A Systematic Literature Review on Requirements Quality in Agile Development

Anonymous[1], **Anonymous**[1]

[1]Anonymous Address

anonymous@email.com, anonymous@email.com

Abstract. *Intuition leads one to believe that a good software product is a product of good requirements. Industrial reports enhance this belief by showing that incomplete requirements are one of the top problems that challenge the success of software projects and drive them to be impaired and ultimately cancelled, causing waste of resources and financial investment. Agile development values individuals and interactions over process and tools, reducing the role of requirements documentation. However, the few existing documentation should still have an excellent quality in order to maintain the project's team knowledge sharing capability. This paper reports on a systematic literature review that aimed to identify what is considered a requirement with good quality in agile development, a first step towards improving requirements documentation and project outcomes in such setting.*

1. Introduction

The main goal of a software methodology is to develop software that matches the client's needs and expectations. In order to accomplish that goal, the development team should perform requirements engineering (RE) tasks that helps them to identify and structure the clients' requirements [Patel and Ramachandran 2010].

The importance of "good" written documentation for software projects is well-known in the industry. It has already been shown that incomplete requirements are one of many potential causes of a project failure [The standish group 2015]. Also, as requirements are used as inputs to estimate size of work to be done, the accuracy of estimates can be improved whenever they are well understood [Usman et al. 2015]. Furthermore, intuition make us believe that better estimation of effort reduces software cost and rework. Finally, the authors on [Paetsch et al. 2003] point out that the later mistakes are discovered the more expensive it will be to correct them. There are studies (e.g., [Langenfeld et al. 2016]) that validate this understanding and measure how different requirements defects cost throughout a project life time.

The Agile Manifesto [Kent Beck et al. 2001] values individuals and interactions much more than processes and tools. On those contexts, lengthy documentations are not as important as working software, implying that the time spent on requirements validation is diminished due to the focus on communication and collaboration. As seen on [Heikkila et al. 2015], agile RE is more flexible and reactive than a traditional, incremental approach. On [Inayat et al. 2015] the importance of user stories, the format used on those contexts as specifications to customer requirements, is described. One of the points of views raised on their work is that user stories shift the concentration from written documentation to communication.

However, [Paetsch et al. 2003] remind us that documentation is useful not only for describing the work to be done, but also for sharing knowledge between people. For example, a new team member will have many questions regarding the project that could be answered by other team members or by reading and understanding ”good” documentation. The later is the preferable method, as asking other team members will slow down work load due to the time it takes to explain a complex project to someone. Also, documentation reduces knowledge loss when team members become unavailable, move to another company, or are working on a new project. Even with all this usefulness of written documentation, RE and its related documentation is seen in agile methods as bureaucratic, which makes the process less agile [Medeiros et al. 2015].

We understand that the focus on reducing a project's documentations does not diminish the need of a documentation with a certain criteria of quality, that supports the team's business decisions effectively. Therefore, the current work aimed to conduct a systematic literature review to discover what agile studies consider to be a ”good” form of documentation and requirements on that context.

The remainder of this paper is organized as follows. Section II describes the concepts that are used as reference on the rest of this paper. Section III presents our research methodology and Section IV summarizes the results of our systematic literature review. Section V concludes the paper and highlights the limitations on our study and the implications for research.

2. Background

2.1. Requirements Quality

According to [IIBA 2015], a requirement is an usable representation of a client's need, focused on the value that will be delivered when it is implemented into the product.

The process with the goal to identify, analyze, document and validate requirements for the system to be developed is known as requirements engineering (RE) [Paetsch et al. 2003]. Paetsch (2003) also define that the purpose of requirements validation is to certify that the requirements are an acceptable description of the system to be implemented. Besides, [Heikkila et al. 2015] describe that requirements validation supports the three other activities by identifying and correcting errors in the requirements. Those errors are measured by different aspects, such as those found on [IIBA 2009] and [IIBA 2015].

