



# Requirements engineering: A systematic mapping study in agile software development

Karina Curcio, Tiago Navarro, Andreia Malucelli, Sheila Reinehr\*

Graduate Program in Computer Science (PPGla), Pontifícia Universidade Católica do Paraná – PUCPR, Curitiba, Brazil



## ARTICLE INFO

### Article history:

Received 19 December 2016  
 Revised 16 December 2017  
 Accepted 24 January 2018  
 Available online 31 January 2018

### Keywords:

Agile software development  
 Requirements engineering  
 Systematic mapping study

## ABSTRACT

**Context:** Requirements engineering in agile software development is a relatively recent software engineering topic and it is not completely explored and understood. The understanding of how this process works on agile world needs a deeper analysis.

**Objective:** The goal of this paper is to map the subject area of requirements engineering in agile context to identify the main topics that have been researched and to identify gaps to develop future researches. It is also intended to identify the obstacles that practitioners face when using agile requirements engineering.

**Method:** A systematic mapping study was conducted and as a result 2171 papers were initially identified and further narrowed to 104 by applying exclusion criteria and analysis.

**Conclusion:** After completing the classification and the analysis of the selected studies it was possible to identify 15 areas (13 based on SWEBOK) where researches were developed. Five of such areas points to the need of future researches, among them are requirements elicitation, change management, measuring requirements, software requirements tools and comparative studies between traditional and agile requirements. In this research, some obstacles that practitioners face dealing with requirements engineering in agile context were also identified. They are related to environment, people and resources.

© 2018 Elsevier Inc. All rights reserved.

## 1. Introduction

Since the agile manifesto was released a lot of research has been intensively developed to explore the agile software development (Dybå and Dingsøy, 2008; Diebold and Dahlem, 2014). The agile manifesto stated some values like “individual and interactions over process and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan” (Agile Manifesto: Manifesto for Agile Software Development 2001).

All of these values add flexibility and consequently agility to the software development process. However, recent studies indicate high rates of projects failure, including those that are using agile processes. According to a study published by the Standish Group (2014) about the results of software projects, five of the eight top projects cancellation factors are related to requirements. Incomplete requirements, low customer involvement, unrealistic expectations, changes in requirements and unnecessary requirements were listed as the main factors.

So, the question that remains is how to deal with it? The answer should be inside of requirements engineering area. In traditional requirements engineering some sequential activities are developed during five specific phases: elicitation, analysis, documentation, validation and verification phases (Kotonya and Somerville, 1997). During all these phases it is necessary to manage the requirements. What is already known is that in the traditional development life cycle, all of these activities are done during the analysis phase of the software development life cycle. It is easy to imagine this occurring when a plan-driven approach is adopted, but when an agile approach is adopted all activities and phases described on traditional requirements engineering are not so clear. Agile requirements engineering (RE) activities are not sequential but are iterative and are performed during each of the several short development cycles [S1]. Requirements engineering in agile development is informal and based on the skills and knowledge of individuals (Dingsøy et al., 2012). It is difficult to describe or characterize the agile requirements engineering because it is still cloudy, not only for software developers but for the research community too. The software development community as a whole is still unfamiliar with the role of the requirements engineering practices in agile methods (Inayat et al., 2015).

\* Corresponding author.

E-mail addresses: [malu@ppgia.pucpr.br](mailto:malu@ppgia.pucpr.br) (A. Malucelli), [sheila.reinehr@pucpr.br](mailto:sheila.reinehr@pucpr.br) (S. Reinehr).

