RESEARCH NOTE

Extending Agile Principles to Larger, Dynamic Software Projects: A Theoretical Assessment

Dinesh Batra, Florida International University, USA

Debra VanderMeer, Florida International University, USA

Kaushik Dutta, National University of Singapore, Singapore

ABSTRACT

The article evaluates the feasibility of extending agile principles to larger, dynamic, and possibly distributed software development projects by uncovering the theoretical basis for agile values and principles for achieving agility. The extant literature focuses mainly on one theory – complex adaptive systems – to support agile methods, although recent research indicates that the control theory and the adaptive structuration theory are also applicable. This article proposes that at least three other theories exist that are highly relevant: transaction cost economics, social exchange theory, and expectancy theory. By employing these theories, a rigorous analysis of the Agile Manifesto is conducted. Certain agile values and principles find theoretical support and can be applied to enhance agility dynamic projects regardless of size; some agile principles find no theoretical support while others find limited support. Based on the analysis and the ensuing discussion, the authors propose a framework with five dimensions of agility: process, design, people, outcomes, and adaptation.

Keywords: *Adaptive Structuration, Agile Manifesto, Agile Principles, Complex Adaptive Systems, Control Theory, Expectancy Theory, Social Exchange, Transaction Cost Economics*

INTRODUCTION

As business and technology environments change at an unprecedented rate, software development agility to respond to changing user requirements has become increasingly critical for software development performance (Lee & Xia, 2010). Software development agility is the ability of an information system development (ISD) method to create change, or proactively, reactively, or inherently embrace change in a timely manner, through its internal components and relationships with its environment (Conboy, 2009). Agility is an organization's ability to sense and respond swiftly to technical changes and new business opportunities (Lyytinen & Rose, 2006). At its core, agility means to strip away as much of the heaviness, commonly associated with traditional software-development

DOI: 10.4018/jdm.2011100104

methodologies, as possible to promote quick response to changing environments, changes in user requirements, and accelerated project deadlines (Erickson, Lyytinen, & Siau, 2005). In response to the need for agility, lightweight agile software development methods have emerged as alternatives to process-heavy plan-based methodologies as organizations seek to deliver software more quickly (Abrahamsson, Conboy, & Wang, 2009), while simultaneously ensuring that the delivered software is of high quality and is closely aligned to the needs of the customer (Larman, 2003).

The call for such methods arose in 2001, with the publication of the Agile Manifesto (http://agilemanifesto.org), which has remained unchanged in a decade even as several agile methods have been proposed. The manifesto is based on four values: "individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan." These values are accompanied by a set of twelve agile principles that provide guidance toward agile practice in development.

The manifesto was written by a group of practitioners interested in bringing together a number of lightweight methodologies, most of which now fall into the agile camp (Boehm & Turner, 2003; Qumer & Henderson-Sellers, 2008), including Scrum (Schwaber & Beedle, 2002), Extreme Programming or XP (Beck, 2000), Adaptive Software Development (Highsmith, 1999), and others. The agile movement grew out of practitioners' impatience with heavier, plan-based methods, and their belief that there must be a better way. Indeed, the use of the word "manifesto," a highly-charged word associated with revolutionary change, in the title was probably intentional – the authors wanted to highlight the radical differences between their agile methods and traditional plan-based approaches.

A decade later, the impact of the Agile Manifesto and its associated ideas is clear: agile methods have taken their place alongside more traditional approaches and are widely used (McAvoy & Butler, 2009). Thousands of practitioners have signed their names in support of the Agile Manifesto (http://agilemanifesto.org/sign/display.cgi), while a 2008 survey by Dr. Dobb's Digest suggests that up to 69% of responding organizations have adopted agile methods in some form, from pilot projects to full deployment of agile methods, and that respondents believe that their use of agile methods result in higher quality deliverables, more productive developers, and more satisfied stakeholders (Ambler, 2008).

However, there is evidence that supports the widely-held view that agile development has been applied only to small projects (Henderson-Sellers & Serour, 2005). Dyba and Dingsoyr (2008) present an extensive review of agile case study reports in the literature. Of the 33 projects referenced in this study, only four project teams had 20 or more members, and only one project team had a size greater than 23, at 60 members. Chow and Cao (2008) examined critical success factors in 109 agile projects. Of these projects, nearly 80% of project teams had fewer than 20 members. The Scrum methodology recommends projects teams of no more than six members (Schwaber, 2005). Beck (2005) states: "You probably couldn't run an XP project with a hundred programmers. Not fifty. Nor twenty, probably. Ten is definitely doable."

It is widely accepted that agile development does, indeed, enhance agility although the supporting evidence is generally restricted to small projects (Boehm & Turner, 2003; Turk, France, & Rumpe, 2005). However, small projects account for only 17% of programming code (Beck & Boehm, 2003). Even free open-source software can be large and significant enough to disrupt commercial incumbents (Brydon & Vining, 2008).

Nevertheless, agile values and principles can provide a starting point to examine agility for large, dynamic projects. A few studies (e.g., Lindvall et al., 2004; Eckstein, 2004; Cao, Mohan, Xu, & Ramesh, 2009; Heeager & Nielsen, 2009; Batra, Xia, VanderMeer, & Dutta, 2010) have examined attempts to introduce agile methods in large development organizations.