The Journal of Systems and Software 110 (2015) 85–100

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Agile methods tailoring – A systematic literature review

Amadeu Silveira Campanelli*, Fernando Silva Parreiras

LAIS Laboratory of Advanced Information Systems, FUMEC University, Av. Afonso Pena 3880 Belo Horizonte 30130009 MG, Brazil

ARTICLE INFO

Article history:
Received 25 November 2014
Revised 19 August 2015
Accepted 20 August 2015
Available online 28 August 2015

Keywords:
Agile method tailoring
Agile practice selection
Software method tailoring

ABSTRACT

Background: The software development industry has been adopting agile methods instead of traditional software development methods because they are more flexible and can bring benefits such as handling requirements changes, productivity gains and business alignment. Objective: This study seeks to evaluate, synthesize, and present aspects of research on agile methods tailoring including the method tailoring approaches adopted and the criteria used for agile practice selection. Method: The method adopted was a Systematic Literature Review (SLR) on studies published from 2002 to 2014. Results: 56 out of 783 papers have been identified as describing agile method tailoring approaches. These studies have been identified as case studies regarding the empirical research, as solution proposals regarding the research type, and as evaluation studies regarding the research validation type. Most of the papers used method engineering to implement tailoring and were not specific to any agile method on their scope. Conclusion: Most of agile methods tailoring research papers proposed or improved a technique, were implemented as case studies analyzing one case in details and validated their findings using evaluation. Method engineering was the base for tailoring, the approaches are independent of agile method and the main criteria used are internal environment and objectives variables.

1. Introduction

Agile methods for software development have been increasingly used by the software development industry (Deemer et al., 2010; Nishijima and Dos Santos, 2013; Qumer and Henderson-Sellers, 2006). Compared to traditional software development methods it presents advantages such as accelerate time to market, increase in quality and productivity, improve Information Technology (IT)/business alignment, and enhanced flexibility (Deemer et al., 2010; Jyothi and Rao, 2011; Nishijima and Dos Santos, 2013; Qumer and Henderson-Sellers, 2006; VersionOne, 2013). Even though the benefits of agile methods are worthwhile and have been proved by scientific and market researches (Ahmed and Sidky, 2009; de Azevedo Santos et al., 2011; Glaiel et al., 2013; Jyothi and Rao, 2011; Moniruzzaman and Hossain, 2013; VersionOne, 2013) the complexity of adopting them is high because of organization culture, resistance to change and need for upper management sponsorship and involvement (Chow and Cao, 2008; Dyba and Dingsoyr, 2009; Gandomani et al., 2013; Koch, 2005).

Based on the current challenges of software development, agile methods are interesting and viable options to achieve quality, project budget control, align with organization's business strategy and deliver value frequently and continuously (de Azevedo Santos et al., 2011; Glaiel et al., 2013; Nerur et al., 2005). The strategy choice for agile methods adoption is a key component to get the organization to take advantage of the benefits brought by agility and to overcome the common issues found on the adoption process (Nerur et al., 2005; Soundararajan et al., 2013). The agile adoption process is complex and requires lots of effort from the organization and teams, cultural adaptation, deals with egos and resistance to change, and demands upper management sponsorship (Abbas et al., 2010; Boehm and Turner, 2005; Chow and Cao, 2008; Dyba and Dingsoyr, 2009; Gandomani et al., 2013; VersionOne, 2013). Besides that, there are cases where practices available on the mainstream agile methods don't make sense for an organization (Kurapati et al., 2012). Adopting the practices proposed by an agile method at once means that the organization should need to spend effort and resources adopting it with no evaluation of the value each proposed agile practice (such as pair programming, continuous integration, unit testing and refactoring) brings to the organization to achieve its business goals (Abbas et al., 2010; Ahmed and Sidky, 2009; Kurapati et al., 2012). Analysis of previous research shows that a comprehensive SLR covering agile methods tailoring doesn't exist yet. This study aims to systematically review the existing literature on agile methods tailoring, summarize

* Corresponding author. Tel.: +55 31 3269 5230.
E-mail address: amadeu.campanelli@gmail.com, amadeu@fumec.edu.br (A.S. Campanelli).

http://dx.doi.org/10.1016/j.jss.2015.08.035
0164-1212/

the research and technical aspects, and identify trends and gaps in the existing research.

These types of challenges motivated academic researches on agile practice adoption and practices selection in order to generate a tailored software development method consistent with the organization's values, culture, reality, needs and strategies (Abbas et al., 2010; Ahmed and Sidky, 2009; Esfahani et al., 2011; Kurapati et al., 2012; Madi et al., 2011; Qumer and Henderson-Sellers, 2008). This way the organization will only spend effort adopting the practices that aggregate value to its strategy and help to achieve its business goals (Abbas et al., 2010; Ahmed and Sidky, 2009; Esfahani et al., 2011).

