Contents lists available at ScienceDirect

Information and Software Technology

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

The impact of inadequate and dysfunctional training on Agile transformation process: A Grounded Theory study

Taghi Javdani Gandomani[a,*], Hazura Zulzalil[a], Abdul Azim Abdul Ghani[a], Abu Bakar Md. Sultan[a], Reza Meimandi Parizi[b]

[a] *Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*
[b] *School of Computing and IT, Taylor's University, 47500 Subang Jaya, Selangor, Malaysia*

ARTICLE INFO

Article history:
Received 3 May 2013
Received in revised form 15 May 2014
Accepted 15 May 2014
Available online xxxx

Keywords:
Agile software development
Agile transformation process
Agile training
Agile transition
Grounded Theory

ABSTRACT

Context: Training is an essential facilitator in moving from traditional to Agile software development.
Objective: This paper addresses the importance of adequate and functional training in Agile transformation process, the causes of inadequate and dysfunctional training, and the heuristic strategies that can be used in software companies for dealing with this phenomenon.
Method: A Grounded Theory study was conducted with participation of 35 Agile experts from 13 different countries.
Results: This research discovered that *inadequate and dysfunctional training* was one of the critical issues that affected Agile transformation process. This study shows that comprehensive and functional training is not often provided to support Agile transformation. This paper shows the primary causes of inadequate and dysfunctional training, its adverse consequences on the transformation process, and *the heuristic and ad-hoc treatments* as the strategies used by Agile teams to cope with this challenge.
Conclusion: Comprehensive training is important in Agile transformation process. Inadequate and dysfunctional training causes several challenges and problems for software companies and development teams when moving to Agile. Several ad-hoc strategies identified by this study can be employed to help software teams and companies facing similar problems.

1. Introduction

Disciplined software methodologies such as Waterfall, Spiral, and RUP, called traditional methods or heavyweight methods, propose highly disciplined development life cycles. Their proposed development models are based on a sequential series of activities and steps that have to be well defined and documented in detail. Because of this structure, comprehensive up-front planning, detailed documentation, and expensive design are the necessary requirements of these methods. Furthermore, defining a stable set of customers' requirements at the inception of the project is very important [1]. However, these methods are often associated with many challenges such as inaccurate understanding of requirements in early stages, poor communications, inability to accept any change, etc. [1].

Unlike traditional methods, Agile methods in software development embrace change in user requirements and emphasize the role of people more than processes in software development [2]. Agile methods provide different life cycles, roles, and activities compared to the traditional methods.

Leaving traditional software methodologies and moving to Agile methods, called Agile transformation or transition process (ATP), is an organizational effort to overcome the inherent problems and challenges of traditional software development methods. This process tries to make software development activities and required roles compatible with Agile approach. It also affects almost all aspects of an organization and acts as revolution in software development process [3]. ATP is about being Agile rather than doing Agile. That is, not only development process and activities but also people's roles, responsibilities, and mindsets have to change and adapt to Agile approach. For instance, an Agile team should be self-organized, hence, all team members should collaborate in decision-making; however, this is very difficult in practice [4]. Sometimes project managers do not like to relinquish their authority and sometimes team members do not like to accept such

[*] Corresponding author. Tel.: +60 172759054; fax: +60 389466576.
E-mail addresses: taghi@upm.edu.my, taghi.javdani@gmail.com (T. Javdani Gandomani), hazura@upm.edu.my (H. Zulzalil), azim@upm.edu.my (A.A. Abdul Ghani), abakar@upm.edu.my (A.B. Md. Sultan), rezameimandi.parizi@taylors.edu.my (R. Meimandi Parizi).

http://dx.doi.org/10.1016/j.infsof.2014.05.011
0950-5849/© 2014 Elsevier B.V. All rights reserved.

responsibilities, because they do not have enough confidence in themselves. A true ATP is also about changing people's perceptions/attitudes/behaviors in software companies [5,6]. Therefore, ATP encompasses all activities and practices needed to change a development approach from traditional to Agile. The scope of these activities is not limited to technical practices and covers social behaviors and human aspects too [3]. This is the real reason why ATP needs a lot of effort and usually takes a long time [7].

Developers, top and middle managers, and customers should be involved in the transformation process. Moving to Agile makes a huge change to all aspects of a company, and due to the wide extent of change, companies are usually faced with a wide variety of barriers, problems, and challenges [4,8]. Process, people, and management-related issues, as well as cultural and technical problems are the main categories of transformation hindrances [4,7,9–11]. Reviewing previous studies shows that training plays a critical role in ATP and inadequate and dysfunctional training makes it harder and ineffective [6,12–14]. To ensure a comprehensive ATP, some important questions should be answered. Are all practitioners of ATP involved in the training courses for the transformation process? What and how should be taught in training packages? What are the real causes of inadequate and ineffective training even with a disciplined training program? What are the real outcomes and consequences of inadequate and dysfunctional training on the transformation process? And ultimately, what should the training process be?