As per [IIBA 2015], acceptable quality requirements exhibit many of the following characteristics: atomic, complete, consistent, concise, feasible, unambiguous, testable, prioritized, understandable. On [IIBA 2009], it is argued that a high quality requirement presents the following minimal characteristics: cohesion, completeness, consistency, correctness, viability, adaptability, non-ambiguity and testability.

The importance of ”good” requirements is highlighted on [IIBA 2015] when the authors say that a high-quality specification is well written and easily understood by its intended audience and that a high-quality model follows the formal or informal notation standards and effectively represents reality.

2.2. Requirements Engineering in Agile Development

Studies such as [Medeiros et al. 2015], [Paetsch et al. 2003], [Heikkila et al. 2015] and [Inayat et al. 2015] have shown that the requirements processes activities are executed in a continuous way, along with the product construction. Even with the lack of a formal and well accepted definition for agile RE, the authors on [Heikkila et al. 2015] propose the following one:

> "In agile RE, the requirements are elicited, analysed and specified in an ongoing and close collaboration with a customer or customer representative in order to achieve high reactivity to changes in the requirements and in the environment. Continuous requirements re-evaluation is vital for the success of the solution system, and the close collaboration with the customer or customer representative is the essential method of requirements and system validation."

[Paetsch et al. 2003] show that requirements validation on agile contexts focuses on frequent review meetings and acceptance tests. On the same paper, the lack of documentation is mentioned to potentially cause long-term problems for agile teams, such as improvement of knowledge loss when team members become unavailable and lack of training material to new members. We believe that the "good" quality of the minimal documentation generated by agile methodologies could mitigate those problems.

2.3. Related Work

Several systematic reviews or mapping of literature have been published on agile RE field. The most relevant ones for our study are summarized next.

[Medeiros et al. 2015] analyze requirements on agile development and highlight requirements documentations as a challenge on this setting. It also identifies that only a small portion of identified studies have reported the usage of practices to validate requirements, which might have caused the low count of problems on that area.

On [Inayat et al. 2015], minimal documentation appears as another challenge. It has a direct negative impact on problems related to requirements traceability on agile methodologies, which can cause difficult situations to distributed teams.

Yet more problems are reported on [Heikkila et al. 2015]. One of them is the insufficiency of the user story format, causing problems to represent and validate requirements documentation on that format. Another one is the teams' reliance on tacit requirements knowledge only, the kind of knowledge that appears from personal experience and is not taught nor written. This trust may cause problems due to personnel turnover.

In none of the work reported by the above mentioned studies the challenges and problems concerning requirements documentations quality were deeply analyzed. The current work aims to analyze further on this topic of interest to the agile and software communities aiming to be a first step towards improving requirements documentation and project outcomes–a long-term goal of our research project.

3. Research Method

According to [Kitchenham and Charters 2007], a systematic literature review is a means of identifying, evaluating, and interpreting all available research relevant to a particular

research question, topic area, or phenomenon of interest. The most common reasons for undertaking it are: to summarize the existing evidence concerning a treatment or technology; to identify any gaps in current research in order to suggest areas for further investigation; or to provide a framework/background in order to appropriately position new research activities.

The current work is motivated by the first item, as it seeks to summarize the existent knowledge on requirements quality on the context of projects that follow agile methodologies.

3.1. Research Questions

[Kitchenham and Charters 2007] highlight that the most critical issue in any systematic review is to ask the right question. In this current work, the right question seems to be pointing towards the goal of clarifying the current knowledge on requirements quality at agile development literature. With that goal in mind, the following research questions (RQ) were posed:

> **RQ1** What is the concept of requirements quality on agile development?
> **RQ2** What aspects are used to evaluate the requirements quality on agile development?

