Information and Software Technology 56 (2014) 911–921

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Understanding agile software development practices using shared mental models theory

Xiaodan Yu [a,*], Stacie Petter [b,1]

[a] School of Information Technology and Management, University of International Business and Economics, Boxue Building 1311, No. 10 Huixin Dongjie, Chaoyang District, Beijing 100029, China
[b] College of Information Science & Technology, University of Nebraska at Omaha, 6001 Dodge Street, PKI 173B, Omaha, NE 68182, USA

ARTICLE INFO

Article history:
Received 24 May 2013
Received in revised form 24 February 2014
Accepted 25 February 2014
Available online 6 March 2014

Keywords:
Agile software development
Agile practices
Shared mental models theory
Extreme programming
Scrum

ABSTRACT

Context: Agile software development is an alternative software development methodology that originated from practice to encourage collaboration between developers and users, to leverage rapid development cycles, and to respond to changes in a dynamic environment. Although agile practices are widely used in organizations, academics call for more theoretical research to understand the value of agile software development methodologies.
Objective: This study uses *shared mental models* theory as a lens to examine practices from agile software methodologies to understand how agile practices enable software development teams to work together to complete tasks and work together effectively as a team.
Method: A conceptual analysis of specific agile practices was conducted using the lens of shared mental models theory. Three agile practices from Xtreme Programming and Scrum are examined in detail, system metaphor, stand-up meeting, and on-site customer, using shared mental models theory.
Results: Examining agile practices using shared mental models theory elucidates how agile practices improve collaboration during the software development process. The results explain how agile practices contribute toward a shared understanding and enhanced collaboration within the software development team.
Conclusions: This conceptual analysis demonstrates the value of agile practices in developing shared mental models (i.e. shared understanding) among developers and customers in software development teams. Some agile practices are useful in developing a shared understanding about the tasks to be completed, while other agile practices create shared mental models about team processes and team interactions. To elicit the desired outcomes of agile software development methods, software development teams should consider whether or not agile practices are used in a manner that enhances the team's shared understanding. Using three specific agile practices as examples, this research demonstrates how theory, such as shared mental models theory, can enhance our understanding regarding how agile practices are useful in enhancing collaboration in the workplace.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Agile software development is a lightweight, incremental software development method with specific practices that emphasize close interaction with customers. One benefit of agile software development practices is the ability for software development teams to adapt to changing requirements from customers while identifying and reducing certain risks that arise during software development [1]. Yet, the full benefits of agile software development are not fully understood [2], and research examining agile software development as an alternative to traditional methods is sparse [1]. Although agile software development methods were created by software developers in response to frustration with traditional software development approaches, critics argue that "there is little scientific support for many of the claims made by the agile community; and [agile practices] are rarely applicable, and are rarely applied by the book" [3, p. 836]. These issues have been a concern and criticism of agile software development among researchers, and sometimes practitioners, because agile software development was developed out of practice with no theoretical foundation. Only a few studies have explored the theoretical

[*] Corresponding author. Tel.: +86 10 6449 5140.
E-mail addresses: yxd.xiaodanyu@gmail.com (X. Yu), spetter@unomaha.edu (S. Petter).
[1] Tel.: +1 402 554 2077.

http://dx.doi.org/10.1016/j.infsof.2014.02.010
0950-5849/© 2014 Elsevier B.V. All rights reserved.

underpinnings of agile software development [4]. As organizations increase their reliance on agile software development, exploring the tenets of agile software development methods using theory can yield insights to understand how agile software development methods provide value to software development teams, such as explaining how each agile practice contributes to higher levels of collaboration among the software development team and customers.

Therefore, this study seeks to answer the following question: "How can theory be applied to agile software practices to explain how agile practices enable higher levels of collaboration during software development?" To answer this question, we apply a theory from cognitive psychology known as shared mental models. The literature on agile software development and the literature on shared mental models have an overlapping interest in improving team performance. The theory of shared mental models investigates the role of shared understanding on team performance and has prescribed methods for fostering shared mental models. Agile practices seek to enhance the software development process through improved interaction among software development teams and customers. Why is improved interaction needed among development teams and customers? What aspects of interactions should be emphasized among software development teams and customers? How does increased interaction improve collaboration during software development? These questions remain unclear. In this paper, we propose that shared mental models theory provides a theoretical foundation to explain the importance of agile practices for collaboration by explicating how agile practices enhance the team's shared understanding. Through this application of shared mental models theory, we offer an exemplar of how theory can explain one the benefits of using agile practices, higher levels of collaboration.

