

# Agile Requirements Engineering: A systematic literature review



Eva-Maria Schön<sup>a,\*</sup>, Jörg Thomaschewski<sup>b</sup>, María José Escalona<sup>a</sup>

<sup>a</sup> University of Seville, Spain

<sup>b</sup> University of Applied Sciences Emden/Leer, Emden, Germany

## ARTICLE INFO

### Article history:

Received 11 March 2016

Received in revised form

25 August 2016

Accepted 30 August 2016

Available online 2 September 2016

### Keywords:

Systematic literature review

Agile software development

Requirements Engineering

Human-computer interaction

User-centered design

## ABSTRACT

Nowadays, *Agile Software Development* (ASD) is used to cope with increasing complexity in system development. Hybrid development models, with the integration of *User-Centered Design* (UCD), are applied with the aim to deliver competitive products with a suitable *User Experience* (UX). Therefore, stakeholder and user involvement during *Requirements Engineering* (RE) are essential in order to establish a collaborative environment with constant feedback loops. The aim of this study is to capture the current state of the art of the literature related to Agile RE with focus on stakeholder and user involvement. In particular, we investigate what approaches exist to involve stakeholder in the process, which methodologies are commonly used to present the user perspective and how requirements management is been carried out.

We conduct a Systematic Literature Review (SLR) with an extensive quality assessment of the included studies. We identified 27 relevant papers. After analyzing them in detail, we derive deep insights to the following aspects of Agile RE: stakeholder and user involvement, data gathering, user perspective, integrated methodologies, shared understanding, artifacts, documentation and Non-Functional Requirements (NFR). Agile RE is a complex research field with cross-functional influences. This study will contribute to the software development body of knowledge by assessing the involvement of stakeholder and user in Agile RE, providing methodologies that make ASD more human-centric and giving an overview of requirements management in ASD.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

Nowadays the business world is characterized by complexity, since market requirements are changing quickly. Accordingly, providers are facing the challenge to reduce time to market while delivering innovative products that customer love. Agile software development (ASD) promises benefits such as on-time delivery and customer satisfaction [1], thus it aims to deliver business value in short iterations. Therefore, the development process is carried out incrementally and empirically, which is an advantage because direction of product development can be changed immediately. Humans and interactions are at the center of such methodologies [2].

Agile methodologies (e.g. Scrum [3], Kanban [4] or Extreme Programming [5]) provide a process model to develop products. These models lack in defining the right kind of product, which fulfils user needs and customer expectations. In order to fill in this gap and to develop products with a good *user experience* (UX), hybrid

development approaches including *Human-Centered Design* ([6] referred to as *User-Centered Design*, UCD) are applied. Although there are some challenges reported while integrating ASD and UCD (see 2.1), the integration of both makes development process more human-centered [7]. Stakeholder and user involvement is a critical success factor for a system to succeed [8] and, if compared with traditional approaches, this involvement is not limited to early phases of development, as stakeholder and user are involved throughout the whole development process instead [9].

Requirements are the base of all software products and consequently Requirements Engineering (RE) plays an important role in system development. Compared to traditional RE approaches ([10], [11]), a list of prioritized requirements (Product Backlog [3]) is used instead of a requirements specification document. The main RE activities (*elicitation, documentation, validation, negotiation and management*) are not clearly separated activities in *Agile RE*. They are repeated each iteration and only required information is elaborated before the next iteration starts. For this purpose, RE in Agile environments is carried out just-in-time with a Little Design Up Front [12].

This article reports the findings of a Systematic Literature Review (SLR) in the field of Agile RE with focus on stakeholder and

\* Corresponding author.

E-mail addresses: [eva.schoen@iwt2.org](mailto:eva.schoen@iwt2.org) (E.-M. Schön), [Joerg.Thomaschewski@hs-emden-leer.de](mailto:Joerg.Thomaschewski@hs-emden-leer.de) (J. Thomaschewski), [mjescalona@us.es](mailto:mjescalona@us.es) (M.J. Escalona).

user involvement. In particular, ASD, RE and UCD have one thing in common: stakeholder and user involvement is described as critical success factor for a system to succeed. To this end, this will be an important aspect in this literature review and will be addressed by the following research questions:

- RQ1: What approaches exist, which involve stakeholders in the process of RE and are compatible with ASD?
- RQ2: Which agile methodologies, which are capable of presenting the user perspective to stakeholders, can be found? In terms of RE, these research questions lead us to the third research question:
- RQ3: What are the common ways for requirements management in ASD?

The paper is structured as follows: [Section 2](#) gives a brief overview of Agile RE context, including a gap analysis of related work. [Section 3](#) presents our research objectives and research questions and deals with our review method covering a description of the search strategy, selection process, quality assessment, data extraction and analysis. [Section 4](#) summarizes the key findings of our study, therefore it offers an overview of the included studies as well as answers to our three research questions. Finally, [Section 5](#) discusses on the meaning of findings and limitations of this study.