3.2. Search Strategy

By following the general approach seen on [Kitchenham and Charters 2007], we highlighted the following list of term groups: ”requirement”, ”quality”, and ”agile”. In order to break them down, we used the list of synonyms for ”requirement” and ”agile” found on [Medeiros et al. 2015] and [Inayat et al. 2015]. For the split of the ”quality” term, we used [Tiwari and Gupta 2015] and [ElAttar and Miller 2012] as reference. As we wanted to narrow our search on the RE studies and not on other quality related subjects (like product quality), those three groups of words were combined with an exclusive term: ”requirements engineering”. The final list of words per group follows below:

1. **Requirements group:** ”requirements”, ”use case”, ”use cases”, ”story”, ”user stories”, ”feature”, ”specifications”, ”formalism”, ”textual descriptions”, ”templates”, ”models”, ”documentation”;
2. **Agile group:** ”agile”, ”agility”, ”scrum”, ”XP”, ”extreme programming”, ”fdd”, ”feature-driven development”, ”featuredriven”, ”lean”, ”kanban”, ”behaviour-driven development”, ”tdd”, ”test-driven development”, ”test-driven”;
3. **Quality group:** ”quality”, ”validation”, ”criterias”, ”heuristics”, ”guidelines”, ”anti-patterns”, ”patterns”, ”mistake”, ”problem”, ”drawback”, ”recommendation”, ”suggestion”, ”warning”, ”rule”, ”syntax”, ”pitfalls”, ”classification”, ”assessment”, ”checklist”.

Note that words inside a group are combined with an *or* clause and the groups themselves are combined with the *and* clause between them and the term ”requirements engineering”.

3.3. Data Sources

In order to focus our effort on the results found, we have used the mentioned terms on a single automated search on a digital repository that concentrates the publications of several others (like ACM, IEEE and Springer). Scopus [1] claims to be the largest abstract and citation database of peer-reviewed literature and helped us to search on many data sources with a single search query. The search was made on the title, the abstract, and the keywords of the publications.

3.4. Study Inclusion Criteria

In order to provide an initial filter on the result obtained from the search query, the following inclusion criteria were adopted:

- Studies focused on describing requirements quality with qualitative or quantitative nature;
- Studies focused on evaluating requirements quality on agile development, despite if focusing on teams or projects;
- Studies focused on listing aspects of requirements quality on agile development.

3.5. Study exclusion criteria

A paper was excluded from our results if at least one of the exclusion criteria was complied, as follows:

- Studies not related with requirements engineering;
- Studies not even marginally related with requirements quality (that focus on product quality, for instance)
- Studies that do not define their context on agile development (projects or teams);
- Studies not written in English.

3.6. Results Selection

The results obtained from the search query execution on Scopus database were exported to the STaRT tool [Fabbri et al. 2012]. With the help of this tool, the list was then revised by the title and abstract reading. Those who passed our inclusion criteria and were not affected by our exclusion criteria were marked to be read in full.

The reading of the full papers was necessary to extract the information we wanted to obtain, be it the definition of requirements quality on agile development (to answer **RQ1**) or the aspects of requirements quality used on it (to answer **RQ2**). Only those studies who helped us answer one or another research questions were part of our final results as presented in details next.

4. Results

The automated search query have obtained 175 studies. From those, 61 were chosen to a full reading and 31 have answered one or another research questions. This whole process is summarized in Figure 1 and the 31 accepted papers are shown in Table 1.

