing software whereas distributed agile software development is now gaining popularity. However, effectiveness of agile practices in GSD is a contradictory issue that needs to be explored in detail. The paper intends to investigate the effectiveness of agile practices in GSD. The research is conducted in two phases. Firstly, literature is explored to extract benefits and challenges of agile practices in GSD. Supporting practices, tools, and modifications to agile practices are also unearthed. Secondly, we have performed a survey to validate the outcome of literature review. We have identified 19 benefits and 14 challenges of distributed agile practices, and also discovered 16 supporting practices and various tools. The results of the literature review and survey both indicate that most of the agile practices can be adapted for distributed projects. Supplementary practices and tools further alleviate the incongruity between GSD and agile methods.*

**Keywords:** *global software development; distributed agile; literature review; industrial survey; scrum; extreme programming*

### 1. Introduction

Global software development has now become a prevailing trend in the software industry. The prospective benefits of GSD that lure software companies are decreased development cost, reduced cycle time, accessibility to remote experts, and proximity to local market [1-2]. However, geographical, temporal, socio-cultural, and organizational distances are the barriers to achieve above benefits and in turn severely affect the communication, coordination, and collaboration processes [1]. There are several challenges that GSD practitioners face during different phases of software development such as decreased visibility of project activities, low frequency of communication between remote counterparts, insufficient shared understanding among distributed team members, delayed feedback, unawareness regarding remote members, slow resolution of impediments, ineffective monitoring of progress, reduced cohesiveness of team, and knowledge management difficulties [3]. These problems can be alleviated through a socio-technical solution such as appliance of agile methods, generous usage of supporting practices and tools that can improve communication, coordination and collaboration during distributed development [S1-S4].

Agile practices provide high flexibility, increased productivity, ample customer satisfaction, and fast development. These practices also enhance communication, and collaboration; decrease defect rates, and manage late changes in requirements [4]. Agile methods can be useful in distributed development but there are two

---

\* Corresponding Author

contradictory opinions for distributed agile software development (DASD). Proponents believe that agile methods are more suitable to GSD as compared to the traditional process [S2, S4]. Agile practices can reduce the effect of various distances in distributed software development projects [S1, S5-S8]. The blend can alleviate major problems of GSD related to offshore visibility, monitoring, and communication [S1, S7, S9]. While opponents of DASD advocate that agile methodology and GSD are contradictory to each other, as agile methods were created to develop software in collocated environment with small teams [5]. The prerequisite for successful application of agile methods are frequent communication, and close collaboration among team members, which are lacking in GSD [S25].

The number of case studies and experience reports, which have successfully incorporated agile practices in GSD are far more than unsuccessful cases. Thus, the literature suggests that agile methods can be applied to GSD [6, S3, S10]. However, thorough analysis of benefits and challenges of applying agile practices is still deficient [6]. Several studies also suggest that agile practices need to be customized for distributed projects [6, S2]. Though, researchers have explored Scrum practices using few case projects, tailoring of XP as well as Scrum practices need more empirical investigation for application in GSD. In order to support these findings, we have investigated the benefits and challenges of applying agile in GSD.

This paper is organized as follows: Section 2 summarizes background and related work in distributed agile software development. Section 3 describes research design and methodology used in the study. Section 4 describes results of literature review followed by Section 5 which presents results of industrial survey. The results presented in Section 4-5 are discussed in Section 7. Section 6 provides a brief outline of the supporting tools used in GSD. Lastly, Section 8 provides concluding remarks, and future work.

## 2. Background and Related Work

Distributed development is speedily embraced by software industry to reduce the development time and cost; since, distributed team can be scaled swiftly using easily available cheaper experts from lower economy countries. Agile methods can smoothly handle volatile requirements, promise higher customer satisfaction, deliver products timely, increase productivity, and; thus provide companies a competitive edge in business environment. Due to the high failure rate during execution of distributed projects using sequential processes; practitioners have started experimenting agile methods in GSD. But, as agile methods were primarily designed for collocated development, their direct application in GSD could lead to discrepancies in software development. Distribution can further deteriorate communication and collaboration among remote team members [S2].

Jalali *et. al.* have conducted systematic literature review to explore current status of agile practices in GSE and have concluded the scarcity of empirical studies in this domain [6]. Thus, it is observed that there is a need for evaluation research in this domain. Paasivaara *et. al.* have reported the findings of four case studies in which seven Scrum practices were successfully adapted for small and mid-sized GSD projects [S3, S9]. Five supporting practices were also used in projects to reduce the effect of distribution. They have also investigated application of Scrum in a large GSD project [S16]. Project team felt that Scrum can be successfully applied in GSD projects, and could solve the major problem of communication faced in GSD setting. Bose has explored solution strategies that 12 published case studies had employed to overcome aforementioned challenges [4]. Adherence of agile principles in the cases studies was also unearthed. Hossain *et. al.* have investigated the challenges of applying Scrum in GSD setting [S10]. It was observed that more empirical studies on application and adaption of Scrum in GSD are required.

Existing literature reviews, case studies, and experience reports indicate that agile methods can be applied in GSD. But, most of the researchers have performed empirical

investigation on Scrum and XP practices by means of only few case studies. In addition, previous studies have not sufficiently explored supporting practices as well as tools. Therefore, we have investigated 39 distributed agile projects specified in literature, followed by validation research through industrial survey. Through this research, practitioners can find pros and cons of agile practices. They can review modifications that other practitioners have applied to agile practices in GSD. They can identify supporting practices and tools needed to supplement with adaptive agile practices.

### 3. Research Design and Methodology

The aim of our study is to investigate various agile practices, specifically XP and Scrum practices in GSD and to discover supplementary practices and tools needed to form the blend of agile and distributed development successful. Therefore, research questions specified in Section 3.1 are framed and research design mentioned in Section 3.2 is planned.

#### 3.1. Research Questions

Following research questions (RQ) have been formulated to achieve the aforesaid aim of research:

RQ1: Which agile practices are used in GSD projects?

RQ2: What are the benefits gained by practitioners, when agile practices are applied in GSD projects?

RQ3: What are the challenges faced by practitioners, when agile practices are applied in GSD projects?

RQ4: What modifications are done in agile practices to make them suitable to GSD projects?

RQ5: What supporting practices are used by practitioners to supplement agile practices in GSD projects?

RQ6: What supporting tools are used by practitioners to supplement agile practices in GSD projects?

#### 3.2. Research Design

In order to answer the above questions, research is performed in two steps. Firstly, literature was explored and results of literature review were used to design questionnaire. Subsequently, we have performed an industrial survey to validate the outcomes of literature review.

**3.2.1. Literature Review:** We have examined the articles related to the application of agile in distributed projects. Databases explored for conducting review are IEEE Xplore, ACM Digital library, Springer Link, and Elsevier ScienceDirect. Studies that were available online, written in English, published in between 2001 to 2015 were extracted. The 40 articles were selected for literature review related to the distributed agile practices. Selected studies consists of 11 case study papers, 14 experience report papers, 2 grounded theory papers, 4 literature reviews and 9 theoretical papers.

