

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:

2021-05-31

Deposited version:

Accepted Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Mattos, L. F. B., Oliveira, A. & Ribeiro, R. (2019). Cooperation between university and industry in the research of the suitability of agile methodologies in software development. In Chova, L. G., Martinez, A. L. and Torres, I. C. (Ed.), Proceedings of ICERI2019 Conference. (pp. 1303-1311). Seville, Spain: International Academy of Technology, Education and Development (IATED).

Further information on publisher's website:

[10.21125/iceri.2019.0396](https://doi.org/10.21125/iceri.2019.0396)

Publisher's copyright statement:

This is the peer reviewed version of the following article: Mattos, L. F. B., Oliveira, A. & Ribeiro, R. (2019). Cooperation between university and industry in the research of the suitability of agile methodologies in software development. In Chova, L. G., Martinez, A. L. and Torres, I. C. (Ed.), Proceedings of ICERI2019 Conference. (pp. 1303-1311). Seville, Spain: International Academy of Technology, Education and Development (IATED)., which has been published in final form at <https://dx.doi.org/10.21125/iceri.2019.0396>. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

COOPERATION BETWEEN UNIVERSITY AND INDUSTRY IN THE RESEARCH OF THE SUITABILITY OF AGILE METHODOLOGIES IN SOFTWARE DEVELOPMENT

L.F. Bordeaux Mattos¹, A. Oliveira², R. Ribeiro²

¹ Instituto Universitário de Lisboa (ISCTE-IUL) (PORTUGAL)

² Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR-IUL (PORTUGAL)
lfbms11@iscte-iul.pt, abilio.oliveira@iscte-iul.pt, ricardo.ribeiro@iscte-iul.pt

Abstract

Choosing a proper approach to manage a software development project is critical to increasing the chances to succeed in the venture. One of the great dilemmas nowadays concerns the option for an Agile or a traditional approach. There are several characteristics of software projects and the business environment in which they are carried out that must be considered while choosing the most appropriate development process. Although Agile development methodologies have been increasingly expanding and consolidating worldwide as an effective way of building software in the last two decades, they are not a one-size-fits-all approach. Knowing when to use Agile and which methodology is most suitable are the great questions that we aimed to answer in this study.

In this paper, through a comprehensive revision of the Agile literature and an exploratory study, with the realization of focus group sessions with Agile practitioners and an interview with an Agile expert actively engaged in the software industry, we explored the characteristics of software development projects and organisational environments, identifying those that favour the use of an Agile approach. We also studied the main aspects that lead Agile teams to opt for one of the common Agile frameworks: Scrum, Extreme Programming (XP), Kanban, or Lean Software Development (LSD). The major goal of our study, clarifying the main aspects to be considered in the choice of an Agile methodology, is to support the decision-maker in the process of selecting an appropriate development approach.

Keywords: Agile software development, Agile suitability, Agile frameworks, Agile methodologies.

1 INTRODUCTION

In the last two decades, software development projects showed an increasing need to produce solutions in a simpler, faster, and cheaper way, as well as with higher flexibility regarding changes in requirements due to the dynamics of the business environments. The linear approach with one-way analysis, design, construction, and test became outdated concerning the adaptability to an evolving scenario [1], clearing the way to the emergence of Agile approaches, capable of delivering faster results, more aligned to the customer's needs [2].

Since the formal creation of Agile concepts in 2001, the software development industry has been gradually increasing the adoption of Agile methodologies. A recent survey pointed out the top five reasons for the adoption of Agile methodologies: 1) accelerate software delivery, 2) enhance the ability to manage changing priorities, 3) increase productivity, 4) improve business / IT alignment, and 5) enhance software quality. The survey also identified the main benefits of adopting Agile as the ability to manage changing priorities, increased project visibility, better business / IT alignment, higher team morale, and delivery speed / time to market [3].

Despite the trend, the truth is that Agile is not an approach applicable to all software development scenarios. Knowing when to use Agile and which of its methodologies is more suitable for the software to be built is a complex task given the many combinations of aspects that may influence the choice of the most appropriate option. The existent Agile suitability filters [4], models that are supposed to help in determining the likely fit or gap for using Agile approaches [5], are relatively obsolete and, moreover, do not show evidence of application by the software industry. We can add to these factors the lack of scientific research related to the methods for selecting and choosing an adequate software development methodology, especially about Agile approaches. Therefore, it is possible to affirm there is no insightful data available from which we can easily deduct the suitability of an Agile methodology for a given software development context.

In this study, we applied a conceptual-analytical approach [6]. A comprehensive revision of Agile literature was combined with an exploratory study composed of focus groups sessions with Agile practitioners and an interview with one Agile specialist (and relevant book author). These rich

information sources and contributors provided meaningful data and insights on the subject researched, which have been qualitatively analysed with the help of Leximancer, a software that analyses document contents – textual data –, allows context analysis, and visually displays the relationship among the extracted information – summarized in themes and concepts [7].