[1] Scopus - https://www.scopus.com/

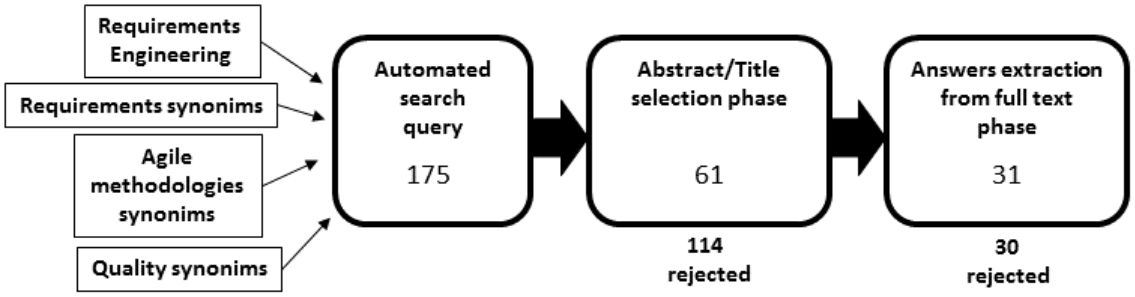

Figure 1. Summary of our search process

Paper ID	Title	Year
P2	The use and effectiveness of user stories in practice	2016
P3	Gamified requirements engineering: Model and experimentation	2016
P8	Preventing incomplete/hidden requirements: Reflections on survey data from Austria and Brazil	2016
P9	Quality criteria for just-in-time requirements: Just enough, just-in-time?	2015
P11	Forging high-quality User Stories: Towards a discipline for Agile Requirements	2015
P12	User scenarios through user interaction diagrams	2015
P13	A Mapping Study on Requirements Engineering in Agile Software Development	2015
P20	An impact study of business process models for requirements elicitation in XP	2015
P24	Integration of agile practices: An approach to improve the quality of software specifications	2015
P26	A process to increase the model quality in the context of model-based testing	2015
P28	Evaluation of BehaviorMap: A user-centered behavior language	2015
P30	Why the development outcome does not meet the product owners' expectations?	2015
P33	Requirements engineering in agile projects: A systematic mapping based in evidences of industry	2015
P34	Requirements communication and balancing in large-scale software-intensive product development	2015
P40	Assessing requirements engineering and software test alignment - Five case studies	2015
P43	Use of method for elicitation, documentation, and validation of software user requirements (MEDoV) in agile software development projects	2014
P45	Combining IID with BDD to enhance the critical quality of security functional requirements	2014
P47	Requirements engineering quality revealed through functional size measurement: An empirical study in an agile context	2014
P48	SnapMind: A framework to support consistency and validation of model-based requirements in agile development	2014
P73	Case studies in just-in-time requirements analysis	2012
P76	Cherishing ambiguity	2012
P81	Task descriptions versus use cases	2012
P108	Towards knowledge assisted agile requirements evolution	2010
P109	Requirements engineering in agile software development	2010
P114	Best practices guidelines for agile requirements engineering practices	2009
P121	Story card Maturity Model (SMM): A process improvement framework for agile requirements engineering practices	2009
P136	Agile methods and requirements engineering in change intensive projects	2008
P149	On Agile performance requirements specification and testing	2006
P159	Good quality requirements in unified process	2005
P160	Towards an aspect-oriented agile requirements approach	2005
P166	Generating complete, unambiguous, and verifiable requirements from stories, scenarios, and use cases	2004

Table 1. Accepted papers identification

4.1. Studies Overview

Given those studies that have answered our research questions, most of them (21 from 31) were published on conferences or workshops, while a few (8 from 31) were published on journals, and the rest of them (2 studies) were found on books chapters. Also, by looking at the publication year from the obtained results, one can see that many studies from the last 4 years were obtained.

When we look at what agile methodology each study follows, we can see that the most of them talk about agile in a broad and generic way, without mentioning any specific methodology. This approach is the same one when talking about format of requirements. Even with the popularity of the User Story format from many studies uses ”agile requirements” as the focus of their work. Many studies have tried to compare one or other methodology or map one requirement format on another, so the were counted twice (or more) on the detailed charts in Figure 2.

4.2. Concept of Requirements Quality on Agile Development

The answers to **RQ1** were related to the concept of requirements quality on agile development and can be found summarized in Figure 3. However, the majority of the studies (51%) did not answer this question directly, as they faced quality of requirements as a list of aspects. Agile RE documentation matching those aspects are considered to be good ones. In contrast, almost a third of the studies (29%) do not focus on the requirements documentations to decide what a good requirement is. Instead, they understand that good requirements is when customer's needs and expectations are met. A small parcel of studies (10%) understand that the definition of quality is a checklist that requirements shall attend to in order to be considered ready to be developed. Another small percentage of studies (10%) understand that the lack of problems on requirements is enough of a guarantee of quality.

