



Review

A systematic literature review on agile requirements engineering practices and challenges



Irum Inayat^{a,*}, Siti Salwah Salim^a, Sabrina Marczak^b, Maya Daneva^c, Shahaboddin Shamshirband^{d,e}

^a Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya (UM), 50603, Malaysia

^b School of Computer Science, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) University, Rio de Janeiro, Brazil

^c Information Science Research Group, University of Twente, Enschede, The Netherlands

^d Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya (UM), 50603, Malaysia

^e Department of Computer Science, Chalous Branch, Islamic Azad University (IAU), 46615-397 Chalous, Mazandaran, Iran

ARTICLE INFO

Article history:

Available online 2 December 2014

Keywords:

Agile software development methods

Agile requirements engineering

Collaboration

Traditional requirements engineering

Systematic review

ABSTRACT

Unlike traditional software development methods, agile methods are marked by extensive collaboration, i.e. face-to-face communication. Although claimed to be beneficial, the software development community as a whole is still unfamiliar with the role of the requirements engineering practices in agile methods. The term “agile requirements engineering” is used to define the “agile way” of planning, executing and reasoning about requirements engineering activities. Moreover, not much is known about the challenges posed by collaboration-oriented agile way of dealing with requirements engineering activities. Our goal is to map the evidence available about requirements engineering practices adopted and challenges faced by agile teams in order to understand how traditional requirements engineering issues are resolved using agile requirements engineering. We conducted a systematic review of literature published between 2002 and June 2013 and identified 21 papers, that discuss agile requirements engineering. We formulated and applied specific inclusion and exclusion criteria in two distinct rounds to determine the most relevant studies for our research goal. The review identified 17 practices of agile requirements engineering, five challenges traceable to traditional requirements engineering that were overcome by agile requirements engineering, and eight challenges posed by the practice of agile requirements engineering. However, our findings suggest that agile requirements engineering as a research context needs additional attention and more empirical results are required to better understand the impact of agile requirements engineering practices e.g. dealing with non-functional requirements and self-organising teams.

© 2014 Elsevier Ltd. All rights reserved.

Contents

1. Introduction	916
2. Related work	916
3. Research method	917
3.1. Planning the review	917
3.1.1. Review objectives and research questions	917
3.1.2. Search strategy	918
3.1.3. Search criteria	918
3.1.4. Inclusion and exclusion criteria	918
3.2. Conducting the review	918
3.2.1. Study search and selection	918
3.2.2. Data extraction and synthesis	918
3.2.3. Methodological quality assessment	919
4. Findings of our review	919
4.1. Overview of studies	919
4.2. (RQ1) What are the adopted practices of agile RE according to published empirical studies?	920

* Corresponding author.

E-mail address: irum@siswa.um.edu.my (I. Inayat).

4.3. (RQ2) What are the challenges of traditional RE that are resolved by agile RE?	924
4.4. (RQ3) What are the practical challenges of agile RE?	924
5. Discussion of the results	925
6. Conclusions	926
6.1. Implications of the study	927
6.2. Limitations of our study	927
Acknowledgement	927
Appendix A. List of reviewed studies	928
References	928

1. Introduction

The Agile Manifesto states that priority should be given to “individuals and interaction over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to changes over following a plan” (Beck et al., 2001). These agile principles incorporate flexibility by cordially receiving changes to project scope and requirements definitions (Bang, 2007). Overall, a high-level project scope is defined upfront and is revisited in each iteration. Therein, requirements are initially defined with the customer and listed in a customer wish list format; every couple of weeks they are discussed (e.g. in the Scrum method), better understood, and reprioritised, to define the scope of the next iteration. The customer works closely with the development team to achieve such definitions and to constantly validate the product being delivered. The development process is dynamic and open to changes in areas that can be identified at any given moment. Literature reports those projects that adopt agile methods exhibiting higher productivity (Eberlein & Julio Cesar, 2002), less rework (Bin, Xiaohu, Zhijun, & Maddineni, 2004), and more efficient defect fixing rates (Lagerberg & Skude, 2013). In addition, agile methods reduce risks in global software development (GSD) and diminish the need for coordination efforts, which result in an increase of productivity (Hossain, Babar, & Verner, 2009).

