Journal of Database Management
Volume 27 • Issue 1 • January-March 2016

Agility Facilitators for Contemporary Software Development

Dinesh Batra, Florida International University, Miami, FL, USA
Weidong Xia, Florida International University, Miami, FL, USA
Shekhar Rathor, Florida International University, Miami, FL, USA

ABSTRACT

Agile software development generally refers to popular practices that are supposed to adhere to the Agile Manifesto with its values and principles. Empirical studies on agile software development have mainly focused on organizational adoption and impacts of agile practices. Furthermore, the literature on agile software development has mostly centered on small, co-located projects. However, agility is needed for software development projects of varied sizes in different organizations across industries. The general nature of agile values and principles and the procedure-driven nature of specific agile methods make it difficult for organizations to determine what they can do to effectively facilitate agility in their software development process. To bridge that literature gap and based on an evolved grounded-theory approach, this study identifies nine agility facilitators and their corresponding dimensions that extend beyond small, co-located projects to software projects of any size and distribution. These agility facilitators are further grouped into two categories: organizational foundation and project processes. In addition, the authors identify four dimensions of agility. The authors propose a framework that describes the organizational mechanisms through which the nine categories of facilitators lead to software development agility.

KEYWORDS

Agile Manifesto, Agile Principles, Agile Values, Agility Facilitators, Software Development Agility

INTRODUCTION

The failure of plan-driven waterfall-based methods in software projects experiencing significant uncertainty and frequent changes in requirements prompted the adoption of the Agile Manifesto (Highsmith & Cockburn, 2001) and the proposal of several agile methods such as Scrum (Schwaber, 2004) and XP (Beck, 2000). Agility is an organization's ability to sense and respond swiftly to technical changes and new business opportunities (Lyytinen & Rose, 2005). Agile software development attempts to strip away as much of the heaviness, commonly associated with traditional software-development methodologies, as possible to promote quick response to changing environments, changes in user requirements, and accelerated project deadlines (Erickson, Lyytinen, & Siau, 2005).

Based on an extensive survey, Dyba & Dingsoyr (2008) concluded that most projects using agile methods are small and use XP, although recent research suggests that Scrum has become more popular (Hossain, Babar, & Paik, 2009; VersionOne, 2015; Vlietland & van Vliet, 2015). The question of

DOI: 10.4018/JDM.2016010101

how to select agile methods for projects that are not small and that encounter significant changes is still unanswered (Batra, Xia, VanderMeer, & Dutta, 2010). Recent research has focused on specific factors that affect agility such as communication (Hummel, Rosenkranz, & Holten, 2015; Korkala & Maurer, 2014), customer involvement (Hoda, Noble, & Marshall, 2011; Matook & Maruping, 2014), and self-organization (Hoda, Noble, & Marshall, 2013). However, there is a lack of systematic and comprehensive interpretation of agility facilitators that can be applied across projects of varied sizes. Furthermore, because of the lack of detailed interpretation of agility facilitators, it is difficult to assess organizational readiness and changes needed to successfully adopt agile practices.

The concern for size scalability is just one of the factors that affects agility in contemporary software development. The issue of agility permeates beyond the confines of one team working on a single project. Contemporary software development entails considerable innovation, discovery, change sensing, and change responding (Conboy, 2009; Vidgen & Wang, 2009) while addressing various people, project, process, and institutional factors such as project scope and size, stakeholder participation, resources, technology, and outsourcing (McLeod & MacDonell, 2011). The agility question often needs to be dealt with not only at the project level, but also at the organizational level. For example, the agile principle of employing only motivated individuals may be feasible for a specific project but it cannot be extended to organization-wide projects because we need to assume that participating individuals, for the most part, have average (but not necessarily low) motivation, which may be adequate for most projects (Batra, VanderMeer, & Dutta, 2011). Similarly, face-to-face communication may be feasible in a small, co-located project, but we should consider diverse communication modes for a larger project that may have an outsourced development element (Korkala & Maurer, 2014; Moore & Barnett, 2004).

The Agile Manifesto is based on the following four values: (1) individuals and interactions over processes and tools, (2) working software over comprehensive documentation, (3) customer collaboration over contract negotiation, and (4) responding to change over following a plan. The Agile Manifesto gives stronger emphasis to one set of values over the second set but was not intended to give no or little weightage to the second set of values. Agile Manifesto is frequently read in such a way that the things on the right, which are items commonly found in too many plan-driven, contractually-driven, standards, and audit-driven environment, are not merely valued less, they effectively have zero value (Glazer, Dalton, Anderson, Konrad, & Shrum, 2008). Because larger projects are contractually-driven (Gefen, Wyss, & Lichtenstein, 2008), the items on the right cannot be ignored and must be explicitly or implicitly incorporated in an agility framework. Given that agile development is recommended if the project is small with fewer than ten developers (Beck, 2000), a project having more than about ten or fifteen members should be considered as larger (or not small). To gain a better understanding of agility facilitators, empirical studies should be conducted with projects that vary on several attributes.

The term "agile" refers to encompassing characteristics as described by the values and principles in the agile manifesto as well as the practices embedded in popular agile methods (Highsmith & Cockburn, 2001). Based on Conboy (2009) taxonomy, we consider "agility" as the creation of or reaction to changes in user requirements for a software development project. A detailed understanding of agility constructs in software development projects (including larger projects) is possible by using an evolved grounded theory approach, which can be employed "to explore areas not yet thoroughly researched" and "to discover relevant variables that later can be tested through quantitative forms of research" (Corbin & Strauss, 2014).

Using an evolved grounded theory approach (Corbin & Strauss, 2014), this study attempts to answer the following two exploratory questions: What are the key organizational factors that facilitate software development agility? What are the specific dimensions of each of the agility facilitators? Our study is motivated by the challenges organizations experience in understanding and managing software development agility facilitators beyond the general agile values and principles as defined by the Agile Manifesto and the procedure-driven agile methods that tend to be limited to small and collocated software development projects.