This paper is organized as follows. First, we explain why the theory of shared mental models is applicable to agile software development by reviewing background information on the related domains: agile software development and shared mental models. Next, in the context of three agile practices, we examine agile practices and their value in the context of shared mental models. This examination of agile practices in the context of shared mental models theory explains how the agile practices offer benefits to the software development team. We conclude by offering recommendations for future research and highlighting the contributions of this research.

2. Background

2.1. Agile software development

In the late 1990s, agile software development methods emerged as a response to the challenges associated with traditional software development methods, which are often perceived as inflexible and unable to address changing requirements from users [5]. Agile is formally defined by Conboy [6] as "the continual readiness of an ISD (information systems development) method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment" [6, p. 340]. Many agile methodologies have been developed [7], with Scrum and Extreme Programming (XP) considered as two of the most widely used agile methods. Each agile methodology is comprised of its own specified practices with varied focuses. For example, XP consists of practices that focus on software development team activities, while Scrum consists of practices to improve project management by quickly exposing risks within the project [7].

The value of agile software development over traditional methods is a focus on people's interactions (developers and users) within a project as one of the primary drivers of success [8]. Although the Standish Group's reports can be controversial [9], this widely cited group has stated that "software applications developed through the agile process have three times the success rate of the traditional waterfall method and a much lower percentage of time and cost overruns" [10, p. 25]. The promise of agile practices is that these methods offer the potential to enable software development teams to adapt to customer's changing requirements through high levels of interaction and collaboration, which can lead to better project outcomes.

Despite the popularity of agile software development, organizations struggle to adopt agile practices. When organizations choose to adopt agile practices, but encounter challenges in adopting agile practices, the general tendency is to abandon the practice [11]. Yet, when agile practices are abandoned or implemented improperly, organizations tend to struggle to develop and maintain collaboration throughout the project [12]. Further, given the large number of different agile software development methodologies and practices, it is common for organizations to adapt a few agile practices rather than implement a full agile software development methodology. Understanding the virtue of each agile software development practice within a methodology is a necessary step toward a successful implementation of agile software development. Yet empirical research has found the perceived benefits of agile software development are not fully understood in research or in organizations [13]. The primary factors that influence the adoption of agile software development among organizations are subjective norms and training [2]. Studies by agile practitioners suggest that only half of organizations that state that they use agile methods actually meet the criteria that define agile software development [14].

By applying theory to agile practices, one is able to understand the value of agile practices as techniques to increase collaboration within the software development team and among the software developers and customers.

2.2. Shared mental models

Shared mental models is a theory from cognitive psychology that focuses on the thought processes or activities that occur at a team level. While heterogeneity of team members can strengthen a team by leveraging diversity, shared mental models theory proposes that effective teams need to maintain a shared understanding within the team [15]. Without some commonality among the team members in terms of understanding the tasks and relationships within the team, accomplishing team goals would be nearly impossible.

Shared mental models was developed based on mental model theory; mental model theory was first proposed by Johnson-Laird to account for individual reasoning [16]. Mental model theory proposed that an individual's mind holds representations of anything that an individual has encountered in the physical world. Such representations are abstract and accessible by individuals when learning new knowledge or solving problems [17].

The theory of shared mental models extends the idea of an individual's mental model to conceptualize teams as a unified information processing unit. A shared mental model is defined as the "knowledge structures held by members of a team that enable them to form accurate explanations and expectations for the task, and, in turn, to coordinate their actions and adapt their behavior to demands of the task and other team members" [15, p. 228]. Shared mental models provide the team with an internal knowledge base that allows team members to decide what actions to take when novel events happened. Through shared mental models, teams form

Download English Version:

https://daneshyari.com/en/article/550567

Download Persian Version:

https://daneshyari.com/article/550567

Daneshyari.com