A Case Study of Coordination in Distributed Agile Software Development

Steinar Hole[1] and Nils Brede Moe[2]

[1] NTNU Department of Computer and Information Science
NO-7491 Trondheim, Norway
steinaho@stud.ntnu.no
[2] SINTEF Information and Communication Technology
NO-7465 Trondheim, Norway
Nils.B.Moe@sintef.no

Abstract. Global Software Development (GSD) has gained significant popularity as an emerging paradigm. Companies also show interest in applying agile approaches in distributed development to combine the advantages of both approaches. However, in their most radical forms, agile and GSD can be placed in each end of a plan-based/agile spectrum because of how work is coordinated. We describe how three GSD projects applying agile methods coordinate their work. We found that trust is needed to reduce the need of standardization and direct supervision when coordinating work in a GSD project, and that electronic chatting supports mutual adjustment. Further, co-location and modularization mitigates communication problems, enables agility in at least part of a GSD project, and renders the implementation of Scrum of Scrums possible.

Keywords: Agile development, Scrum, case study, coordinating work, mutual adjustment, direct supervision, standardization, global software development.

1 Introduction

Many organizations turn toward global software development (GSD) in their quest for cheap, higher-quality software with a short development cycle. GSD is becoming the norm by promising potential advantages like global resources, attractive cost structures, round-the-clock development and closeness to local markets [1].

To unleash the potential, methods and tools for distributed software development are designed to enable dispersed team members to share programming tasks and development practices [2]. Methods and tools are needed to mitigate GSD problems related to coordination, communication, control [3], and increased complexity [4].

Recently, there has been a growing interest in applying agile approaches in GSD to solve some of the coordination and communication challenges [3]. Several reports on the successful implementation of agile values and principles in different GSD projects conclude that there are significant differences between the fundamental principles of agile and distributed approaches, while there is a growing interest in assessing the viability of using agile practices for distributed teams [5-7].

R.V. O'Connor et al. (Eds.): EuroSPI 2008, CCIS 16, pp. 189–200, 2008.

Agile development approaches and GSD approaches differ significantly in their key tenets, e.g. regarding coordination mechanisms [6]. Traditional GSD focuses on command-and-control, formal communication, and is usually implemented using a mechanistic (bureaucratic with high formalization) organizational structure. Agile development focuses on leadership-and-collaboration, informal communication and the desire for an organic organizational form [8]. Therefore, applying agile principles to GSD marks an intersection of two seemingly incompatible approaches.

Ramesh et al. [6] demonstrate how the balancing between agile and distributed approaches can help when introducing agility in GSD. They suggest that project leaders and champions should participate in coordinating the activities of the local and remote teams to help achieve project goals. Motivated by the work of Ramesh et al. [6], we investigate how work is coordinated when introducing agile methods in a GSD environment. Our research question is:

"How are tasks coordinated in GSD teams applying agile methods?"

The remainder of the paper is organized as follows. Section 2 describes GSD and agile development, and the challenges associated with merging these two approaches. Section 3 describes our research method. In Section 4, we present results from a multiple case study on agile methods and practices applied to three GSD projects. Findings are discussed in Section 5. Section 6 concludes and suggests future research.

2 Background

In this section we present background information on agile development and GSD. We use literature to describe challenges with coordination in an agile GSD context.

2.1 Agile Methods and Scrum

Agile software development comprises a number of practices and methods [9-11]. Among the most known and adopted agile methods are Extreme Programming (XP) [12] and Scrum [13]. XP focuses primarily on the implementation of software, while Scrum focuses on agile project management [14]. In this study the focus is on Scrum since Scrum is an agile approach to the management of software development projects [9-11], and thus focuses on the coordination of work.

Scrum and agile development favor a leadership-and-collaboration style of management where the traditional project manager's role is replaced with the Scrum master's role of a facilitator or coordinator [9-11]. The Scrum master is in charge of solving problems that prevents the Scrum team (5-9 people) from working effectively. He or she is often described as a coach or facilitator and does not organize the team (designers and developers); the team organizes itself and makes decisions concerning what to do. The Scrum master works to remove the impediments of the process, makes decisions in the daily meetings and validates them with management [13].