4.3. Aspects Used to Evaluate Requirements Quality on Agile Development

The research question **RQ2** seeks to understand what aspects were used on agile methodologies to evaluate the quality of requirements. The aspects obtained on the accepted papers are shown on Table 2, along with the papers identification codes that referenced that characteristic. The last column, counting the number of papers referencing the rows aspect, shows us that the traditional aspects found on [IIBA 2015] and [IIBA 2009] are still relevant, as we see that characteristics such as completeness, correctness, testability, lack of ambiguity, and consistency are the top ones mentioned.

However, alternative techniques like SMART and INVEST [Wake 2003] have been present as well, specially to complement attributes that are not covered by traditional references, such as the size of requirements and the importance of them to the client.

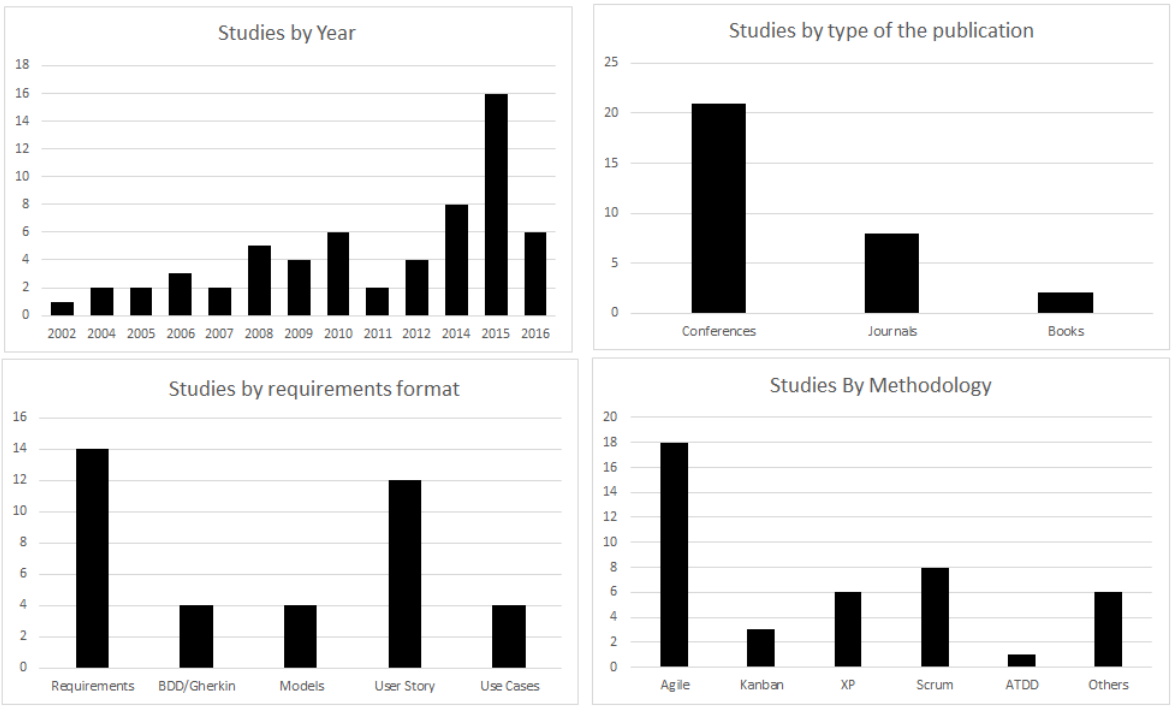

Figure 2. Studies overview

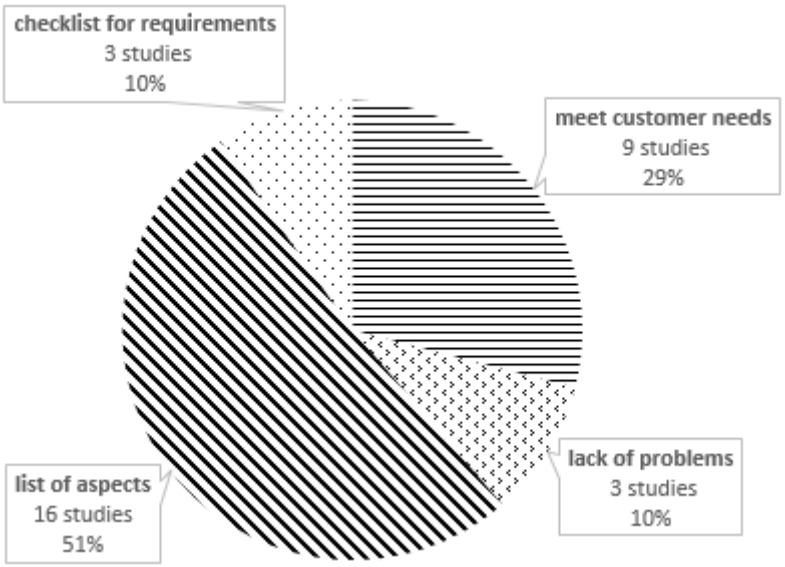

Figure 3. Summary of answers to RQ1

Quality Aspect	Paper IDs	Count
Complete	P8, P9, P11, P12, P13, P20, P24, P26, P33, P34, P43, P45, P47, P81, P108, P109, P114, P121, P159, P166	20
Correct	P8, P9, P11, P12, P24, P26, P33, P34, P43, P45, P73, P81, P108, P109, P136, P149, P159, P160	18
Testable	P2, P3, P8, P9, P11, P13, P24, P34, P40, P43, P45, P114, P121, P136, P159, P166	16
Unambiguous	P8, P9, P11, P13, P20, P24, P33, P34, P43, P47, P76, P109, P114, P121, P160, P166	16
Consistent	P9, P11, P13, P20, P24, P33, P34, P45, P47, P48, P108, P109, P121, P136, P159, P160	16
Traceability	P9, P11, P13, P24, P40, P45, P73, P121, P136, P159	10
Small	P2, P3, P9, P11, P20, P45, P114, P121, P159	9
Valuable	P2, P3, P9, P11, P30, P73, P114, P121, P159	9
Understandable	P28, P34, P45, P48, P81, P109, P136, P159	8
Independent, modular	P1, P2, P3, P9, P11, P45, P114, P121, P159	8
Specific	P9, P11, P24, P30, P33, P109, P159	7
Estimable	P2, P3, P9, P114, P121, P136	6
