Chapter 2.25
Agile Outsourcing to India:
Structure and Management

Boris Roussev
University of the Virgin Islands, USA

Ram Akella
University of California, USA

ABSTRACT

The combination of low labor costs, technological sophistication, project management skills, and successful software establishment makes India a particularly attractive location for software production outsourcing. Furthermore, in most situations, information and communication technologies render virtual presence practically equivalent to physical presence, thus enabling effective communication and cooperation in a distributed mode. This chapter introduces a project structure creating agile conditions for large outsourcing software projects. The agility advantage is achieved by scaling down a large project into a number of small-sized projects working in agile settings. We divide the work into R&D activities, located onsite, and production activities, located offsite. The proposed approach makes Agile applicable to the stressed condition of outsourcing environments without compromising the quality and the pace of the software development effort. Creating a context congenial to agile methods hinges on maintaining a good balance between the functions and sizes of onsite and offsite teams, on redefining the developers' roles, and on reorganizing the information flow between the different development activities to compensate for the lack of customer onsite, team co-location, 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. Even the term "physical presence" comes with a qualification. Information and communication technologies (ICTs) enable cooperation in a distributed mode. Technologies, such as groupware and video-conferencing, are increasingly becoming feasible for

organizations to use in international projects. In addition, these technologies are changing the way we perceive presence and absence. The largely digital nature of software development allows changing its geography of provision. Advances in ICT have been essential for loosening the spatial constraints on software development.

The combination of low labor costs, technological sophistication, project management skills, and successful software establishment makes India a particularly attractive location for software production outsourcing. From 1996-1997 to 2003-2004, the export software sales of the Indian IT industry had an average annual growth rate of 38.7% to reach a total of US$12.2 billion in 2003-2004 (Nasscom, 2004).

Even though India has had a qualified labor pool and the enabling technologies, along with the great pressure exerted on firms in developed nations to lower costs, the Indian software industry still operates in the low value-added segments, typically in applications development, testing, and maintenance, while the high-end work, such as developing the IT strategy, building the software architecture, designing the system, integrating the project with enterprise packages, and designing custom components are all discharged by firms in developed countries (Dossani & Kenney, 2003).

Agile methods (Beck, 1999) are popular software development processes designed to be used on small- to mid-sized software projects. The Agile movement started in the mid-1990s. The first major project to apply an agile method was the Chrysler Comprehensive Compensation system, a payroll system developed in 1996. This method, called extreme programming, was described in Beck (1999) and became the foundation for the Agile Alliance Manifesto (Agile Alliance, 2001).

Agile methods 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 in four critical values—simplicity, communication, feedback, and courage—informing a set of key practices (Pollice, 2004), which will be considered later on. Boehm and Turner (2004) 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."

Many of the agile practices are incompatible with the context of outsourcing, for example, customer onsite, team co-location, short lifecycle, and embracing change. But above all, agile methods can be applied only to small-sized projects.

In the past, there have been several attempts to reproduce the conditions for agility in large-sized projects. To the best of our knowledge, no such attempt has been made for outsourcing projects. Pollice (2001), Evans (2004), and Boehm and Turner (2004) propose to scale up agile methods by balancing agility and discipline. Pollice and Evans, for instance, look out for common grounds 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 (1994).

In this chapter we show how to reengineer large-sized (and small-sized) outsourcing projects to benefit from the "sweet spot" of Agile, while avoiding its "bitter spot." The proposed approach makes Agile applicable to the stressed condition of outsourcing environments, without compromising the quality of the software development effort. Creating a context congenial to agile methods hinges on maintaining a good balance between the functions and sizes of onsite and offsite teams, on redefining the developers' roles, and on reorganizing the information flow between the different development activities.

Related Content