Software is developed by the self-organizing team in increments called "sprints", starting with planning and ending with a review. The team coordinates on a daily basis. Features to be implemented are registered in a backlog, and a product owner decides which backlog items should be developed in the following sprint. These items are specified in a sprint backlog.

The product backlog comprises a prioritized and constantly updated list of business and technical requirements for the system being built or enhanced. Backlog items can include features, functions, bug fixes, requested enhancements and technology updates. Multiple stakeholders can participate in generating product backlog items, such as customer, project team, marketing and sales, management and support [11].

2.2 Coordinating Mechanisms in Agile Development and GSD

The issue of agile approaches in distributed development has caught the attention of several researchers. There have been many studies reporting on the successful implementation of agile practices in GSD [5-7, 15-17], but a number of implementation barriers are also mentioned by these authors. The combination of both agile and GSD is poorly understood although it is expected to be beneficial [3]. Exploring theories on coordination of work is one way of understanding this combination.

Coordination of work is an important aspect of teamwork and team leadership [18]. Coordination together with communication and collaboration are recognized as the key enablers of software development processes [19]. There are three basic coordinating mechanisms that seem to describe the fundamental ways in which organizations can coordinate their work [20]:

1. Mutual adjustment - based on the simple process of informal communication
2. Direct supervision - one person takes responsibility for the work of others by issuing instructions and monitoring their actions
3. Standardization - of which there are four types: work processes, output, skills (as well as knowledge) and norms

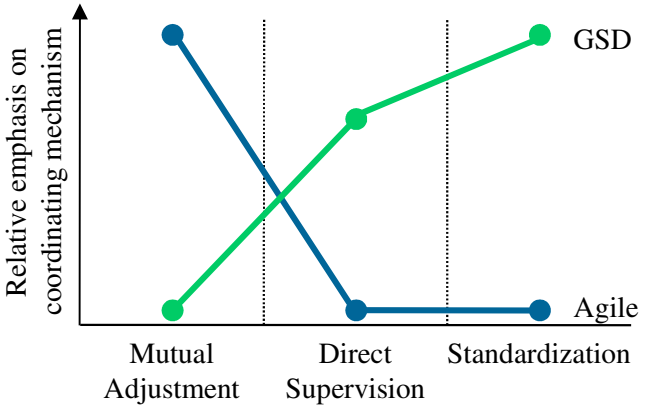

Fig. 1. Relative emphasis on coordinating mechanisms: Agile development relies purely on mutual adjustment, while GSD emphasizes standardization and some direct supervision

GSD usually relies mainly on formal mechanisms (coordination by standardization), which exploit detailed architectural design and plans to address impediments to team communication induced by geographical separation [3, 6]. Agile development relies on people and their creativity rather than on processes [21], and emphasizes informal communication (mutual adjustment) as the primary coordinating mechanism [8].

The major challenge of applying agile methods or practices in a GSD context is to balance the coordinating mechanisms (Fig. 1). However there are obvious conflicts when trying to balance mutual adjustment, direct supervision and standardization.

3 Research Design and Method

The goal of this research is to understand how the introduction of agility affects coordination of tasks in global software development teams. It is therefore important to study software development teams in practice. We have collected data from three teams using Scrum and participating in globally distributed software projects.

We report on a multiple case holistic study [22], in which we studied one phenomenon in several projects in one company. In a multiple case study, each case must be selected carefully so that it either a) predicts similar results or b) predicts contrasting results but for predictable reasons [22]. We chose option a).

3.1 Study Context

This study was done in the context of a larger action research program, where several companies have introduced elements from agile development in response to identified problems. The software company is medium-sized with approximately 150 employees in four major departments. The second author of this paper participated in the introduction and training of Scrum, and observed the company while using Scrum. The first author conducted the interviews, which we use as the primary source of data for this study. The projects participating in the study were all using Scrum for the first time; however this company was experienced with using GSD.