Answering each question needs substantive data and appropriate research methodology. Grounded Theory (GT), as a qualitative research method, is helpful for such a study. By conducting a GT study, answers to the above questions can be found. This GT study involved 35 Agile practitioners from 13 different countries. All of them had significant experiences in ATP and voluntarily participated in this study. Having significant experience in Agile software development and participating in at least one ATP were compulsory requirements for participating in this study.

In this study, several semi-structured and online interviews with the aforementioned experts were conducted using open-ended questions, however, since all of them were from different countries, face-to-face interview was not possible. Supplementary data and information were also collected via memoing and two levels of literature review as GT addressed. However, supplementary data were used only for discovering conceptual connections between the emerged concepts. Following GT flow and data analysis, it emerged that *inadequate and dysfunctional training* is a significant issue in ATP.

Attendees identified several reasons for inadequate and dysfunctional training, such as partial training, inappropriate content, non-practical, and time-boxed training. At the same time, inadequate and dysfunctional training creates several problems and barriers for the transformation process, such as unrealistic expectations, difficulty of change in the organization, and a lack of deep understanding of Agile values. They also addressed some hidden and informal strategies that are used as temporary solutions for dealing with inadequate training that are called "*heuristic and ad-hoc treatments*".

The contributions of this paper are: (a) a description of causes of inadequate and dysfunctional training on ATP; (b) its adverse consequences on ATP; (c) ad-hoc strategies used by software teams to pursue ATP despite inadequate and dysfunctional training; (d) using the six C's coding model to describe the causes, consequences, and employed strategies, and finally, (e) a description of theoretical and practical implications of the results.

The rest of this paper is organized as follows: Section 2 provides a short background on Agile methods and ATP challenges and issues. Section 3 explains the research methodology. Section 4 presents the results of the study, followed by Section 5 that explains the evaluation of the findings. Section 6 presents a detailed discussion on the findings of the study, while the limitations of the study are presented in Section 7. Finally, Section 8 concludes the paper and suggests some future research directions.

2. Background

After using traditional software methodologies for nearly four decades, Agile methods, by focusing on the challenges of those methods, have provided new values such as light-weight documentation, fast delivery, customer satisfaction and high quality [15]. Several Agile methods have been gathered under the Agile umbrella when Agile manifesto was created and Agile software development was officially introduced [2]. Scrum [16], Extreme Programming (XP) [17], Crystal methods [18], Feature driven development (FDD) [18], Dynamic systems development method (DSDM) [18], Test Driven Development (TDD) [19], and lean software development [20] are examples of the well-known Agile methods, as addressed in the literature [21,22]. It is important to know that these methods are different in some activities and goals, but all of them have focused on the same values. Among them, some methods focus on project management (like Scrum and DSDM), while some others focus on software development (like XP and Crystal).

Previously, training was reported as a critical factor for successful process improvement [23] and it was stressed that without appropriate and accurate training, and understanding of necessary details, process improvement is impossible [23]. The importance of training was also emphasized in many studies as a critical factor for adapting new methodologies [12,24]. Also, significant correlation between successful implementation of Agile methods and receiving training has been proven [25]. While comprehensive and effective training is difficult and expensive, it helps organizations to overcome most of the challenges they are faced with during ATP [6,26]. In many cases, lack of suitable training is addressed as a significant barrier in Agile transformation and adoption [9,26,27].

The important point is that involvement and collaboration of practitioners and stakeholders (employees, managers, customers, etc.) is a necessary requirement for Agile software development [16,22,28]. Therefore, they should be involved in ATP, and furthermore, training packages should cover their training needs. Designing the content of a training package and its details varies between organizations due to the different combination of experts and different levels of their knowledge. In many companies before moving to Agile, team members attend some training courses [26,29]. However, most of the training courses only target developers and exclude other practitioners. Managers need their own specific training, since there can be a lot of management problems in ATP [4]. Insufficient or a lack of management buy-in is often due to a low understanding of Agile values and this can be solved by training [9]. The other major participants in ATP are the customers [9,27,30,31]. Customers play a significant role in Agile development and the lack of their involvement can cause transition to fail in some Agile methods. Lack of customer involvement and collaboration, which is a barrier in many cases, can also be solved by appropriate and efficient training and motivation [26,30].

Literature review, proper time, quality, and quantity have been topics of debate in GT [32,33]. Although GT does not involve formulating the research problem up front based on major literature review, application of literature review is not forbidden. However, Glaser [34] strictly warned GT researchers about adverse effects of conducting a major literature review in the same area of research during the early stages of GT. Following Glaser's recommendation, this research was started with a minor literature review in the