Requirements Engineering (RE) practices such as observations, interviews, workshops and strong team collaboration are embedded in iteration-based agile methods (Zhu, 2009). Likewise, RE practices such as customer involvement, requirements prioritisation (Cao & Ramesh, 2008; Ramesh, Baskerville, & Cao, 2010), requirements modelling (Boness & Harrison, 2007), requirements documentation (Wolfgang, 2011), have also been suggested to be used with agile methods.

Although the practices mentioned above provide an essence of the “agile way” of dealing with requirements, the software development community still knows little about the role of the RE processes and practices in such a flexible and dynamic way of working, and how such practices can resolve frequently reported issues in traditional RE processes. Although claimed to be beneficial, the adoption of agile methods might impact the way that RE activities are conducted and pose some new challenges to their realisation. We are motivated to close this gap of knowledge and embarked on mapping out the published evidence available about RE practices adopted and challenges faced by agile teams. The purpose is to learn how traditional RE issues are resolved by this new software development approach.

The remainder of this paper is structured as follows: Section 2 discusses previous literature reviews on agile software engineering, identifies a gap in literature and a need for a deeper investigation of RE processes in agile software engineering. Section 3 presents our research questions and the method followed for the review of contemporary practices in agile RE. Section 4 summarises the key findings of our study. Section 5 provides a discussion on

the results. Section 6 concludes the article, provides implications for researchers and industry practitioners and defines the limitations of this study.

2. Related work

In the software engineering research literature, there are a few examples of reviews on agile methods, as (summarised in Table 1) usability issues (Hasnain, 2010) in agile methods and ways to resolve them (Silva da Silva, Martin, Maurer, & Silveira, 2011); agile methods in GSD (Hossain, Babar, & Paik, 2009; Jalali & Wohlin, 2011; Rizvi, 2013), and in open source software development (Gandomani, Zulzalil, Ghani, & Sultan, 2013).

Hossain et al. (2009) conducted a systematic literature review to focus on the practices used in the GSD projects using Scrum methods, the challenges that restrict the use of Scrum methodology and the solution to prevent them. The findings help researchers and practitioners to understand the challenges involved in using Scrum for GSD projects and the strategies available to deal with them.

Hasnain (2010) conducted a systematic literature review to identify the agile practices as well as the human and technical factors pointed out in agile studies, published within 2003–2007. The review revealed that agile RE practices had only been discussed in the literature from the overall perspective of agile methods and not in the context of any particular methods such as Scrum, test-driven development, etc. Hasnain’s findings suggest that more empirical results are required on agile methods, in particular XP (Extreme Programming) (Beck, 1999) and Scrum (Schwaber & Beedle, 2001), in order to discuss the details from the practitioner’s point of view.

Silva da Silva et al. (2011) conducted a systematic literature review on the topic of the integration of agile methods and user-centred design approaches. The review focused on usability issues in agile methods with respect to design. The findings show that usability issues in agile methods can be addressed by incorporating a user centred design specialist (UCDS) role in agile teams. The authors also defined practices to resolve usability issues in agile methods such as Little Design Up Front, Big Design Up Front, low fidelity prototypes, user testing, interaction models, and close collaboration.

Barlow et al. (2011) examined the effect of the usage of agile development practices in large organisations. The literature review contributed towards the formulation of a framework that provides guidelines to large organisations adopting agile methods. The findings of this review assist the practitioners to adopt software development methods in their organisations.

Jalali and Wohlin (2011) conducted a systematic literature review on studies comprising the combination of agile methods with global software engineering from 1999 to 2009. The review results showed that much attention had been given to agile methods from 2004 to 2009. In addition, the findings revealed that

Download English Version:

<https://daneshyari.com/en/article/10312634>

Download Persian Version:

<https://daneshyari.com/article/10312634>

[Daneshyari.com](https://daneshyari.com)