3.2 Data Sources and Analysis

To address the research questions, we conducted semi-structured interviews with the persons most responsible for coordination of work in the three projects, i.e. a Scrum master, a project manager and a product owner. One person was selected from each project. The interviews lasted from 30 to 40 minutes, and aimed at understanding how Scrum was applied in a GSD context. The interview guide was based on the three coordinating mechanism as proposed by Mintzberg [20] in addition to questions related to Scrum. We focused on understanding coordination of work, communication within and between the teams, feedback-sessions, planning and estimation, use of documentation, roles and specializations, and how decisions were made. All the interviews were transcribed.

4 Agility in GSD Projects

We now present the three GSD projects under study, how Scrum was implemented in these projects, and how work was coordinated in the projects.

4.1 Project India I

The goal of the project is to develop a system for integrity management of pipelines both offshore and onshore. Today several customers are interested in buying the

product, and so far three contracts have been signed. One of the biggest challenges in this project is to align requirements from potential customers from all over the world. Scrum was introduced one year after the project had started.

The project consists of six developers working full time (one is a Scrum master), two GUI designers, one product owner, and one project manager working 50% on this project. Four of the developers are situated in India together with one tester. To improve communication one of them was in periods moved to Norway.

The sprints usually lasted three weeks, ending on a Friday with a retrospective- and review-meting. The next sprint was planned the following Monday. The team organized a 15 minutes stand-up every morning discussing project related issues. The product owner usually joined all the different Scrum meetings.

Coordinating GSD Work in the India I Project. Before using Scrum the team relied on standardization and direct supervision when coordinating work with their Indian team. In the beginning, the remote team was given some easy tasks specified by the Norwegian team. The Scrum master said: "In the beginning the quality was varying, and then we thought they should only concentrate on the testing. Then they said 'No, this is not fun, please give us something more exiting to work on', and then they got different tasks, and this worked pretty well."

After using Scrum for 6 months the project had implemented all the Scrum practices, and felt they were succeeding with continuously improving their Scrum process. The team tried to work as if they were all collocated, ignoring the geographical and time differences. The Scrum master said: *"It is a big barrier being distributed. We used a lot of time on discussions between people in the two sub-teams. It did not work. The solution was to appoint one of the remote developers the role of a local Scrum master. And then we mostly communicated with her."*

To improve the communication it was decided to let the Indian Scrum master stay in Norway for a period. The Scrum master said: *"This improved the situation a lot. The productivity increased while she was here. The important issue is to communicate with only one person."* She was participating in all the Scrum meetings while situated in Norway. At the same time it was also decided to let the remote team work on its own module.

Even though they started applying Scrum, and assigning a member of the remote team as a local Scrum master, the coordination between the two teams was still described as a traditional way of developing software. During the planning meetings in Norway, the local team would plan and suggest initial estimates for all the tasks in the project, and then assign tasks to their remote partner. Later the remote team would turn these tasks into sub-tasks, and provide new estimates. In the end, the Norwegian team would check the results.

The Norwegian Scrum master, the Scrum master from India and one of the Norwegian developers had frequent meetings (2-3 times a week) with the remote team. This was a kind of distributed stand-up. In the meetings between the two sub-teams they relied on chat and e-mail. The Scrum master said: *"We tried to use telephone-conferences, but it did not work well, because of language problems. It is also easier to understand each other when relying on written communication. Also extensive use of chatting makes it possible to ask a question right away. It takes time to organize a telephone-conference."* He continued: *"It was also difficult to only use 15 minutes on the telephone. Often we used an hour. Chat is better."*

Coordination of work with the remote team was mostly based on direct-supervision. The Scrum master from India was involved in the meetings but she then decided who should do what.

4.2 Project India II

The goal of the project was to develop a system for quality audits in organizations. This project represents the second release of the system and will provide multi user support. Two departments of the studied company are involved, each acting as an internal customer responsible for contracts with their own international customer.

The project consists of a product owner, who is also a project manager, and an architect from Norway, while development is outsourced to India. Four remote developers are working 100% on the project, one of them as a team leader. In addition a few remote developers contribute part time on the project. The Indian team members are given specialized responsibilities, like GUI.