The problem of tailoring agile methods to be adopted by organizations is a known problem and researchers have been working on it but there is still no definitive answer for the theme (Abbas et al., 2010; Ahmed and Sidky, 2009; Esfahani et al., 2011; Kurapati et al., 2012; Madi et al., 2011). A secondary study such as this can help to identify problematic situations and potential solutions on the research theme. The SLR was chosen as the research method to detect trends in agile methods tailoring research and identify gaps and opportunities for further investigation (Kitchenham et al., 2009).

The objectives of this study are to summarize the existing literature on agile methods tailoring, to provide understanding how the research on agile methods tailoring is conduced (how empirical is the research, which approaches are being explored, what types of results have been generated), to identify the research community's view of agile method tailoring (which agile methods have been studied and how these approaches propose the agile practice selection), and to identify research gaps on the theme. These objectives could help organizations to understand how to fit agile practices on their context, adopt new practices with minimal disruption of the existing environment and start getting some of the benefits of agile methods.

The remainder of the paper is organized as follows. Section 2 provides the technical background and related work while Section 3 discusses the research methodology and data extraction procedures. The results of the study are presented on Section 4 and findings are discussed on Section 5. The conclusions and future work are presented on Section 6.

2. Background and related work

This section reviews two key concepts involved on this research: agile software development and software methods tailoring.

2.1. Agile software development

The Agile Manifesto (Beck et al., 2001) has consolidated values and principles from existing agile methods and approaches, and made the agile movement stronger and organized on the software development industry.

The agile values are:

- Individuals and interactions
- Working software
- Customer collaboration
- Responding to change

The agile principles are:

- Early and continuous delivery of valuable software
- Change is welcome
- Deliver frequently
- People interaction (business and developers)
- Motivated people
- Face-to-face communication
- Working software is progress
- Constant pace
- Technical excellence and good design
- Simplicity
- Self-organized teams
- Continuous improvement

Table 1
15 most adopted agile practices according to Version One State of Agile 2013 Research (VersionOne, 2013).

Position	Practice	Adoption percentage (%)
1	Daily stand-up	85
2	Iteration planning	75
3	Retrospectives	74
4	Unit testing	72
5	Release planning	70
6	Burndown/Team-based estimation	69
7	Velocity	60
8	Continuous integration	58
9	Automated builds	56
10	Coding standards	55
11	Dedicated product owner	55
12	Integrated Dev/QA	50
13	Refactoring	47
14	Digital task board	45
15	Open work area	44

Agile methods are in their essence based on values and principles defined on the Agile Manifesto (Beck et al., 2001) and composed by agile practices (Jalali and Wohlin, 2010). Agile practices should help accomplish agile principles in a method and can be grouped into management practices, software process practices and software development practices (Lee and Yong, 2013). Example of management practices are: on-site customer, daily stand-up meetings and open work area. Software process practices include simple design and collective code ownership. Pair programming and unit testing are examples of software development practices. Table 1 shows a list of most adopted practices in 2013 according to Version One State of Agile 2013 Research (VersionOne, 2013).

Handling unstable requirements, delivering working software in short time frames, with high quality and under budget are the main characteristics of agile methods compared to traditional ones (Jyothi and Rao, 2011). Being agile is to be able to rapidly adapt to change in a flexible way (Qumer and Henderson-Sellers, 2006). The capability is reflected by the attributes of flexibility, velocity, leanness, learning and response to change. According to Qumer and Henderson-Sellers (2006) agility can be defined as the ability to accommodate changes (expected or not) in a dynamic environment being simple, economic and having quality in a short iteration strategy applying previous knowledge and generating new ones on this experience. Handling unstable requirements, delivering working software in short time frames, with high quality and under budget are the main characteristics of agile methods compared to traditional ones (Jyothi and Rao, 2011).

2.2. Agile methods

The current mainstream agile methods are: Extreme Programming (XP), Scrum (Deemer et al., 2010; Schwaber and Sutherland, 2011), Kanban, Lean, Feature-Driven Development (FDD), Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Crystal and Rational Unified Process (RUP). Scrum is currently the most adopted one (VersionOne, 2013). Each method focuses on specific values and there is no standard on how a method should implement its agile features.

XP has been created to help small teams to develop software when requirements are vague and change frequently (Beck, 2000). It is considered a lightweight methodology and focuses on cost savings, unit tests before and during code activities, frequent full system integration, pair programming, simple design and frequent releases of working software (Beck, 2000).