**3.2.2. Formulation of Questionnaire:** The results of literature review were used to formulate the questionnaire. The questionnaire was designed help of two experienced software practitioners, and a veteran researcher of this domain. It consists of eight segments; namely, introduction, personal details, project specific details, usage of agile

practices, benefits achieved by agile practices, challenges faced in applying agile practices, supporting practices employed, and supporting tools used.

The introduction section explains the intention of questionnaire and the research. The personal detail section contains questions related to experience in software industry, experience of distributed agile projects, and designation. Project specific detail contains questions about collaboration form of distributed project, number of distributed locations, onshore and offshore locations, total members in the project, development process used, duration, and domain of the project. The usage frequency of various agile practices is measured in Section 4. The benefits of distributed agile practices extracted from the literature are specified in Section 5 of the questionnaire. Challenges that were extracted from literature were specified in Section 6. Respondents have to choose one of the options among (strongly agree, agree, neither agree nor disagree, disagree, strongly disagree), to signify his views regarding benefits, and challenges of distributed agile software development. Section 7 lists supporting practices extracted from literature. Responses for Section 4 and Section 7 are indicated by choosing one of the five options among (frequently used, moderately used, rarely used, not used, and don't know). In supporting tools section, respondents had to fill in the tools they have used in their distributed agile project. The questionnaire was constructed in Google forms.

**3.2.3. Industrial Survey:** The survey was conducted from 17 June 2015 to 19 July 2015. The questionnaire was mailed to 105 software practitioners across the world using purposive sampling method. The practitioners chosen for the survey were professionals working in global software development projects. We had requested practitioners to fill it only if they have applied agile practices in their GSD projects. We received 37 responses out of which 29 were valid.

## 4. Results of Literature Review

The literature was reviewed to explore the effectiveness of agile practices in GSD. Contextual factors such as form of collaboration, total members in the project, number of locations, development process used, domain of project, onshore locations and offshore locations, were considered for the evaluation of literature review. These factors reveal several interesting facts.

- Most of the projects discussed in case studies and experience reports have team size up to 20. This shows that either distributed agile methods are suitable only for small and medium projects, or there is a lack of application, or reporting of distributed agile methods in large projects.
- Most of the projects were distributed among two or three sites. This exhibits that most companies distribute development among two or three sites as the complexity related to communication, coordination and control increases with distribution.
- There are almost equal number of offshore outsourcing and offshore insourcing projects. This indicates that both collaboration forms are well accepted.
- 58% of projects mentioned in case studies and experience reports have used Scrum, this indicates that Scrum is the most popular agile method.
- The year wise distribution of studies shows that the interest of practitioners and researchers has increased after 2005.
- In literature, United State is the most recurrent onshore location and the most popular offshore location is India.
- Agile methods were applied to distributed projects belonging to diverse domains.

Outcome of literature review is described in two subsections. Section 4.1 discusses benefits, challenges, and modifications of agile practices for GSD. Section 4.2 describes supporting practices required to implement GSD.

#### 4.1. Benefits, Challenges, and Modification of Agile Practices for GSD

We have explored the benefits, challenges and modifications performed for implementing agile practices in GSD, which are presented in following subsections. Consolidated list of usage, benefits, and challenges of distributed agile practices is also prepared after thorough analysis and represented in Tables 1-3.

**4.1.1. Pair Programming:** Pair programming allow two programmers to write code collectively at one machine [7]. Remote developers often synchronize their work hours willingly and spent ample time with each other [S2]. However, remote pair programming is inappropriate for easy and well understood functionalities [S2]. Therefore, it was practised only when implementation seems complicated [S11]. In a project, practitioners have supplemented or substituted it by code reviews [S12].

**4.1.2. Planning Game:** In the beginning of every iteration managers, developers, and customers assemble to uncover, estimate, and prioritize requirements for the iteration [8]. Due to time zone difference, only development lead and project management personnel participated in planning meetings via speakerphone [S13].

**4.1.3. Continuous Integration:** Developers integrate new code into code base frequently, update tests regularly and run all tests to verify that new code will not break the code base, otherwise new code is rejected [7, 8]. In GSD, it is difficult to continuously integrate software products which require time consuming regression testing and lengthy code release processes [S11]. Practitioners have applied continuous integration in GSD by maintaining central code repository [S14]. Offshore team integrated their modifications directly into main code branch [S15]. Developers integrated their code daily into centralized version control system [S13, S16, S17]. In every 8-24 hours complete product was built and regression test suite was executed [S13, S16, S18]. In case of build failure, the responsible team has to debug errors and rebuild it [S16]. For implementing daily builds, check-in of code was not permitted after a specific time of the day [S19].

**4.1.4. Test Driven Development:** In test driven development (TDD), developers implement acceptance tests for the code before developing the code itself [7, 8]. It is a lightweight and unambiguous method to precisely specify the requirements [S20, S21]. It solves the problem of linguistic difference as, test suites can be inferred in the same way by different distributed teams. It helps in keeping distributed development teams in sync with requirements and preserving overall consistency by expressing changes in requirements or module behaviour in form of modified test cases. A developer who wants to use the functionality of a module developed by remote developer can understand the behaviour of a module, if its interface is supplemented with test cases. This shared and improved understanding will reduce problems in integration testing. It is easier for a developer to comprehend requirements in the form of test cases as compared to other less familiar specification methods such as UML [S21]. Acceptance tests written by managers and customers help to improve the understanding of developers about the functionality to be developed from clients' perspective [S2, S4, S20]. Test suites can be reused and extended by adding new test cases as required [S21]. However, TDD cannot be practised in projects where quality assurance teams belong to separate departments [S11].

In distributed development, architect and testers create few module-wise test suites and interface description and provide them to distributed teams as a part of high level design. These artefacts provided syntactic and semantic details of modules to be developed. Test suites are stored in a central repository to reduce the need for cross-site communication [S21]. Distributed developers publish their work intention through functional and

acceptance tests to avoid collisions and integration races [S20]. Unit tests can be used to converse architecture and project status [S22].

**4.1.5. User Story:** User story is a short high level specification of requirements by customer. It is written on a story card [8]. It acts as a communication medium between developers and users [S23]. However, it is complicated, and time consuming to synchronize and access user story, along with use case diagram, if they are stored in different formats [S24]. Also, it is hard to infer story cards stored in text document [S15]. In DASD projects, user story cards, supplemented with use case diagrams can be managed in globally accessible backlogs [S12, S24]. They can be associated with a crisp documentation of 1 or 2 pages comprising short explanation of the feature, functional test cases, probable impact on other modules, and business driver context [S13, S25]. User stories can be of varying length depending on its complexity. The confusions related to them can be resolved in daily scrums, by email or instant messaging [S17].

**4.1.6. Collective Ownership:** Collective ownership implies that all developers own entire code. Any developer can alter any code in the system at any time [8]. Some practitioners feel that this practice creates confusion, and varied responsibilities [S14]. In a distributed project, photo-chart of the whole project team, burndown charts, and story boards were replicated to facilitate collective ownership among remote teams [S19]. In another project, a similar office set-up at all locations reflects the sense of working together [S26].

**4.1.7. Coding Standards:** In the beginning of the project, rules and regulations for coding are decided. Developers follow these coding rules during the development [7, 8]. It supports code uniformity and helps in inspection of code [S2]. It improves readability of code developed by distributed teams [S11]. It helps developers to communicate in terms of code and maintains consistency [S27].