Scrum was applied from the inception of this project because, according to the product owner, "*our customer didn't understand the creation of an old-fashioned functional specification, so we thought: Okay, let's try an agile approach*." They agreed on a contract that allowed the use of a backlog with a constantly updated list of business and technical requirements, and continuous deployment of short deliveries. The backlog was maintained by the product owner. In addition to the described Scrum practices, they used continuous integration and semi-automatic deployment, and code reviews.

Coordinating GSD Work in the India II Project. The project started after the first initial backlog was created by the product owner. After the initial design was created, the work was then planned and divided into sprints in cooperation with the Indian team. This failed. The product owner said: *"I quickly gave up these sprints, that is, to define them together with the remote team."* She continued: *"It was very difficult because of problems with the communication. […] We didn't understand each other, and then there were cultural differences, too."*

The product owner explained how they changed their way of coordinating work, after finding it too time consuming to do the sprint planning in cooperation with the remote team: *"We then started sending them work-packages specified in detail, but we realized it would be a too big job to do this for each work package."* The solution was then to create a principal work plan and then further specify and document backlog items with use-cases described in documents.

The product owner and the remote team leader communicate daily, often several times a day. She said: *"There has been a team leader down there who assigned the tasks to the team, so I've only been dealing with him."*

The assignment of tasks to the Indian team became less detail oriented and instead there was an increased focus on continuous communication. It seems like the product owner tried to act more as described in the Scrum literature. She was maintaining the backlog and specifications, while letting the Indian team work out the specifications: *"I do not know everything, therefore I try to communicate: 'This is the use case, you need to solve this. Work it out.' And it works, and then they ask: 'Can we discuss', and of course, we do."*

Coordination of work with the remote team was mainly based on direct supervision and standardization in the form of written specifications and reporting of status, but the team was also relying on frequent informal communication. However, the biggest challenge was getting feedback from the remote team. The product owner said: *"What I miss, though, is that they should detect problems and show initiative."*

4.3 Project Eastern Europe

The goal of the project is to develop a system for collection and visualization of data from ship-inspections. When ships are inspected, the results are stored in the system, and the collected data are visualized through 3D models. The 3D engine was first developed as a prototype 5 years ago, before it was integrated into the core system and then released. Each time the product is sold to a new customer it requires adaptation and modification of the system. Several contracts with different customers from all over the world have been signed.

Four to five developers are situated in the remote team in an East European country, while two developers are situated in Norway, together with two persons from the support department, one from sales and a project manager acting as a product owner. The Norwegian team implements the daily Scrum. These meetings are also used for discussion of future solutions. They tried to implement sprints for the whole project, but failed. Tasks are mostly assigned to the Norwegian team's members by the project manager, who said, while pointing at the backlog: *"There, I've been putting some signatures on who is going to do what."*

Coordinating GSD Work in the Eastern Europe Project. The project was originnally applying a traditional, waterfall inspired model. This changed a year ago when a new project manager was assigned. The two distributed teams tried to use a common Scrum process. They were conducting several joint stand-ups each week, and implemented shared responsibilities. Originally, the remote team was only responsible for the creation of 3D models, but when it was decided to integrate them in the total development process, they faced new challenges. The project manager said: *"We thought that we should try Scrum, but because we wanted the remote team to take part in development and bug fixing, stand-up became a challenge. […] We didn't manage to interact and cooperate, it became too time consuming."*

According to the project manager, the remote team was unfamiliar with the system. This unfamiliarity made communication time consuming. The project manager said: *"We felt that the Norwegian team members used too much time communicating with the remote team."* The project manager also felt that the remote team did not deliver as expected. She said: *"Often, the software seemed inadequately tested."* This dissatisfaction was communicated to the remote team.

The project manager considered the problem to be difficulties gaining a thorough understanding of the complex source code, and commented on how tasks were divided: *"If we had managed to identify bigger chunks of new functionality to be developed by the remote team, it might have been easier for them."* To improve the situation it was decided to divide responsibility between the teams and to give the remote team tasks that required less cross-site coordination. The Norwegian team is now responsible for the core system, bug fixing, new functionality and customer relations, while the remote team is mainly responsible for system configuration and the

creation of 3D models for each customer. The project manager said: *"Because of their 3D competency, it works, because then they don't have to communicate with us all the time. […] It's only if they lack a specification or domain knowledge, for instance when they miss an overview of what to put on the ship, then they come back and ask."*

Coordination of work between the teams was mainly based on standardization, and to some degree direct supervision. The level of mutual adjustment was low.