The emerged key concepts suggest three major groups of concepts, best defined as categories, to consider while assessing the Agile approach suitability. These categories encompass relevant factors (N=9) to consider while performing the assessment. Our findings also suggest five main aspects to take into consideration while choosing among Scrum, Extreme Programming (XP), Kanban, or Lean Software Development (LSD) as the most suitable Agile methodology to use in a software development effort.

2 BACKGROUND

2.1 The Agile Approach

The literature defines Agile as an approach for software development based on an iterative and incremental concept which, instead of in-depth planning at the beginning of the project, embraces changing requirements over time and encourages constant feedback from the business [2]. Agile cross-functional teams work on iterations to build a product over a fixed period of time. This work is organised into a product backlog, which is prioritised based on business values. The goal of each development cycle is to produce working valuable software [5]. In Agile, leadership encourages teamwork, accountability, and face-to-face communication. Business stakeholders and developers work together to align the software purpose with business needs and company goals. More than the application of these ideas in combination, Agile involves adopting a new mindset based on Agile values and principles [8]. These values and principles are the central elements of Agile practices that, combined, form the different types of methodologies.

2.2 Agile Suitability Models

Agile suitability models are references that help in determining the likely fit or gap for using Agile approaches. Such models assess the factors associated with Agile suitability to provide recommendations [5]. In this way, these models were considered a starting point for our research on the characteristics that could support the option for an Agile approach. Even though we did not find concrete evidence of practical application of such models, they provided significant indications on the major aspects we should focus during our investigation. Four models for assessing Agile suitability have been analysed.

“The Slider” is an Agile suitability model created to show that an Agile or a traditional choice is not necessarily binary [4]. It was designed aiming to point out a “no”, a “mixture”, or a “fully” suitability for an Agile approach, according to the assessment of certain characteristics. The existence of an innovative culture; higher levels of uncertainty or instability in the environment for the development; high customer availability and responsiveness; and less criticality would favour and Agile approach according to the model. In the opposite direction, Agile usage would be more difficult and might favour a more traditional approach. The Slider model suggests that, in case of mixed results, a combination of Agile and traditional methods could suit well.

“The DSDM Suitability Filter” is derived from the software development methodology Dynamic Systems Development Method (DSDM). The DSDM Suitability Filter consists of a list of “Yes / No” questions whose idea is to check conformance to project characteristics that favour Agile [4]. According to the model, the presence or absence of characteristics such as the acceptance of the Agile philosophy before starting work; the concept of empowered teams; significant end-user engagement and easy access to these end-users; the stability of the development team, their skills and size; a supportive relationship based on trust and collaboration; and the technology provide clear indication of an adequate environment to use Agile. The model does not suggest that the absence of these attributes indicates that Agile is inadequate, rather, it highlights potential risk areas to manage.

“The Boehm & Turner Model” suggests a visual way of assessing the project characteristics that its authors believe helping in the determination of the Agile approach suitability. The idea is to assess the project along five main attributes: Personnel, since Agile initiatives run better with a lower number of beginners and a higher number of experienced developers; Dynamism, measuring the likelihood of changes in the development effort (the higher, the more suited to an Agile approach); Culture, verifying whether the organisation is ready to accommodate concepts such as emergent requirements, empowered teams, and servant leadership; Team Size, because Agile initiatives are easier executed

and managed using small teams; and Criticality, since Agile is more suitable for the development of applications where failure results in loss of convenience [4].

“The PMI Model” was inspired by the previous suitability models. However, the model expanded its application to consider the middle ground of hybrid methodologies aiming to help organisations to determine whether software development initiatives could be best undertaken using predictive, hybrid, or Agile approaches. It consists of the evaluation of organisational and project attributes such as Cultural aspects, to check if a supportive environment with buy-in for an Agile approach is in place; Team characteristics, verifying if the development team has a suitable size to be successful in using Agile and if its members have the necessary experience and easy access to business representatives; and Project attributes, evaluating rates of changes, the possibility of an incremental delivery, and project criticality [5].

3 RESEARCH METHOD

To undertake our study, we conduct a qualitative research comprising a comprehensive literature revision, complemented by an exploratory study. In the literature revision, characteristics, concepts, best practices, main applications, success cases, previous studies, and suitability models of Agile were revised. Publications whose authors are Agile researchers, specialists, and practitioners framed our work and provided meaningful insights for the analysis. In complement to the literature revision, an exploratory study composed by an interview with one Agile expert and three focus groups with Agile practitioners (N=15) has been carried out. The interview and focus groups aimed to collect the opinions of the participants on the factors that might support the choice of an Agile approach in software development. Another goal was to discuss and identify the main characteristics of software development projects and organisational environments that favour the use of Scrum, Extreme Programming, Kanban, and Lean Software Development within a given development context. For the qualitative analysis of the data obtained, we used the software Leximancer.