**4.1.8. Onsite Customer and Proxy Customer:** Customer representative remains onsite to answer queries of development team, clarify doubts and confusion regarding requirements, and carry out performance tests [8]. In distributed development, business analysts collect and discuss requirements with customers. He then works as a proxy customer for offshore team, solves their queries [S16, S27]. However, it is difficult to employ this concept for the products, which do not have direct customers [S2].

**4.1.9. Metaphor:** It is a story that describes how system works and is shared between customers and developers. It guides development of the system [7, 8]. It facilitates communication between different stakeholders of project [S27].

**4.1.10. Refactoring:** Refactoring is a technique to restructure the system by reducing complexity and duplication to increase performance, maintainability and flexibility [7, 8]. It helps in removing bugs supports and communication among the developers [S2, S27].

**4.1.11. Simple Design:** Developers are advised to design simple solutions that satisfy current requirements and continuously evolve it according to the need [7, 8]. Simple design enhances communication between distributed team members [S27].

**4.1.12. 40 Hours Week:** Team members should work not more than 40 hours in a week. Two consecutive overtime weeks are forbidden [7, 8]. It is considered infeasible in GSD as offshore team usually work 50-60 hours per week to synchronize time zones [S2, S5].

**4.1.13. Sprint:** It is an iterative cycle used by Scrum team to develop or enhance functionality [7]. The short sprints align members at all locations towards goal [S1, S22,

S23, S28]. However, some team members found it difficult to divide task into smaller sized components for sprints [S22]. Frequent changes at the end of the sprint cause rework, frustration, and can delay its completion [S18, S24].

However, GSD practitioners have applied sprint with few modifications such as beginning sprints were dedicated for settling crucial requirements, upfront designs, and constructing high level architecture [S4, S6]. Well understood requirements were developed first to enhance learning and confidence of developers [S25]. Some practitioners had used flexible short cycles rather than time-boxed development while other practitioners accumulated and packaged code of several sprints and deliver it into larger releases [S4, S11]. In a project, work was distributed according to features and not according to specialization, thus sprint can be executed at a single location [S6]. Lengths of some sprints were varied to accommodate holidays of different countries [S3, S16, S28]. Some project teams had implemented short sprints (2-weeks) to handle frequently changing requirements [S11, S22, S28], while in other projects, initially the sprints were 4-6 weeks which were reduced to 2-1.5 weeks [S3, S14, S28]. Thus, the length of sprints varied from 1.5 week to 4 weeks in distributed projects [S1, S11, S13, S16-S18, S28].

**4.1.14. Daily Scrums or Stand Up Meeting:** Short daily meeting of about 15 minutes in which every team members discuss three questions about the work he has completed yesterday, his goal for today, obstacles he is facing [8]. It enforces team members to share their ideas and critiques, commit publicly, and respond timely to any changes in the project [S3, S4, S11, S29]. It prevents speculative design and coding by distributed members [S19, S30]. Daily scrums resolved confusions related to requirement document [S17]. In spite of being the most popular agile practice, daily scrums face some challenges. It is difficult to answer scrum questions daily, if backlog is not properly partitioned into small tasks [S22]. Scrum meetings often lost focus due to lengthy technical discussions [S24].

Distributed scrum meetings were arranged on every working day during common working hours with the help of various tools [S4, S11, S13, S16, S17, S19, S20, S22, S28, S29, S31]. In projects with multiple distributed locations, scrum meetings are organized one after the other with offshore locations, so that scrum masters and product owners could attend meetings of more than one location [S3, S16]. It was observed that distributed scrum meetings were more successful in case of small teams. Therefore, a team had restricted the number of participants to utmost 8 members [S31]. An offshore team can assign a representative who could participate in daily onshore scrum meetings and can pass the minutes to offshore members [S12, S28]. Different sites can conduct their separate daily stand-up meetings, in addition to daily distributed Scrum meetings [S22, S31]. During execution of pilot distributed agile projects, outcome of scrum meetings could be formally documented to record agreements and decisions [S24]. These short meetings were used to schedule longer meetings between team members [S3, S16, S32]. Distributed scrums were considered more beneficial than site specific daily meetings [S3]. In some GSD projects, scrum meeting lasted for at most 15 minutes [S3, S16, S28] whereas in others it lasted for 15 to 45 minutes [S13, S30].

**4.1.15. Sprint Planning Meeting:** Prior to each sprint, sprint planning meeting is conducted to decide about the functionalities that is to be developed in the sprint. These functionalities are recorded in sprint backlog [7]. These meetings provide an opportunity for remote members to participate, clarify their doubts, build a collective understanding, and develop awareness of tasks [S3, S7]. However, it is difficult to conduct activities such as effort estimation in a distributed manner, which needs rigorous discussion [S3]. Sprint planning meeting is organized in two or three phases in GSD projects. At the end of the meeting, sprint backlog is prepared [S3, S16, S18, S28]. In other projects, distributed members participated by publishing post-it notes on the web page [S2].

**4.1.16. Sprint Review or Sprint Demo:** At the end of sprint, scrum master and scrum team demonstrate the features implemented in the sprint to management, customers, users, and product manager. In this meeting, sprint outcome is assessed against sprint goals [7]. Sprint review meetings disclose the gaps between requirements and implementation [S22]. It allows team to discover their status in project [S22]. Dependencies between requirements can be effectively shared through it [S12, S33]. However, it is difficult to evaluate a sprint if its end is shifted due to some reason [S3].

There are different ways in which sprint review meetings were conducted in DASD projects. Distributed review meetings were conducted with the help of different distributed tools [S1, S3, S9, S14, S16, S22, S28, S34]. In some projects, the management or onshore team presented demo to customers and provided feedback to development team [S10, S28, S35] whereas, in another project, project manager/ product owner acted as a proxy customer, verified the completed code after sprint [S1, S28]. Separate sprint demos were arranged at each location and scrum masters attend the overall project meetings [S3, S18]. Instead of having review meetings, the team released accomplished code to onshore quality assurance and acceptance testing team for review and feedback [S1, S28]. Face-to-face demos could be conducted during visit [S9].

**4.1.17. Retrospective Meeting:** It is a meeting after each sprint in which team analyzes “what helped them and what does not” and discovers improvement possibilities for the next sprint [7-8]. It facilitates team to explore good practices and lessons from previous iteration; and finds hidden conflicts [S16, S23]. Contributions of local as well as remote members should be recognized in these meetings [S9]. DASD projects have adapted retrospective meetings in various ways. Local sub-teams arranged separate retrospectives whose results were posted on wiki [S1, S28]. Distributed retrospective meetings were attended by product owner, scrum master and team. It spanned for at most one hour [S16]. In a project, retrospectives were conducted after every sprint initially. But, when the operation was smoothened, they were organized alternately or issues were resolved in other meetings [S1, S28]. In few projects, retrospectives were combined with sprint review of current sprint and/or planning of next sprint [S3, S18, S28]. In a project, the team had arranged retrospective sessions at the start and end of the iteration [S22].