5 Discussion

In this section we present our key observations in light of our research question: *How are tasks coordinated in GSD teams applying agile methods?* To answer this question we need to evaluate the degree to which the projects conformed to the generally accepted elements of the Scrum methodology. None of the projects succeeded in implementing a shared Scrum process for both the local and remote team, and only the local team in the India I project was using Scrum as intended. One reason for the reported problems was the failed attempt to implement mutual adjustment in the distributed process. Agile development relies on mutual adjustment. All the projects ended up using the traditional approach relying on direct supervision and standardization when coordinating remote work. Figure 2 summarizes the coordinating mechanisms used between the teams. The graphs are drawn from the bases of the interview data and are also discussed with the interviewees.

5.1 Challenges Implementing Mutual Adjustment in GSD

All three projects tried to implement daily stand-ups as they are the most important instrument for mutual adjustment. However, they all experienced these meetings as time consuming, because of the flow of questions from the remote site. Language and cultural differences were also a reason for the problems with these meetings. Communication problems, often reported in GSD projects [6, 23], led to the replacement of daily meetings with direct supervision and detailed specifications. This probably made it difficult to solve the communication problems [24], discuss the backlog, and to self-organize; one of the key tenets of agile development [25].

Ramesh et al. [6] suggest four practices to improve communication; synchronize work hours, provide for informal communication through formal channels, balanced coordination and constant communication. India II was only partly synchronized, but managed to communicate frequently and relied on formal channels, i.e. communication through people with dedicated roles. India I reduced the need for synchronization and coordination through modularization and communicated frequently with the remote Scrum master. The team from Eastern Europe used synchronized work hours, enabling constant communication, but the amount of communication and the lack of formalized channels negated the positive effect.

All three projects were using Scrum for the first time, and it is possible that more mature Scrum teams would communicate more efficiently because they may be more knowledgeable about and have a better understanding of issues related to applying an agile approach in a GSD project. Furthermore, none of the remote teams were trained in Scrum and this probably resulted in a lack of process understanding.

