ScienceDirect

Computers in Human Behavior

Volume 51, Part B, October 2015, Pages 915-929

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]

Show more

Share Cite

https://doi.org/10.1016/j.chb.2014.10.046

Get rights and content

Highlights

- We mapped out 17 requirements engineering practices adopted by agile practitioners so far.
- Identified 5 challenges of traditional requirements engineering overcome by adopting agile requirements engineering.
- Found 8 challenges posed by following agile requirements engineering.
- Findings suggest that research context needs attention in terms of more empirical.
- The empirical results can help to analyse impact of adopting agile requirements engineering.

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.

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 and Ramesh, 2008, Ramesh et al., 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.

Section snippets

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 et al., 2009, Jalali and 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 …

Research method

In our research process, we followed the guidelines proposed by Kitchenham and Charters (2007). Subsequently, we presented the main steps of our systematic review, namely planning, conducting and reporting the review results (Kitchenham & Charters, 2007). …

Findings of our review

In this section, we describe the findings of our review in light of our research questions. …

Discussion of the results

An important aspect highlighted in the analysis of our 21 selected studies is the geographic locations of authors. It is observed that nearly 1/3 of all contributions were from North American countries (authors based in the US and Canada). This is unsurprising, considering the fact that the Agile Manifesto was created by North-American software development practitioners. European countries took second place in the contributions, involving the UK, Italy, the Netherlands, Belgium, Germany, …

Conclusions

This paper presents a systematic review of literature on practices and challenges of agile RE. This review was conducted by following available guidelines for conducting systematic literature reviews (Kitchenham & Charters, 2007) to search and categorise all existing and available literature on agile RE. Of the 543 initial papers located in well-known electronic research databases, 21 relevant papers were extracted through a multistage sifting process with independent validation in each step. …

Acknowledgement

This research is supported by UM High Impact Research Grant UM.C/625/1/HIR/MOE/FCSIT/05 from the Ministry of Higher Education, Malaysia. We are also thankful to the Brightsparks Unit, University of Malaya for their support in this research. …

Special issue articles Recommended articles

References (69)

M. Daneva *et al.*
Agile requirements prioritization in large-scale outsourced system projects: An empirical study
Journal of Systems and Software (2013)

T. Dybå *et al.*
Empirical studies of agile software development: A systematic review
Information and Software Technology (2008)

R. Hoda *et al.*
The impact of inadequate customer collaboration on self-organizing agile teams
Information and Software Technology (2011)

C. Pacheco *et al.*
A systematic literature review of stakeholder identification methods in requirements elicitation
Journal of Systems and Software (2012)

R. Wieringa
Empirical research methods for technology validation: Scaling up to practice
Journal of Systems and Software (2014)

Abdullah, N. N. B., Honiden, S., Sharp, H., Nuseibeh, B., & Notkin, D. (2011). Communication patterns of agile...

Al-Ani, B., Bietz, M. J., Wang, Y., Koehne, B., Marczak, S., Redmiles, D., et al. (2013). Globally distributed system...

Ambler, S. W. (2008a). Beyond functional requirements on agile projects. Dr. Dobb's....

Ambler, S. W. (2008b). Beyond functional requirements on agile projects. Dr....

T.J. Bang
An agile approach to requirement specification

View more references

Cited by (400)

Requirements engineering challenges and practices in large-scale agile system development

2021, Journal of Systems and Software

Show abstract

Requirements engineering: A systematic mapping study in agile software development

2018, Journal of Systems and Software

Show abstract

Agile Requirements Engineering: A systematic literature review

2017, Computer Standards and Interfaces

Show abstract

A systematic literature review on Internet of things in education: Benefits and challenges

2020, Journal of Computer Assisted Learning

Towards Sustainable Energy: A Systematic Review of Renewable Energy Sources, Technologies, and Public Opinions

2019, IEEE Access

On the pragmatic design of literature studies in software engineering: an experience-based guideline

2017, Empirical Software Engineering

View all citing articles on Scopus

View full text