Negotiable	P2, P3, P9, P114, P121	5
Modifiable	P33, P43, P114, P136, P159	5
Viable	P9, P43, P109, P136, P159	5
Measurable	P9, P11, P20, P24, P149	5
Well Formed	P11, P20, P114, P121	4
Uniform	P9, P11, P24	3
Creative	P3, P76	2
Time bounded	P9, P11	2
Pragmatic	P24	1
Expressive	P13	1

Table 2. List of aspects on each accepted paper

The concern with requirements traceability highlighted on [Inayat et al. 2015] is also one criteria that appears on some publications (10 out of 31).

Other characteristics of requirements, such as expressivity, uniformity, and language clarity are not taken into account that often. It may indicate that the writing of the requirements description is not a concern on agile requirements, due to the practices found on [Inayat et al. 2015] that focus on face-to-face communication to define details.

5. Conclusion

The current study has performed a systematic literature review following the procedures described on [Kitchenham and Charters 2007] with the goal to answer what is a ”good” requirement and what aspects are used to evaluate it on projects or teams that follow agile development. This initial study is part of a major research project that aims

to improve agile RE documentation and project outcomes. We will focus on using the Behavior-Driven Development approach [Smart 2014] as a theoretical background to frame our investigation given its promising benefits of facilitating communication, establishing a common language for development team and customer to used when defining the product scope, and providing up-to-date documentation at any given time (called 'living documentation'). These are aspects often pointed out as challenges to help teams to succeed even in the agile era. We aim to empirically investigate how good agile RE and related documentation are provided by the adoption of the referred approach aiming to propose guidelines to facilitate the writing process of requirements documentation and the improvement of project outcomes (e.g., quality of source).