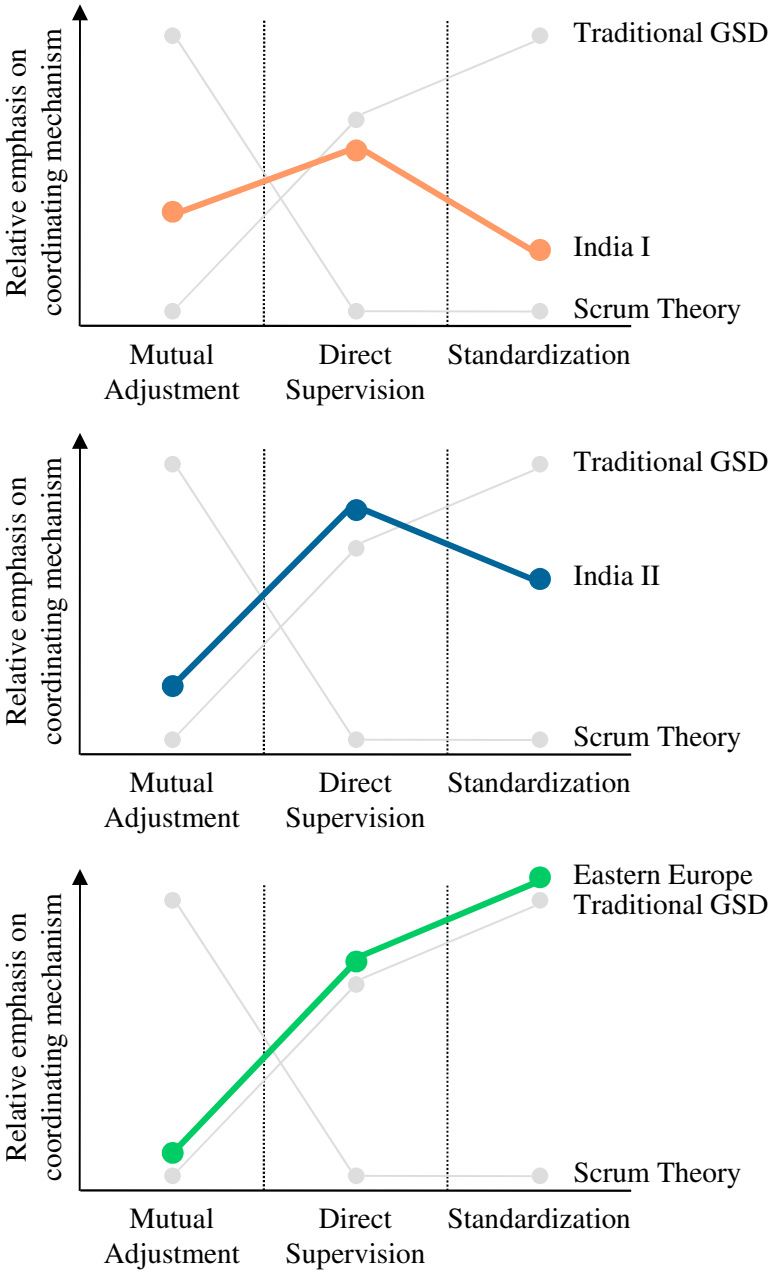

Fig. 2. Relative emphasis on coordinating mechanisms between the onshore and offshore teams: More emphasis is placed on direct supervision and standardization than on mutual adjustment

5.2 Implementing Scrum Practices

There was no joint Scrum process between the teams; however India I succeeded in implementing Scrum in Norway by dividing the project into modules, appointing a remote Scrum master, and by moving her to Norway for periods. The other projects used a similar approach, making the remote team responsible for specific modules. This reduced the need for everyone to communicate with everyone, and made communication less critical. The Eastern Europe project chose to assign standardized tasks to the remote team, as less complex tasks reduce the need for mutual adjustment [20]. Fowler [26] argues that this kind of modularization is important to succeed with distributed Scrum, because a remote team that is responsible for an entire module from planning to testing gets a deeper understanding of the tasks it is working on. He also suggests continuous integration to avoid surprises when integrating the modules.

Modularization also makes it possible to implement a Scrum of Scrums approach [27], where several teams follow their own Scrum process. The total process will then be coordinated through meetings between the Scrum masters. India I was in an early phase of implementing Scrum of Scrums.

Two of the projects improved their level of mutual adjustment after first substituting this coordinating mechanism with standardization and direct supervision. Electronic chatting was the best remedy to support mutual adjustment, since it is instant, written text is less hampered by noise than speech, and it was perceived as timesaving compared to using a telephone conference.

All projects focused on direct supervision after failing to use Scrum, but after some months, they all felt they could reduce their level of direct supervision because of an increased level of trust. Among the reasons for increased trust are frequent and reliable communication [24] and frequent visits by distributed partners [6]. Trust is a prerequisite for effective mutual adjustment [24].

6 Conclusion and Future Work

This paper presented data from a multiple case study. None of the projects succeeded in implementing mutual adjustment, and Scrum was only implemented in one local team. In the end the projects applied a subset of Scrum practices. We found that:

- A high level of trust is important for reducing direct supervision and standardization which is important to enable mutual adjustment.
- Co-locating the remote Scrum master with the local team and making the remote team responsible for dedicated modules, makes it possible to implement Scrum in part of a GSD project, and to implement Scrum of Scrums. This also reduces the need for everyone to communicate with everyone in the GSD project.
- The communication problems caused by distribution are a threat to mutual adjustment, however electronic chatting enables mutual adjustment.
- In addition, there is a need for more research utilizing formal analytical methods on how work is coordinated in mature agile GSD teams, e.g. teams using Scrum of Scrums, and when there is a common Scrum process.

Acknowledgement

We appreciate the input received from the project participants of the investigated company and from the review by Hamish Barney and Odd Nordland. This research is supported by the Research Council of Norway under Grant 174390/I40.

