

Preface

Despite the progress in the field of software engineering, software projects are still being late, are over budget, and do not deliver the expected quality. Two major trends have emerged in response to these: global sourcing and the application of agile methods. The new paradigms soon became anecdotally popular for their benefits of cheaper and faster development of high quality software. Many companies recently started to look into merging these two promising approaches into one strategy.

Globally Distributed Development

Global sourcing promises organizations the benefits of reaching mobility in resources, obtaining extra knowledge through deploying the most talented people around the world, accelerating time-to-market, increasing operational efficiency, improving quality, expanding through acquisitions, reaching proximity to market and many more. However, these benefits are neither clear-cut nor can their realization be taken for granted, as the literature may lead one to believe [1]. In fact, there are many challenges related to communication, coordination and control when developing software with global software teams [2].

Agile Development

Agile development has recently attracted huge interest from software industry [3]. It is being recognized for its potential to improve communication and, as a result, reduce coordination and control overhead in software projects. Methods for agile software development constitute a set of practices for software development that have been created by experienced practitioners [4]. The “agile manifesto” was published in 2001 by the key people behind the early agile development methods. The manifesto states that agile development should focus on four core values [5]:

- Individuals and interactions over processes and tools,

- Working software over comprehensive documentation,
- Customer collaboration over contract negotiation,
- Responding to change over following a plan.

Agile methods can be seen as a reaction to plan-based or traditional methods, which emphasize “a rationalized, engineering-based approach” [6] in which it is claimed that problems are fully specifiable and that optimal and predictable solutions exist for every problem. The “traditionalists” are said to advocate extensive upfront planning, codified processes, and rigorous reuse to make development an efficient and predictable activity [7]. By contrast, agile processes address the challenges of the increasingly complex nature of software development by relying on people and their creativity rather than on formalized processes [6]. The goal of optimization is being replaced by those of flexibility and responsiveness [8]. Ericksson et al. [9] define agility as follows: *agility means to strip away as much of the heaviness, commonly associated with the traditional software-development methodologies, as possible to promote quick response to changing environments, changes in user requirements, accelerated project deadlines and the like.* (p. 89)

The Role of Agility in Distributed Development

Global software development has matured considerably since its inception and has become an integral part of the information technology landscape. Now, rather than deciding whether or not to get involved in global sourcing, many companies are facing decisions about whether or not to apply agile methods in their distributed projects. These companies are often motivated by the opportunities of solving the coordination and communication difficulties [4] associated with global software development.

Empirical evidence from case studies conducted by Paasivaara and Lassenius [10], and Holmström, Fitzgerald et al. [11] show successful implementation of agile values and principles in different globally distributed projects. This motivates assessing the viability of agile practices for distributed software development teams. The interest in becoming agile and distributed is also illustrated by the increasing number of research publications and seminars devoted to the topic.

Implementing Agility Across Time and Space

Despite the increased attention, merging the two strategies is no easy task due to significant differences in fundamental principles of agile and distributed development approaches. In particular, while agile principles prescribe close interaction and co-location, the very nature of distributed software development does not support these prerequisites. Taylor, Greer et al. [12] claim that distributed agile software development suffers substantial difficulties because of its complex development environment and there is little empirical evidence describing actual development experiences. The lack of clear understanding of who, what, when, why and how in agile

distributed development motivated us to collect experiences from various companies that had started, and also benefitted from, becoming agile and distributed.

Aims of the Book

The idea to write a book on agile and distributed software development gradually evolved as the critical mass of questions related to merging seemingly incompatible approaches emerged. The questions that the authors aimed to answer with this book include:

- What shall companies expect from merging agile and distributed strategies?
- What are the stumbling blocks that prevent companies from reaching the agile benefits in distributed environment, and how to recognize unfeasible strategies and unfavorable circumstances?
- What helps managers cope with the challenges of implementing agile approaches in distributed software development projects?
- How can distributed teams survive the decisions taken by the management and become efficient through the application of agile approaches?

Book Overview

This book consists of five parts.

1. In the **Motivation** part the editors introduce the fundamentals of agile distributed software development and explain the rationale behind the application of agile practices in globally distributed software projects.