**4.1.18. Product and Sprint Backlog:** Product backlog is a prioritized list of all tasks to be accomplished in the project. The list continuously changes according to new or modified functionalities during the entire project life cycle. Sprint backlog contains prioritized subset of product backlog functionalities that team decided to accomplish during the sprint [7]. In GSD projects, backlogs were maintained centrally and can be accessed globally [S1]. They help to form an unambiguous, shared view of project and can be used to track progress of the project [S3, S14, S18, S23]. Developers can use them to regularly update the completion status of task allocated to them [S28]. The responsibilities for updating backlog are occasionally unclear [S3].

**4.1.19. Burndown Chart:** It is a chart that represent amount of work to be completed across time. It is used to visualize the progress of project against time [8]. It can be used to predicate the impact and cost of addition of new work during the sprint, and whether the sprint goal will be achieved or not [S22]. It also helps to establish performance based trust between distributed members [S23].

**4.1.20. Scrum of Scrums Meeting:** Scrum of Scrums meeting (SoS) is a technique to extend Scrum to a large team in which the team is divided into sub-teams of 5-10 members. Each sub-team conduct its own daily scrum in which one member is designated as a representative. These representatives of all sub-teams participate in daily meetings

called Scrum of Scrums. These meetings are used to adjust sprint plan, and update product backlog, and facilitate suggestions [S1, S23, S28, S30].

In a GSD project, all Scrum masters and team representative participate in weekly SoS meeting. The representative had to answer three scrum questions and additional questions regarding laying or planning to lay impediments in other team's way [S3, S16]. All scrum masters conducted proxy SoS meetings weekly via Skype [S1, S28]. A GSD project did not use SoS, but small sprint length provided facility to solve cross location problems and dependencies [S1, S28]. A meeting during the mid of sprint operated as a SoS meeting. Additionally, two team members from each location arranged telephonic meeting twice a week [S28]. In another project, Scrum master, product owner, and occasionally, few customers arranged distributed meetings twice a week [S28, S32]. The agenda of the meeting was proposed in advance on wiki space to aware all team members. Minutes of meeting were also stored in wiki space [S32]. All Scrum Masters resided at a single location and conducted SoS weekly [S17].

The usage of various distributed agile practices is mined through literature review to determine the popularity of various agile practices. Table 1 illustrates these agile practices, literature references, and literature support count (LSC). Literature references specify the study ID in which the agile practice is supported. LSC was calculated by counting the number of references associated with a particular practice. Daily Scrums and sprints are agile practices that are discussed most in the literature, whereas 40-hours week and metaphor are least referred practices.

**Table 1. Usage of Agile Practices in GSD**

ID	Agile Practice	Literature References	LSC
PP	Pair programming	S11, S20, S36, S30, S2, S15, S12, S27, S32	9
PG	Planning game	S29, S27	2
CI	Continuous integration	S13-S19, S27, S37	9
TDD	Test driven development	S2, S11, S20, S21	4
US	User story	S12, S13, S17, S23-S27, S37	9
CO	Collective ownership	S2, S11, S19, S26, S27	5
CS	Coding standards	S2, S11, S27	3
OPC	Onsite customer and Proxy customer	S13, S27, S28	3
ME	Metaphor	S27	1
RE	Refactoring	S2, S27	2
SD	Simple design	S2, S27	2
FHW	40- hours week	NIL	0
ST	Sprint	S1, S3, S4, S6, S11, S13, S14, S16-S18, S22-S24, S26-S28, S37	17
DS	Daily scrums/ Stand up meetings	S1-S4, S11-S13, S16, S17, S19, S20, S22-S24, S26, S28-S31, S36, S37	21
SP	Sprint planning	S1-S3, S16, S18, S26, S28, S37	8
SR	Sprint review or sprint demos	S3, S9-S11, S14, S16, S18, S22, S23, S26, S28, S34, S35, S37	14
RE	Retrospectives	S3, S9, S11, S16, S18, S22-S24, S26, S28, S37	11
BAC	Product and sprint backlog	S3, S16, S18, S22, S23, S28, S32	7
BC	Burndown charts	S22, S23, S28	3
SOS	SoS meetings	S3, S16, S17, S23, S28, S30, S32	7

We have identified 19 common benefits of distributed agile practices which are listed in Table 2. The IDs are assigned to benefits for further reference. The benefit "Agile practices focus on solving issues, misunderstanding, and impediments timely" coded as B4 has scored maximum LSC. Daily Scrums and sprint are practices that help to achieve 14 benefits out of 19 benefits.

**Table 2. Benefits of Applying Agile Practices in GSD**

ID	Benefits	ID of Agile practices	Literature References	LSC
B1	Improve quality of communication between the sites	TDD, US, SPR, DS, SP	S2, S3, S7, S11, S16, S21-S23, S27	9
B2	Enhance frequent communication between sites	PP, TDD, SPR, DS, SP, SR, RE, BAC, SOS	S1, S3, S12, S17, S20, S22, S23, S27, S29, S30, S32, S33	12
B3	Reduce communication delay	SPR, DS	S3, S12, S27, S29, S33	5
B4	Focus on solving issues, misunderstanding, and impediments timely	CI, TDD, SPR, DS, SP, SR, RE, BAC, SOS	S1-S4, S7, S11-S13, S16-S21, S28-S30, S33, S36, S38	20
B5	Reduce knowledge management difficulties	PP, TDD, DS	S21-S23, S30, S36	5
B6	Induce more one-to-one communication between the sites	DS	S3, S32	2
B7	Enhance visibility of project activities	SPR, DS, SP, SR, BAC, SOS	S2-S4, S7, S12, S16, S22, S23, S28, S30, S34, S38	12
B8	Aid in evaluating and measuring the progress of the project	CI, DS, SP, SR, BAC, BDC, SOS	S2-S4, S7, S13, S16, S22, S23, S28, S30, S39	11
B9	Problems of the project can be more easily noticed at early stages	CI, TDD, SPR, DS, SR, RE, SOS	S3, S4, S7, S16-S18, S21, S22, S28, S34, S39, S40	12
B10	Reduce risks of outsourcing through continuous feedback and demos	PP, CI, SPR, SR, BDC, SOS	S3, S9, S12, S16-S18, S22, S23, S27, S28, S33, S39	12
B11	Help in completing project within schedule	SPR, SR	S3, S7, S11	3
B12	Increase overall productivity of team members	CI, SPR, DS, SR, RE	S11, S18, S19	3
B13	Improve quality of software product	PP, CI	S2, S18, S39	3
B14	Improve stakeholder collaboration	PP, SPR, DS, SP, SOS	S2, S3, S17, S18, S22, S28, S30	7
B15	Improve trust among stakeholders	CI, SPR, DS, SR, BDC, SOS	S23, S29, S30, S33, S39	5
B16	Increase team awareness and cohesion	PP, SPR, DS, SP	S2, S7, S16, S18, S22, S23, S28	7
B17	Improve shared understanding among distributed developers	PP, TDD, DS, SP, SR, BAC	S2-S4, S7, S12, S14, S16, S17, S20, S22, S33	11
B18	Increase motivation of team members	PP, CI, SPR, RE, SOS	S2, S9, S11, S18, S30, S39	6
B19	Enhance client satisfaction	SPR	S18, S33	2

Through literature review, we have identified 14 common challenges faced by DASD practitioners, which are listed in Table 3. It also specifies the LSC for each challenge. “Slow, unreliable and ineffective communication networks” and “impossible to use several tools due to narrow bandwidth” are issues discussed in highest number of studies. LSC of Table 3 is quite less as compared to Table 2; this shows that either agile practices offer ample benefits as compared to challenges or authors have not specified enough challenges that they come across while applying distributed agile practices.