## 2. Background – Agile Requirements Engineering

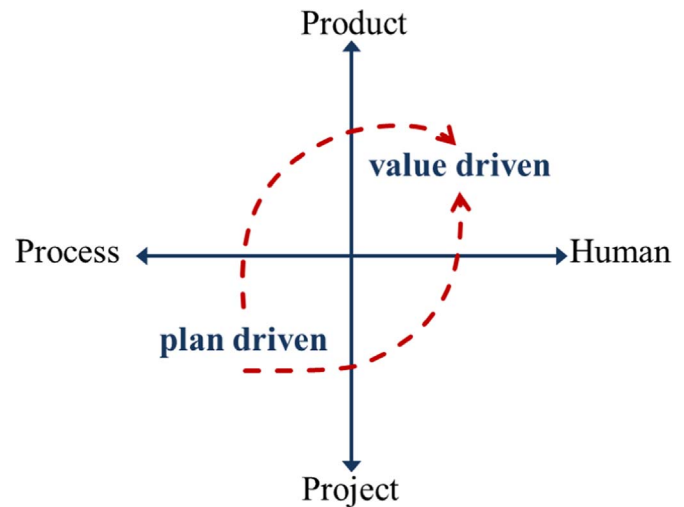
In the mid-80s, Takeuchi et al. [13] already stated that a sequential phases approach to product development is not well suited due to the lack of flexibility. Since then, new process models have been developed. On one hand, there are iterative process models like *Rational Unified Process* [14,15]. On the other hand, there are Agile methodologies such as Scrum [16,3], Extreme Programming (XP) [5], Feature-Driven Development [17] and Kanban. The usage of Kanban for Information Technology (IT) was mainly influenced by Anderson in between 2004–2010 [18,4].

A number of papers regarding lightweight process models have been published. In 2001, the leaders of these different streams joined together and created the Manifesto for Agile Software Development [2]. The Agile Manifesto includes values and principles that help to optimize the software development process and have also a strong influence on nowadays team collaboration within ASD [19]. The Agile Manifesto provides the four core values listed below:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Since the Agile movement, software development has moved away from plan-driven to value-driven process models (see [Fig. 1](#)). People in plan-driven environments often negotiate about pricing models, project plans and how many product features they can develop with the available resources. They are emphasizing the generated outputs (e.g. number of created features or number of releases during a time period). In contrast, people in value-driven environments discuss visions, experiences and human values as well as how they can address them through the product. They concentrate on outcomes, which means that they are focused on the difference that outputs entail. Therefore, product development with agile methodologies is mainly driven by human values.

In the context of agile methodologies, RE is carried out iteratively during the whole development process instead of during a



**Fig. 1.** In ASD people focus on outcomes and how they can fulfil human needs through the outputs they produce.

closed phase in the beginning. To this end, a just-in-time model is often used to refine high level requirements into low level tasks that can be implemented by developers. Therefore, business people, stakeholders, users and developers work together in a collaborative manner. The model is artifact-based and starts with capturing requirements by means of epics. An epic is a large user story [20], that can be refined by utilizing story maps [21]. A story map consists of user stories or persona stories [22], which are split into tasks. The whole workflow can be managed by means of Kanban boards for design, development and delivery [23].

Agile techniques like Continuous Delivery (CD) [24] have an impact on the manner and the frequency of usability testing nowadays [25]. New information is given along the system development by the user and the system itself. This knowledge is processed during further steps and it conditions the decision-making process. Therefore, requirements are treated like assumptions, which are validated continuously.

There are also initiatives focusing on the alignment of RE and testing, which investigate the practice of using test cases as requirements [26]. In this context, detailed requirements are often documented as test cases, rather than using additional requirements specification, in order to reduce the effort of updating separate artifacts [26].

### 2.1. Summary of related literature reviews

In the literature, many reviews are conducted in order to do research on ASD. The next paragraphs briefly summarize the most related ones.

In 2011, Silva et al. [27] carried out a SLR on the integration of ASD and UCD and analyzed how usability issues are addressed in agile projects. They included a comprehensive classification process based on a system covering research-related and content-related information. The authors identified the following key aspects, which play an important role for the integration: little up-front design, prototyping, user stories, user testing, inspection evaluation and one sprint ahead. Besides, they presented a process model for the integration of ASD and UCD that took into account their findings.

Salah et al. [28] addressed a similar area. Their review aimed to identify challenging factors for the integration of ASD and UCD. They presented the challenges in a very understandable manner with good examples. Additionally, they explored practices and success factors to face these challenges. The reported challenges

Download English Version:

<https://daneshyari.com/en/article/6883192>

Download Persian Version:

<https://daneshyari.com/article/6883192>

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