

# Chapter V

## Agile IT Outsourcing

**Boris Roussev**

*University of the Virgin Islands, USA*

### ABSTRACT

*Agile methods are lightweight, iterative software development frameworks used predominantly on small and mid-sized software development projects. This chapter introduces a project structure and management practices creating agile conditions for large software projects outsourced either offshore or onshore. Agility is achieved by slicing a large project into a number of small projects working in agile settings. Development is divided into research and development activities that are located on-site, and production activities located off-site. The proposed approach makes agile methods applicable to the stressed conditions of outsourcing without compromising the quality or pace of the software development effort. Creating an agile environment in an outsourcing project relies on maintaining a balance between the functions and sizes of on-site and off-site teams, on redefining the developers' roles, and on reorganizing the information flow between the different development activities to compensate for the lack of customers on-site, team colocation, and tacit project knowledge.*

### INTRODUCTION

We live in a digital world where any activity not requiring a physical presence can be outsourced to any place that is connected to the Internet. Even the term *physical presence* comes with a qualification. Information and communication technologies enable cooperation in a distributed mode. Technologies, such as groupware and videoconferencing, are increasingly becoming

feasible for organizations to use in distributed projects.

Advances in ICT have been essential for loosening the spatial constraints on software development. The largely digital nature of software development allows changing its geography of provision. The combination of low labor costs, technological sophistication, satisfactory (but not outstanding) project management skills, and successful software establishment makes south

Asia a particularly attractive location for software production outsourcing (Dossani & Kenney, 2003). However, the decision to outsource software production is not a simple matter of upside (Natovich, 2003). Product and project managers have been looking for ways to mitigate risks in outsourcing projects to maximize reward (Craumer, 2002). Even though the total contract value of IT outsourcing transactions continues to increase (Hale, Souza, Lo, Adachi, & Babaie, 2005; Technology Partners International [TPI], 2005), many large organizations are bringing key IT projects back in house because outsourcing contracts have failed to meet expectations (Huber, 2005).

Agile methods (Agile Alliance, 2001; Beck, 1999; Schuh, 2005; Sridhar, Mahapatra, & Mangalaraj, 2005) are popular software development processes designed for use on small to mid-sized software projects. They are based on the notion that object-oriented software development is not a rigidly defined process, but an empirical one that may or may not be repeated with the same success under changed circumstances.

Agile methods are based on four critical values: simplicity, communication, feedback, and courage, informing a set of key practices (Pollice, 2004), which will be considered in more detail later. Boehm and Turner (2004, p. 27) define agile methods as “very light-weight processes that employ short iterative cycles; actively involve users to establish, prioritize, and verify requirements; and rely on tacit knowledge within a team as opposed to documentation.” However, many of the agile practices seem to be incompatible with the context of outsourcing, especially when outsourcing offshore; for example, agile practices work better when there are customers on-site, team colocation, short life cycles, and the value of embracing change. Above all, agile methods are applicable mainly to small and mid-sized projects because of many of the characteristics of agile development. For example, smaller, colocated teams require fewer lines of communication and less coordination efforts, can use less formal development

processes, can reach an agreement faster, and can adapt to change more rapidly (Schuh, 2005; Stephens & Rosenberg, 2003).

In the past, there have been several attempts to reproduce the conditions for agility in large projects. To the best of our knowledge, no such attempt has been made for outsourcing projects. Pollice (2001a, 2001b), Evans (2004), and Boehm and Turner (2004) propose to scale up agile methods by balancing agility and discipline. Pollice and Evans, for instance, look for common ground between agile and RUP (Jacobson, Booch, & Rumbaugh, 1999), while Boehm and Turner try to get the best of both agile and plan-driven (waterfall) worlds. In contrast, Kruchten (2004) proposes to scale down large projects to meet the agile “sweet spot” based on experience reported in Brownsword and Clements (1996) and Toth, Kruchten, and Paine (1993). The sweet spot, or in other words the ideal agile context, is characterized by a small team of developers sharing common values, team colocation, customers on-site, and a short life cycle.

In this work, we show how to reengineer large outsourcing projects to benefit from the sweet spot of agile methods while avoiding its “bitter spot.” The proposed approach makes agile methods applicable to the stressed condition of both offshore and onshore outsourcing environments without compromising the quality of the software development effort. Creating a context congenial to agile methods hinges on balancing the functions and sizes of on-site and off-site teams, on redefining the developers’ roles, and on reorganizing the information flow between the different development activities.

The rest of the chapter is structured as follows. Next it examines some critical issues in outsourcing software development activities. Then it presents the structure of the agile outsourcing project. The chapter then elaborates on the inception and architectural activities in agile outsourcing, which are crucial to slicing large projects into multiple, relatively independent agile

15 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: [www.igi-global.com/chapter/agile-outsourcing/8758](http://www.igi-global.com/chapter/agile-outsourcing/8758)

## Related Content

---

### Design and Execution of Dynamic Collaborative Learning Experiences

Nicola Capuano, Sergio Miranda, Pierluigi Ritrovato, Giuseppina Rita Mangione and Anna Pierri (2013). *International Journal of e-Collaboration* (pp. 26-41).

[www.irma-international.org/article/design-execution-dynamic-collaborative-learning/75211](http://www.irma-international.org/article/design-execution-dynamic-collaborative-learning/75211)

### COBRAS: Cooperative CBR Bibliographic Recommender System

Hager Karoui (2009). *Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling* (pp. 184-202).

[www.irma-international.org/chapter/cobras-cooperative-cbr-bibliographic-recommender/6642](http://www.irma-international.org/chapter/cobras-cooperative-cbr-bibliographic-recommender/6642)

### An Empirical Investigation of Two Group Idea Generation Techniques: Manual Versus Electronic Gallery Writing

Kaushik Ghosh and Milam Aiken (2013). *International Journal of e-Collaboration* (pp. 61-77).

[www.irma-international.org/article/empirical-investigation-two-group-idea/77846](http://www.irma-international.org/article/empirical-investigation-two-group-idea/77846)

### Stepping into the Role of Professional Writer

Christine Aikens Wolfe, Cheryl North-Coleman, Shari Wallis Williams, Denise Amos, Glorianne Bradshaw and Toby Emert (2009). *Handbook of Research on Electronic Collaboration and Organizational Synergy* (pp. 250-260).

[www.irma-international.org/chapter/stepping-into-role-professional-writer/20178](http://www.irma-international.org/chapter/stepping-into-role-professional-writer/20178)

### Collaborative Computing-Based K-Nearest Neighbour Algorithm and Mutual Information to Classify Gene Expressions for Type 2 Diabetes

Sura Zaki Al Rashid (2022). *International Journal of e-Collaboration* (pp. 1-12).

[www.irma-international.org/article/collaborative-computing-based-k-nearest-neighbour-algorithm-and-mutual-information-to-classify-gene-expressions-for-type-2-diabetes/304044](http://www.irma-international.org/article/collaborative-computing-based-k-nearest-neighbour-algorithm-and-mutual-information-to-classify-gene-expressions-for-type-2-diabetes/304044)