**Table 3. Challenges of Agile Practices in GSD**

ID	Challenges	Agile Practices	Literature References	LSC
C1	Agile methods rely on informal communication which is difficult in GSD		S5, S24, S34, S36	4
C2	Due to time zone difference, it is difficult to arrange regular stand up meetings	DS	S3, S22, S24	3
C3	Insufficient tools for conducting distributed meetings	SP	S13, S15, S16	3
C4	Sound quality is not good enough to discuss issues	DS	S3, S31	2
C5	During teleconference, it is difficult to recognize speakers	SP	S3	1

C6	Slow, unreliable, and ineffective communication networks hamper communication in agile GSD	PP, DS, SR, RE	S3, S9, S10, S16, S22, S31, S32	7
C7	Language differences may cause silence of some participants in distributed meetings	DS	S31	1
C8	Due to time zone difference, it is difficult to arrange long meetings	SP, SR	S3, S10, S16, S22	4
C9	Impossible to use several tools due to narrow bandwidth	PG, DS, SP, SR, RE	S3, S9, S10, S15, S16, S22, S36	7
C10	Cultural values of some team members do not allow them to discuss their problems openly	DS, SP, SR, RE	S3, S10, S16, S36	4
C11	Different holiday pattern in different countries causes synchronization challenges while executing sprints	SPR	S3	1
C12	Agile methods rely on tacit knowledge whereas it is difficult to manage tacit knowledge in GSD		S5	1
C13	Agile methods support low documentation, whereas comprehensive documentation is required in GSD		S4, S5, S8, S12, S19	5
C14	Agile methods believe in continuous planning whereas GSD supports up front planning		S6, S24	2

#### 4.2. Supporting Practices needed for GSD

The supporting practices along with benefits and literature references are specified in Table 4. Supporting practices assist many of the distributed agile practices. Usage of multiple communication tools is largely supported by the literature.

**Table 4. Supporting Practices**

ID	Supporting Practices	Benefits	References	LSC
SP1	Relocation of a team member for the project duration	• Alleviates communication deficit, reduces socio-cultural distance. Relocated person acts as a technical as well as cultural liaison.	S6, S13, S16	3
SP2	Collocate the team during crucial phases	• Create shared understanding of the project, develop rapport and trust among team.	S6, S12, S22, S23, S27, S29	6
SP3	Synchronization of work hours	• Encourage synchronous communication, quick feedback, and facilitate distributed meetings and remote pair programming.	S1-S4, S10, S13, S15, S16, S19, S20, S28, S30, S36	13
SP4	Frequent visits/ Exchanging ambassadors/ Rotating gurus	• Enhances cross-site communication, collaboration, team cohesion, trust, and reduces socio-cultural distance. Supports transfer of domain knowledge.	S4, S6, S16, S19, S20, S26, S29-S31, S36, S38	11
SP5	Multiple communication tools	• Promotes communication, coordination, collaboration, and aids knowledge management. • Videoconferencing allowed members to see facial expressions. It is used to start and evaluate sprint, and discuss crucial problems.	S1, S4, S6, S8, S11-S14, S16, S19, S20, S26, S27, S29, S31, S32, S36-S38, S40	20
SP6	Strict communication plan	• Keeps team members aware of each other project activities and defends team members to get distracted due to excessive communication.	S8, S15	2
SP7	Light Documentation on shared electronic workspace.	• Decreases misunderstanding, improves coordination, transparency and collaboration between teams.	S4, S12, S19, S20, S24, S31	6
SP8	Photos of all team members were pasted at all locations.	• Provides feeling of teamness, increases familiarity and trust about remote members.	S19, S29	2
SP9	Clocks of all time-zones placed at each site.	• Supports coordination.	S22	1
SP10	Mirroring and balanced sites	• Reduce dependency, delay, and frustration at remote sites, empower decision making,	S3, S20, S29, S31	4

		communication, coordination, and trust.		
SP11	Encourage professional accreditation	<ul style="list-style-type: none"> <li>• Enhances trust and confidence in distributed team.</li> </ul>	S6	1
SP12	Previous working relations	<ul style="list-style-type: none"> <li>• Created a cohesive team culture.</li> </ul>	S4	1
SP13	Reward team performance	<ul style="list-style-type: none"> <li>• Promotes team cohesiveness</li> </ul>	S37	1
SP14	Employ people with technical and soft skills	<ul style="list-style-type: none"> <li>• Foster collaboration.</li> </ul>	S37	1
SP15	Cultural and language trainings	<ul style="list-style-type: none"> <li>• Reduces socio-cultural, linguistic distance and improves collaboration.</li> </ul>	S6, S29	2
SP16	Social annual get-together	<ul style="list-style-type: none"> <li>• Increases cohesiveness of team.</li> <li>• Make distributed team know each other.</li> </ul>	S16	1

## 5. Results of Industrial Survey

This section discusses result analysis of industrial survey. Section 5.1 discloses interesting facts related to contextual information of respondents. Section 5.2 discusses usage of agile practices by respondents. Benefits gained by agile practices, challenges faced in applying agile practices, and supporting practices employed are discussed in Section 5.3, 5.4, and 5.5, respectively.

### 5.1. Contextual Information of Respondents

Detailed contextual information such as, experience in software industry, experience in distributed agile development, designation, form of collaboration, total members in the project, number of locations, development process used, domain of project, onshore locations and offshore locations are used for the evaluation of industrial survey. Certain useful facts are observed through the survey.

The 83% of respondents have experience above 6 years, which shows that respondents have fine knowledge about software development. The 48% respondents were applied agile methods in distributed projects for more than 3 years. Thus, they would have good understanding of distributed agile practices. Respondents were performing different roles in companies. Therefore, responses exhibit diversity. The 63% of projects were outsourced as compared to 34% insourced projects in GSD. This indicates the popularity of offshore outsourcing over offshore insourcing. The 79% of projects have small to medium team size (up to 20) and 93% of projects have two or three distributed sites. The 76% of respondents were using scrum in distributed projects whereas 14% were using combination of XP and scrum methods in GSD. This indicates that scrum practices are well suited for GSD environment. Most recurrent onshore location is US and most popular offshore location is India.

### 5.2. Usage of Agile Practices

Usage frequency of agile practices in distributed projects is shown in Figure 1. It unveils that continuous integration, sprint, sprint planning, and daily Scrum are some of the popular practices. Coding standards, user story and backlogs are frequently or moderately used by most practitioners. Metaphor and planning game are agile practices that are not used by many of the respondents. Result shows the amalgamation of Scrum and XP in distributed agile projects and also illustrates the popularity of Scrum in software industry over other development methods.