References

1. Damian, D., Moitra, D.: Global software development: How far have we come? IEEE Software 23, 17–19 (2006)
2. Canfora, G., Cimitile, A., Di Lucca, G.A., Visaggio, C.A.: How distribution affects the success of pair programming. International Journal of Software Engineering and Knowledge Engineering 16, 293–313 (2006)
3. Agerfalk, P.J., Fitzgerald, B.: Flexible and distributed software processes: Old petunias in new bowls? Communications of the ACM 49, 26–34 (2006)
4. Carmel, E., Agarwal, R.: Tactical approaches for alleviating distance in global software development. IEEE Software 18, 22–29 (2001)
5. Holmstrom, H., Fitzgerald, B., Agerfalk, P.J., Conchuir, E.O.: Agile practices reduce distance in global software development. Information Systems Management 23, 7–18 (2006)
6. Ramesh, B., Cao, L., Mohan, K., Xu, P.: Can distributed software development be agile? Communications of the ACM 49, 41–46 (2006)
7. Paasivaara, M., Lassenius, C.: Could Global Software Development Benefit from Agile Methods? In: Casper, L. (ed.) ICGSE, International Conference on Global Software Engineering, pp. 109–113 (2006)
8. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to agile methodologies. Communications of the ACM 48, 72–78 (2005)
9. Erickson, J., Lyytinen, K., Siau, K.: Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research. Journal of Database Management 16, 88–100 (2005)
10. Cohen, D., Lindvall, M., Costa, P.: An Introduction to Agile Methods. In: Zelkowitz, M.V. (ed.) Advances in Computers, Advances in Software Engineering, vol. 62. Elsevier, Amsterdam (2004)
11. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile software development methods - Review and analysis, vol. 478. VTT Publications (2002)
12. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change. Addison-Wesley, Reading (2004)
13. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall PTR, Upper Saddle River (2001)
14. Abrahamsson, P., Warsta, J., Siponen, M.T., Ronkainen, J.: New directions on agile methods A comparative analysis, pp. 244–254 (2003)
15. Farmer, M.: DecisionSpace infrastructure: agile development in a large, distributed team. Agile Development Conference, pp. 95–99 (2004)
16. Nisar, M.F., Hameed, T.: Agile methods handling offshore software development issues. In: Hameed, T. (ed.) International Multitopic Conference 2004, pp. 417–422 (2004)
17. Sulfaro, M.: Agile Practices in a Large Organization: The Experience of Poste Italiane. In: Concas, G., Damiani, E., Scotto, M., Succi, G. (eds.) XP 2007. LNCS, vol. 4536. Springer, Heidelberg (2007)

18. Salas, E., Sims, D.E., Burke, C.S.: Is there a "big five" in teamwork? Small Group Research 36, 555–599 (2005)
19. Layman, L., Williams, L., Damian, D., Bures, H.: Essential communication practices for Extreme Programming in a global software development team. Information and Software Technology 48, 781–794 (2006)
20. Mintzberg, H.: Mintzberg on Management: Inside Our Strange World of Organizations (1989)
21. Cockburn, A., Highsmith, J.: Agile software development: The people factor. Computer 34, 131–133 (2001)
22. Yin, R.K.: Case Study Research: Design and Methods. Sage Publications Inc., Thousand Oaks (2003)
23. Herbsleb, J.D., Mockus, A.: An empirical study of speed and communication in globally distributed software development. IEEE Transactions on Software Engineering 29, 481–494 (2003)
24. Moe, N.B., Smite, D.: Understanding Lacking Trust in Global Software Teams: A Multi-Case Study. In: Münch, J., Abrahamsson, P. (eds.) PROFES 2007. LNCS, vol. 4589, pp. 20–32. Springer, Heidelberg (2007)
25. Dyba, T., Dingsoyr, T.: Empirical Studies of Agile Software Development: A Systematic Review. Information and Software Technology (2008)
26. Fowler, M.: Using an Agile Software Process with Offshore Development (2003), http://www.martinfowler.com
27. Sutherland, J., Viktorov, A., Blount, J., Puntikov, N.: Distributed Scrum: Agile Project Management with Outsourced Development Teams. In: HICSS, p. 274 (2007)