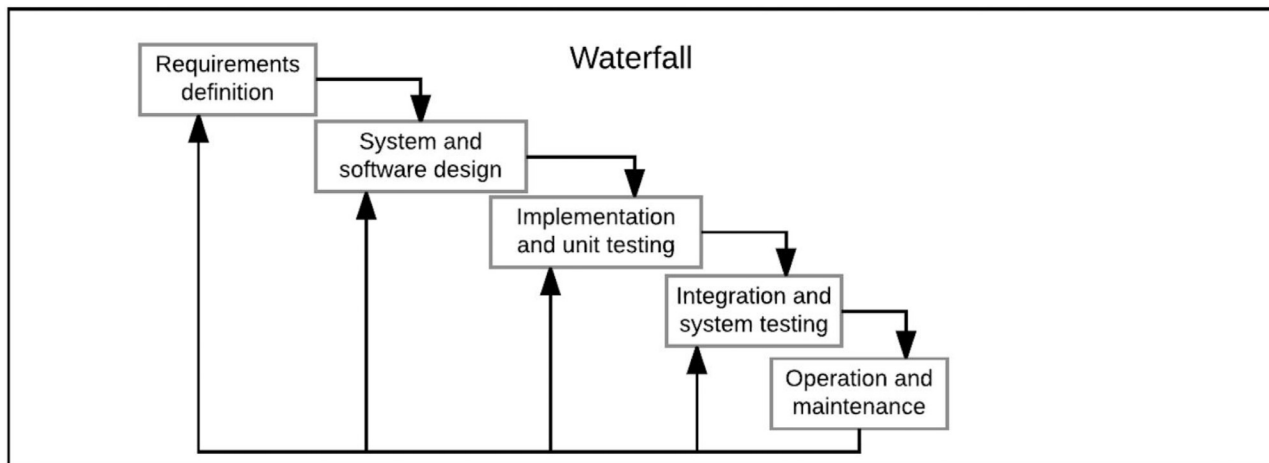


Fig. 1. Waterfall model based on Sommerville (2001).

Motivated by this perception, the goal of this study is to explore the agile requirements engineering and bring to light more discussions promoted by literature foundations. To attend this goal it is intended to create a clear and objective way of visualizing the results instigating those who intend to follow this line of research. At the beginning of the research it was noticed that the field of this research is very broad and we need to find a way to promote adequate visibility of the results, focusing on the extensions of the areas that the agile requirements engineering can achieve rather than its depth. As described by Kitchenham and Charters (2007) a systematic mapping study allows the evidence in a domain to be plotted at a high level of granularity. Systematic mapping studies or scoping studies are designed to give an overview of a research area through classification and counting contributions in relation to the categories of that classification (Petersen et al., 2008). It provides a structure of the type of research reports and results that have been published by categorizing them. It often gives a visual summary, the map, of its results (Petersen et al., 2008). To make the results emerge correctly and to reduce the bias of this research a systematic mapping methodology was applied. The research questions that we aimed to answer are the following:

- 1) On which requirements engineering topics are the researches on requirements engineering in agile software development concentrated?
- 2) What are the gaps concerning the requirements engineering in the context of agile software development?
- 3) What obstacles do the agile requirements engineering is facing (environment, people and resources)?

## 2. Theoretical background

Requirements engineering is concerned with identifying, modeling, communicating and documenting the requirements of a system and the context in which the system will be used [S2]. The use of the term “engineering” implies that systematic and repeatable techniques should be used to ensure that system requirements are complete, consistent and relevant (Kotonya and Sommerville, 1997). A requirement engineering process is a structural set of activities, which are followed to derive, validate, and maintain a system requirements document. In this section we will provide the theoretical background about requirements engineering in order to have a better understanding of the research context.

### 2.1. Traditional requirements engineering process

The term “traditional requirements engineering” is based on the waterfall life cycle model, which emerged in 1970s. In this approach all process for developing a system are executed in a sequential order, in which progress is seen as flowing steadily downwards through the phases of: requirements definition, system and software design, implementation and unit testing, integration and system testing and operation and maintenance (Sommerville, 2001). The representation of this model can be seen in Fig. 1.

In traditional requirements engineering a number of processes for gathering requirements in accordance with the needs and demands of the users are involved and all of them are executed during the requirements definition phase. The process begins with the elicitation activity where requirements and the system boundaries are discovered through the stakeholders. In this phase many techniques can be used, such as prototyping, brainstorming, interviews and use cases [S3]. Then the requirements analysis and negotiation activities start to get a better understanding of the whole business and to check if the elicited requirements are consistent, complete and feasible. Sometimes, during these activities, the requirements can be modeled to make them clearer for the developers. It is also possible to prioritize the requirements to satisfy some limitations such as time, resources or technical capabilities.

In the documentation activity the requirements are written and become a baseline for specifying all types of functional and non-functional requirements. The next activity in the sequence is the validation. The validation checks if the requirements statements are consistent and if they satisfy customer’s needs. Test cases are also used in this phase to reveal the ambiguities and vagueness in written requirements [S3]. To support these activities a requirements management process should be introduced to manage changes to the requirements during all prior presented phases.

In waterfall approach it is normal to freeze parts of the development such as the specification (Sommerville, 2001). Problems are left for later resolution, ignored or programmed around and it may bring some problems like badly structured systems.

### 2.2. Agile requirements engineering

Differently from the traditional requirements engineering the term “agile requirements engineering” is recent and emerged from the agile manifesto in 2001, as the agile software development started to be explored. During the agile manifesto twelve principles were stated and one of them was directly related to requirements:

Download English Version:

<https://daneshyari.com/en/article/6885329>

Download Persian Version:

<https://daneshyari.com/article/6885329>

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