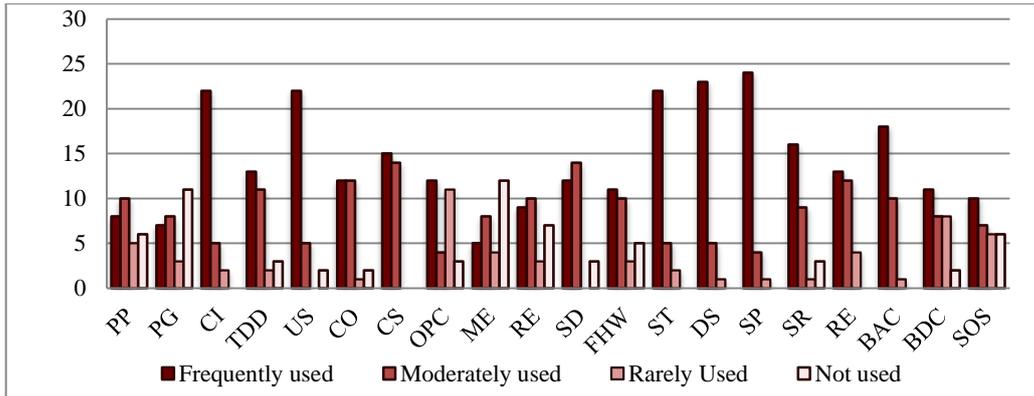


Figure 1. Usage of Agile Practices in GSD

### 5.3. Benefits of Applying Agile Practices

The survey result shown in Figure 2, strongly advocates that agile practices focus on solving issues, misunderstanding, and impediments timely, which is coded as B4. Most of the respondents also believe that agile practices enhance communication between sites, coded as B1, and B2. Respondents also realized that agile practices also improved visibility of project activities; they aid in evaluating and measuring the progress of the project, and improve shared understanding, coded as B7, B8, and B17, respectively. In all, it can be interpreted that agile practices help in alleviating most of the challenges faced in GSD. Most of the respondents disagree that agile practices increase motivation of team members, as shown by B18.

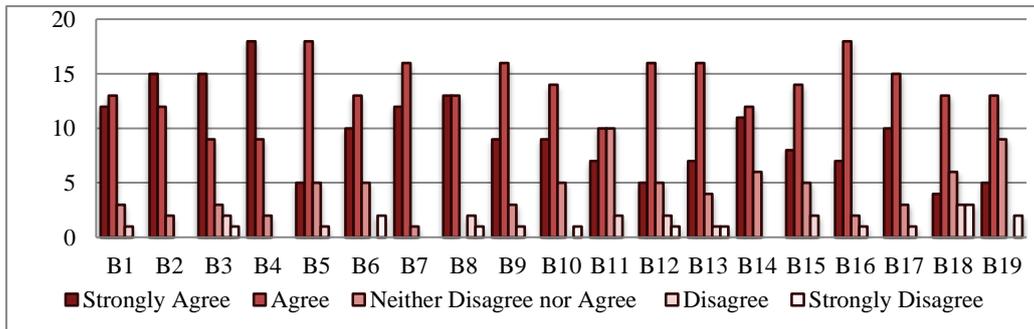
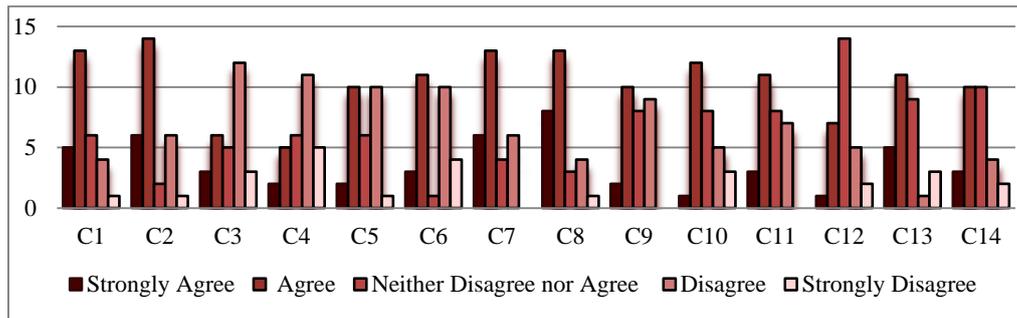


Figure 2. Benefits of Agile Practices in GSD

### 5.4. Challenges Faced in Applying Agile Practices in GSD

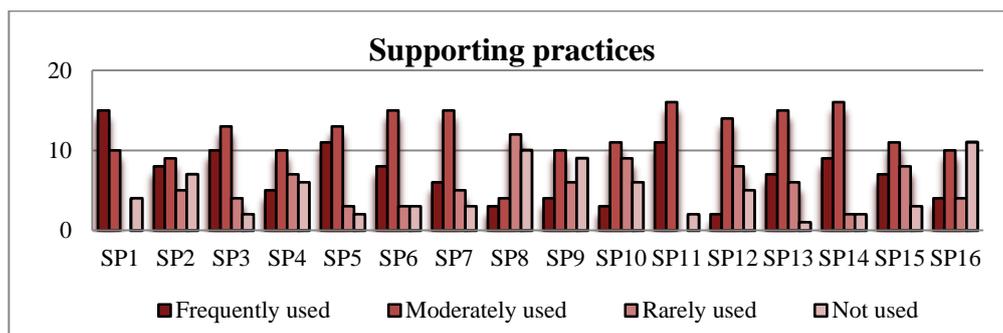
Challenges faced while applying agile practices identified from the survey are shown in Figure 3. Difficulty in arranging short as well as long meetings, coded as C2 and C8 are considered as major challenges by respondents. They also agree that linguistic difference restrict active participation of offshore members in meetings (C7). 72% respondents either have no perception or disagrees that it is difficult to manage tacit knowledge in GSD projects, coded as C12.



**Figure 3. Challenges of Agile Practices in GSD**

### 5.5. Supporting Practices

The usage frequency of supporting practices as derived from survey responses is shown in Figure 4. The practice of relocating an offshore person at onshore location, coded as SP1, is followed by most of the respondents. Multiple communication tools (SP5), professional accreditation of team members (SP11), and hiring people with good technical as well as soft skills (SP14) are the mostly used by the practitioners. The whole team at all locations (SP8) is rarely used or not used the practitioners. Other unused practices are social get-together annually (SP16), and keeping clocks of time zones of all project locations (SP9).



**Figure 4. Supporting Practices in GSD**

### 6. Supporting Tools

Both literature review and survey responses have indicated the use of multiple communication and collaboration tools in their projects. Teleconferencing, email, instant messaging, videoconferencing, wikis, Skype, and Lync are some of the popular communication and collaboration tools. While, tools such as webcam, chat room, VOIP, OCS, WebEx, Screen sharing, Shared whiteboard software, Microsoft Net meeting, Blue Jeans, SharePoint, and GoToMeeting are less frequently discussed in literature and survey. Both literature and survey reveal that Jira, Microsoft project plan (MPP), and TFS tools are often used for managing distributed agile projects whereas only few survey respondents have preferred BaseCamp, Stash, Mingle, and Youtrack tools.