Therefore, our **RQ1** focused on what is the concept of requirements quality on agile development. We found that the majority of the studies faced quality of requirements as a list of aspects to be followed and verified - the documents matching those aspects are considered to be "good" ones. The listing of those aspects were the answer to our **RQ2**. The majority of the studies focus on the traditional aspects found on [IIBA 2009] and [IIBA 2015], but some of them are already pointing out to new criteria as those found on [Wake 2003].

We now have enough information to be discussed with agile practitioners in order to develop a clearer picture of what requirement quality is and what aspects it involves–our second step before moving to our core empirical study of the use of Behavior-Driven Development and the quality of requirements and project outcomes.

5.1. Study Limitation

Even with the claim that Scopus is the largest abstract and citation database of peer-reviewed literature, we realize that we could have directly used some of the databases indexed by Scopus in order to double-check whether all relevant studies are included in our literature review. We tried to minimize this risk by reusing known synonyms to our terms, taken from other systematic mapping and reviews that had covered other sources that we did not.

5.2. Final Consideration

Our long-term research goal aims to identify how Behavior-Driven Development can help better documenting requirements and generating project outcomes with the goal of providing tools for project managers to avoid often repeated drawbacks in software development such as obsolete documentation and requirements misunderstandings. These, year after year, are reported as reasons for project failure, which cause waste of resources and financial costs. Agile is present for over a decade now and we are still to see studies showing how much the adoption of agile methodologies have helped software teams to reduce costs and help customers to better achieve their business goals.

References

ElAttar, M. and Miller, J. (2012). Constructing high quality use case models: a systematic review of current practices. *Requirements Engineering*, 17:187–201.

Fabbri, S., Hernandes, E. M., Thommazo, A. D., Belgamo, A., Zamboni, A., and Silva, C. (2012). Managing literature reviews information through visualization. In *International Conference on Enterprise Information Systems*, pages 36–45.

Heikkila, V. T., Damian, D., Lassenius, C., and Paasivaara, M. (2015). A mapping study on requirements engineering in agile software development. In *Euromicro Conference on Software Engineering and Advanced Applications*.

IIBA (2009). *A Guide to the Business Analysis Body of Knowledge (BABOK Guide) 2nd Edition*. International Institute of Business Analysis.

IIBA (2015). *A Guide to the Business Analysis Body of Knowledge (BABOK Guide) 3rd Edition*. International Institute of Business Analysis.

Inayat, I., Salim, S. S., Marczak, S., Daneva, M., and Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*.

Kent Beck, Alistair Cockburn, M. F. et al. (2001). Agile manifesto: Manifesto for agile software development. Visited in: 2016-08-13.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.

Langenfeld, V., Post, A., and Podelski, A. (2016). Requirements defects over a project lifetime: An empirical analysis of defect data from a 5-year automotive project at bosch. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 145–160.

Medeiros, J., Alves, D., Vasconcelos, A., Silva, C., and Wanderley, E. (2015). Requirements engineering in agile projects: A systematic mapping based in evidences of industry. In *Ibero-American Conference on Software Engineering*, pages 460–473.

Paetsch, F., Eberlein, A., and Maurer, F. (2003). Requirements engineering and agile software development. In *Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 308–313.

Patel, C. and Ramachandran, M. (2010). Best practices guidelines for agile requirements engineering practices. *Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization*.

Smart, J. (2014). *BDD in Action: Behavior-Driven Development for the Whole Software Lifecycle*. Manning Publications, Shelter Island, NY.

The standish group (2015). CHAOS. https://www.projectsmart.co.uk/white-papers/chaos-report.pdf. Visited in: 2016-08-13.

Tiwari, S. and Gupta, A. (2015). A systematic literature review of use case specifications research. *Information and Software Technology*, 67:128–158.

Usman, M., Mendes, E., and Börstler, J. (2015). Effort estimation in agile software development: A survey on the state of the practice. In *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering*. ACM.

Wake, B. (2003). INVEST in good stories and SMART tasks. http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/. Visited in: 2016-08-13.