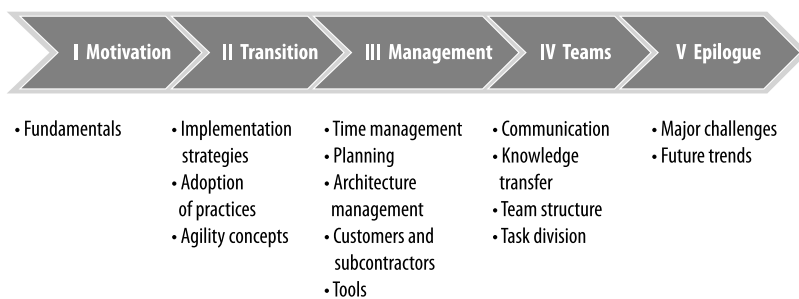


Fig. 1 Book layout

2. The second part of the book is called **Transition**. Here we have gathered seven chapters that discuss the transition to being agile and distributed. The chapters describe implementation strategies, adoption of particular agile practices for distributed projects, and general concepts of agility.

3. The third part of the book, **Management**, focuses on managerial aspects and decisions in agile distributed software projects. Practical implications for project planning, time management, customer and sub-contractor interaction, tool support and architecture-centric development are presented in eight chapters.
4. The fourth part is devoted to agile and distributed **Teams**. Here we have collected six chapters that provide in-depth hands-on advice for the team members and their managers. Topics discussed include agile distributed team configuration, effective communication and knowledge transfer, the role of architecture in task division, and allocation of roles and responsibilities.
5. finally, in the **Epilogue** we summarize the contributions of the different chapters and present results from a Delphi-inspired study that highlights the major areas of concern and future trends for research and practice in agile distributed development.

Most of the chapters in this book offer practical advice based on experiences obtained in and from the industry. These experiences are collected through personal observations of practitioners, empirical research in particular studied contexts or extensive continuous observations gained from various sources.

Target Audience

This book is primarily targeted at practitioners (managers and team members) involved in globally distributed software projects - those who are practicing agile methods and those who are not. We believe that it will serve as a useful source of practical advice, which are based on the real life examples of application of agile practices in distributed development, and will hopefully motivate companies to try improving their sourcing strategies by adopting best practices and benefits that agile promises.

Many book chapters are based on the sound empirical research and identify gaps and commonalities in the existing state-of-the-art and state-of-the-practice. We thus believe that our book can be also of relevance and interest for the academic audience, in particular, researchers working in the field, as well as lecturers and students of global agile software development.

References

1. Ó. Conchúir, E., Ågerfalk, P. J., Fitzgerald, B., & Holmström Olsson, H. (2009). Global software development: Where are the benefits?. *Communications of the ACM*, 52(8), 127–131.
2. Ågerfalk, P. J., Fitzgerald, B., Holmström, H., Lings, B., Lundell, B., & Ó. Conchuir, E. (2005). A framework for considering opportunities and threats in distributed software development. In *Proceedings of the international workshop on distributed software development (DiSD)* (pp. 47–61). Vienna: Austrian Computer Society.
3. Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9–10), 833–859.

4. Ågerfalk, P. J., & Fitzgerald, B. (2006). Flexible and distributed software processes: Old petunias in new bowls? *Communications of the ACM*, 49(10), 26–34.
5. Beck, K., et al. (2001). Agile manifesto. Available online. <http://agilemanifesto.org>. Cited 15 Feb 2010.
6. Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 72–78.
7. Boehm, B. (2002). Get ready for agile methods, with care. *IEEE Computer*, 35(1), 64–69.
8. Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies—The traditional goal of optimization and control is making way for learning and innovation. *Communications of the ACM*, 50(3), 79–83.
9. Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: The state of research. *Journal of Database Management*, 16(4), 88–100.
10. Paasivaara, M., & Lassenius, C. (2006). Could global software development benefit from agile methods? In *International conference on global software engineering* (pp. 109–113).
11. Holmström, H., Fitzgerald, B., Ågerfalk, P. J., & Ó. Conchuir, E. (2006). Agile practices reduce distance in global software development. *Information Systems and Management*, 23(3), 7–18.
12. Taylor, P. S., Greer, D., Sage, P., Coleman, G., McDaid, K., & Keenan, F. (2006). Do agile GSD experience reports help the practitioner? In *Proceedings of the 2006 international workshop on global software development of ACM* (pp. 87–93).

*Darja Šmite
Nils Brede Moe
Pär J. Ågerfalk*

Agility Across Time and Space

Implementing Agile Methods in Global Software Projects

Šmite, D.; Moe, N.B.; Ågerfalk, P.J. (Eds.)

2010, XXXVI, 341 p., Hardcover

ISBN: 978-3-642-12441-9