Subversion, TFS, and Concurrent Version System are some of frequently used configuration management tools. Few survey respondents have supported the use of Git, Virtual Source Safe, Oracle tools, Jenkins, Cruise Control.Net, and Clear case for configuration management. Bug and issue management tools are least discussed in literature. Survey responses indicate that most practitioners use Jira for managing bugs and issues. Other tools used for bug management are Bugzilla, TFS, Sharepoint, Youtrack, BugTracker, and Quick Test Pro. Jira and TFS are frequently used tools for maintaining backlogs and burndown charts as indicated in literature as well as survey.

Tools such as VersionOne, Ms Excel, ScrumWorks, Scrumwise, MPP, GreenHopper, and Lotus Notes are also used for maintaining backlogs and burndown charts.

## 7. Discussion

Based on the results of literature review and industrial survey, we have derived certain inferences. Over the years, Scrum has become the most used agile method in GSD setting in contrast to distributed extreme programming. Daily Scrum and sprint are most popular DASD practices whereas, metaphor and planning game are least used practices in GSD.

The results of literature review and survey both strongly advocate that agile practices aid in solving issues, misunderstanding, and impediments timely. These practices increase shared understanding, help to expose issues at early stage of the project, improve visibility of project, and assist in monitoring the progress of project. Thus, it can be interpreted from this study that agile practices are capable of solving most of the issues of GSD.

We have found 14 challenges of applying agile practices in GSD. Several authors of literature consider slow, unreliable, and ineffective communication networks as an issue in distributed agile projects, whereas survey respondents do not consider it as a big challenge. Survey respondents realize that arranging distributed meetings are more challenging. Most of the respondents feel that bandwidth provided to them is sufficient to use tools whereas literature stated challenges related to inability to use several tools due to insufficient bandwidth. Results of literature review and industrial survey both indicate that practitioners are less worried about the challenges of applying agile practices in GSD and more enthusiastic towards the benefits of distributed agile practices. Thus, the practitioners are whole-heartedly prepared to adapt distributed agile practices.

It is observed from literature review and survey that supporting practices are used to supplement distributed agile practices. Supporting tools used in distributed agile projects are discussed infrequently in the literature, except tools used for communication and collaboration. It is observed that Jira and TFS are most extensively used tools.

## 8. Conclusion

Global software development has become a ubiquitous norm of software development whereas agile methods have turned into contemporary way of developing software. In order to increase the understanding of distributed agile practices, we have investigated agile practices along with supporting practices and tools, through literature review and industrial survey. The results of literature review and survey both show similarity, and indicate that most of the agile practices can be successfully applied in distributed projects, with slight modifications. Agile practices, supplemented with supporting practices and tools, can overcome challenges that GSD projects come across due to various distances. Benefits of agile practices outweigh the challenges of applying them. Through this study, software professionals can find the pros, cons, and modifications in various agile practices in distributed setting. This paper would aware software practitioners of the major challenges of DASD. Thus, this study provides researchers as well as practitioners the current state of affairs of agile in GSD. In future, we will work towards designing process model for GSD using adaptive agile practices.

## References

- [1] J. D Herbsleb and D. Moitra, "Global Software Development", IEEE Software, vol. 18, no. 2, (2001), pp. 16-20.
- [2] R. Sangwan, M. Bass, N. Mullick, D. J. Paulish and J. Kazmeier, Editors, "Global Software Development Handbook", Auerbach Series on Applied Soft. Engg., Auerbach, USA, (2006).
- [3] R. Jain and U. Suman, "A Systematic Literature Review on Global Software Development Life Cycle", ACM SIGSOFT SEN, vol. 40, no. 2, (2015), pp. 1-14.
- [4] I. Bose, "Lessons Learned from Distributed Agile Software Projects: A Case-based Analysis", Communications of the Association for Information Systems, vol. 23, no. 34, (2008), pp. 619-632.

- [5] A. Cockburn, Editor, "Agile Software Development: The Cooperative Game", Addison-Wesley, (2007).
- [6] S. Jalali and C. Wohlin, "Global software engineering and agile practices: a systematic review", *J. Software Maintenance and Evolution: Research and Practice*, vol. 24, no. 6, (2011), pp. 643-659.
- [7] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, "Agile software development methods: Review and analysis", VTT Technical report, (2002), Espoo, VTT Publications, Finland.
- [8] D. Cohen, M. Lindvall and P. Costa, "An Introduction to Agile Methods", *Adv. Comput.*, vol. 62, (2004), pp. 1-66.

### List of Studies included in Literature Review

- [S1] P. L. Bannerman, E. Hossain, and R. Jeffery, "Scrum practice mitigation of global software development coordination challenges: A distinctive advantage?", *Proceedings of the 45th Hawaii International Conference on System Sciences*, HI, USA (2012) January 4-7, pp. 5309-5318.
- [S2] H. Holmstrom, B. Fitzgerald, P. Agerfalk, and E. Conchuir, "Agile Practices Reduce Distance in Global Software Development", *Information Systems Management*, vol. 23, no. 3, (2006), pp. 7-18.
- [S3] M. Paasivaara, S. Durasiewicz, and C. Lassenius, "Using Scrum in Distributed Agile Development: A multiple Case Study", *Proceedings of the 4th ICGSE, Brazil*, (2009) August 23-26, pp. 195-204.
- [S4] B. Ramesh, L. Cao, K. Mohan and P. Xu, "Can Distributed Software Development be Agile?", *Comm. ACM*, vol. 49, no. 10, (2006), pp. 41-46.
- [S5] D. Batra, "Modified Agile Practices for Outsourced Software Projects", *Comm. ACM*, vol. 52, no. 9, (2009), pp. 143-148.
- [S6] F. Dumitriu, D. Oprea and G. Mesnita, "Issues and strategy for agile global software development adoption", *Proceeding of the 3rd World Multiconference on Applied Economics, Business and Development*, Iasi, Romania, (2011) July 1-3, pp. 37-42.
- [S7] E. Hossain, P.L. Bannerman and R. Jeffery, "Scrum Practices in Global Software Development: A Research Framework", *Proceedings of the International Conference on Product Focused Development and Process Improvement*, Springer-Verlag, LNCS, vol. 6759, (2011), pp. 88-102.
- [S8] M. Vax and S. Michaud, "Distributed Agile: Growing a practice together", *Proceedings of Agile '08*, Toronto, Ontario Canada, (2008) August 4-8, pp. 310-314.
- [S9] M. Paasivaara and C. Lassenius, "Using Scrum Practices in GSD Projects", Edited D. Šmite, N. B. Moe, P. J. Ågerfalk, first, Springer, ed., (2010), pp. 259-278.
- [S10] E. Hossain, M. Babar, and H. young Paik, "Using Scrum in Global Software Development: a Systematic Literature Review", *Proceedings of the 4th ICGSE, Ireland*, (2009), pp. 175-184.
- [S11] J. M. Bass, "Influences on agile practice tailoring in enterprise software development", *Proceedings of AGILE India, Bengaluru, India*, (2012) February 17-19, pp. 1-9.
- [S12] S.V. Shrivastava, and H. Date, "Distributed Agile Software Development: A Review", *Journal of Computer Science and Engineering*, vol. 1, no. 1, (2010), pp. 10-17.
- [S13] L. Layman, L. Williams, D. Damian, and H. Bures, "Essential Communication Practices for Extreme Programming in a Global Software Development Team", *Information Software and Technology*, vol. 48, no. 9, (2006), pp. 781-794.
- [S14] J. V. Hillegersberg, G. Ligtenberg, and M.N. Aydin, "Getting Agile Methods to Work for Cordys Global Software Product Development", *LNBIP*, vol. 91, no. 1, (2011), pp. 133-152.
- [S15] M. Kircher, P. Jain, A. Corsaro, and D. Levine, "Distributed extreme programming", *Proceedings of the International Conference on eXtreme Programming and Flexible Processes in Software Engineering*, Sardinia, Italy, (2001) May 20-23, pp. 147-154.
- [S16] M. Paasivaara, S. Durasiewicz, and C. Lassenius, "Distributed agile development: using scrum in a large project, in: *Global Software Engineering*", *Proceedings of 6th ICGSE, Helsinki, Finland*, (2011) August 15-18, pp. 87-95.
- [S17] J. Sutherland, A. Viktorov, J. Blount and N. Puntikov, "Distributed Scrum: Agile Project Management with Outsourced Development Teams", *Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, Waikoloa, Big Island, HI, USA, (2007) January 3-6, pp. 274a-274a.
- [S18] G. Papadopouloa, "Moving from traditional to agile software development methodologies also on large, distributed projects", *Procedia-Social and Behavioral Sciences*, vol. 175, (2015), pp. 455-463.
- [S19] K. Sureshchandra and J. Shrinivasavadhani, "Adopting agile in distributed development", *Proceedings of the 3rd ICGSE, Bangalore, India*, (2008) August 17-20, pp. 217-221.
- [S20] K. Braithwaite and T. Joyce, "Xp Expanded: Distributed Extreme Programming", *Proceeding of the 6th International Conference Extreme Programming and Agile Processes in Software Engineering*, LNCS, vol. 3556, (2005) June 18-23, pp. 180-188.
- [S21] B. Sengupta, V. Sinha, and S. Chandra, "Test-Driven Global Software Development", *Proceedings of the International Workshop on Global Software Development*, Scotland, (2004) May 24, pp. 39-41.
- [S22] S. Berezuk, "Back to Basics: The Role of Agile Principles in Success with an Distributed Scrum Team", *Proceeding of the AGILE '07*, Washington, DC USA, (2007) August 13-17, pp. 382-388.
- [S23] L. Pries-Heje and J. Pries-Heje, "Why Scrum Works: A Case Study from an Agile Distributed Project in Denmark and India," *Proceedings of the AGILE*, Utah, (2011) August 8-11, pp. 20-28.

- [S24] M. Cristal, D. Wildt, and R. Prikladnicki, "Usage of Scrum Practices within a Global Company", Proceedings of 3rd ICGSE, Bangalore, India, (2008) August 17-20, pp. 222-226.
- [S25] P. J. Ågerfalk, B. Fitzgerald, "Flexible and Distributed Software Processes: Old Petunias in New Bowls?", Comm. ACM, vol. 49, no. 10, (2006), pp. 26-34.
- [S26] M. Hallikainen, "Experiences on Agile seating, facilities and solutions Multisite environment", Proceedings of the 6th ICGSE, Helsinki, Finland, (2011) August 15-18, pp. 119-123.
- [S27] Y. Xiaohu, X. Bin, H. Zhijun, and S. Maddineni, "Extreme Programming in Global Software Development", Proceedings of the Canadian Conference on Electrical and Computer Engineering, Ontario, Canada, vol. 4, (2004) May 2-5, pp. 1845-1848.
- [S28] E. Hossain, P. L. Bannerman, and R. Jeffery, "Towards an Understanding of Tailoring Scrum in Global Software Development: A Multi-case Study", Proceedings of the ICSSP'11, Honolulu, Hawaii, USA, (2011) May 21-22, pp. 110-119.
- [S29] S. Dorairaj and J. Noble, "Agile Software Development with Distributed Teams: Agility, Distribution and Trust", Proceedings of the AGILE' 13, Tennessee, USA (2013) August 5-9, pp. 1-10.
- [S30] R.K. Gupta and P. Manikreddy, "Challenges in Adapting Scrum in Legacy Global Configurator Project", Proceedings of the 10th ICGSE, Ciudad Real, Spain, (2015) July 13-16, pp. 46-50.
- [S31] B. Hogan, "Lessons learned from an extremely distributed project", Proceedings of the AGILE' 06, Minneapolis, Minnesota USA, (2006) July 23-28, pp. 321-326.
- [S32] C. Young and H. Terashima, "How Did We Adapt Agile Processes to Our Distributed Development?", Proceedings of AGILE '08, Toronto, Ontario, Canada (2008) August 4-8, pp. 304-309.
- [S33] R. Sriram and S.K. Mathew, "Global software development using agile methodologies: A review of literature", International Conference on Management of Innovation and Technology, Bali, Indonesia (2012) June 11-13, pp. 389-393.
- [S34] M. Paasivaara, and C. Lassenius, "Could Global Software Development Benefit from Agile Methods?", Proceedings of the 1st ICGSE, Florianópolis, Brazil, (2006) October 16-19, pp. 109-113.
- [S35] E. Hossain, M. Babar, H. young Paik, and J. Verner, "Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: a Conceptual Framework", Proceedings of the Asia-Pacific Software Engineering Conference, Penang, Malaysia, (2009) December 1-3, pp. 457-464.
- [S36] G.K. Ghosh, "Challenges in Distributed Scrum", Proceedings of the 7th ICGSE, Porto Alegre, Brazil, (2012) August 27-30, pp. 200-200.
- [S37] S. Dorairaj, J. Noble and G. Allan, "Agile Software Development with Distributed Teams: Senior Management Support", Proceedings of the 8th ICGSE, Italia, (2013) August 26-29, pp. 197-205.
- [S38] R. Phalnikar, V.S. Deshpande, and S.D. Joshi, "Applying Agile Principles for Distributed Software Development", Proceedings of the International Conference on Advanced Computer Control, Singapore, (2009) January 22-24, pp. 535-539.
- [S39] K. Dullemond, B. van Gameren, and R. van Solingen, "How Technological Support Can Enable Advantages of Agile Software Development in a GSE setting", Proceedings of the 4th ICGSE, Limerick, Ireland, (2009) July 13-16, pp. 143-152.
- [S40] T. Schümmer and J. Schümmer, "Support for distributed teams in eXtreme programming", Proceedings of the Extreme programming examined, AddisonWesley, (2001), pp. 355-377.

## Authors



**Ritu Jain** is a PhD research scholar at School of Computer Science & IT, Devi Ahilya University, Indore, MP, India. She has done her Master of Computer Application in 2005. She has been into academics since last 10 years. She has qualified UGC-NET (Computer Applications) in June 2012. Her research interests include Global Software Development and Agile Methods.



**Ugrasen Suman** is currently working as a professor at School of Computer Science & Information Technology, Devi Ahilya University, Indore, MP, India. He has received his PhD degree in Computer Science from Devi Ahilya University Indore, India. His areas of research are Software Engineering, Information System, Software Reuse, Software Maintenance, & Reengineering, Agile Methods, Software Architectures, and Service-Oriented Computing. He has published more than 75 research papers in National & International Journals/ Conferences. He has authored a book,

Software Engineering: Concepts & Practices (Cengage Learning, 2013).