3.1 Data Collection

Data collection process involved one interview and three focus group sessions held in different cultural and organisational environments, with participants having diverse levels of experience in Agile. The main objective was to capture various perceptions on the subject.

We interviewed an author / specialist with 20 years of experience in Agile and software development. The interviewee works for the European Commission (EC) in Brussels, Belgium, and is in charge of scaling Agile practices within the EC. The first focus group gathered four middle management professionals with low experience in Agile from a Humanitarian organisation with an office in Brussels, Belgium. The organisation has been working on the implementation of Agile practices in some of its projects. The second focus group was composed of six IT project managers from a Portuguese organisation of the Banking sector. They are experienced in Agile (on average two and a half years), and the organisation uses Agile in their system development projects. The third focus group put together five Agile developers from a Brazilian company of the Telecommunications sector, with strong experience in Agile (on average five years). The organisation has been using Agile methodologies in IT projects for many years.

The first specific objective of the interview with the Agile specialist and the focus groups with Agile developers and practitioners was to verify the factors that might support the choice of an Agile approach in software development. The second specific goal was to explore the characteristics that may help in indicating the best Agile methodology to be used among Scrum, Extreme Programming, Kanban, or Lean Software Development. The questions created to stimulate the debate among the participants focused on the conditions that favour the use of an Agile approach; the characteristics and behaviours that leaders, teams, and members of development teams must have to work with an Agile methodology; the adequate environmental factors (physical or cultural) for a software development project to use an Agile methodology; and the characteristics the resulting software may have that contribute to the use of an Agile methodology. A specific set of questions to explore the knowledge of the participants on the aspects that influence the choice of the Agile methodology (among the four studied) was elaborated and debated in the activities as well.

3.2 Data Analysis

The data collected in the interview and focus groups were analysed with the help of the software Leximancer, which allowed a qualitative analysis of the non-structured data collected. The concept maps generated (cf. Fig.1 and Fig.2) highlighted meaningful concepts and showed the relationship among

them. From the analysis of these concepts, it was possible to identify more clearly: 1) the factors that may support the choice of an Agile approach in software development; 2) the aspects that help to point out the best Agile methodology to use (among the four studied).

4 FINDINGS AND DISCUSSION

A concept map creates a view of the conceptual structure of the content of the data collected. The interactive nature of the map permits to explore the concepts and their connections to each other [9]. It allowed us to understand how the different concepts were linked, facilitating the confirmation of clearer relationships and the identification of non-evident relationships among them. To analyse the data collected regarding the aspects related to the choice of an Agile approach, we generated the first concept map used in this research (see Fig.1).

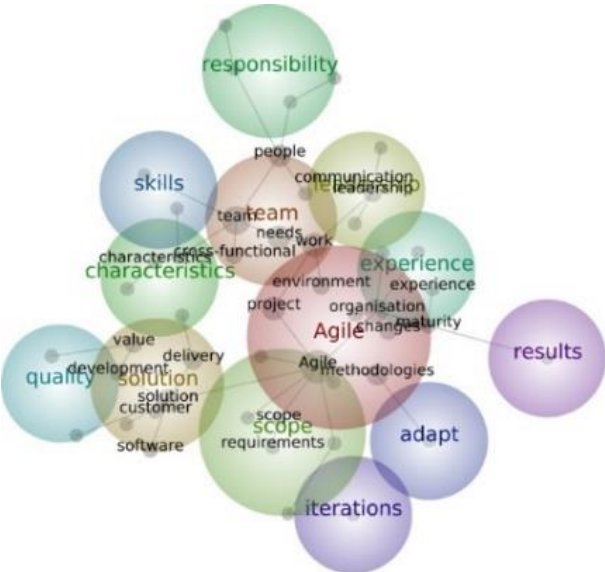


Figure 1. Map of the concepts related to the choice of an Agile approach.

This concept map enabled us to obtain insights and to extract additional inputs that emerged from the experiences of the participants.

The second concept map generated (see Fig.2) allowed us to explore the aspects related to the selection of an Agile methodology (among the four studied) to be used in a given software development initiative. We verified the key aspects that contribute to point out in which cases we should use the Agile frameworks Scrum, Extreme Programming, Kanban, and Lean Software Development.



Figure 2. Map of the concepts related to the choice of an Agile methodology.

4.1 Factors that may support the choice of an Agile approach

Three groups of concepts, or categories, emerged as a result of the analysis of the data related to the factors that can help with the choice of an Agile approach. The first category we have identified is the “Organisational Environment”. It includes the concepts associated with the organisational context where the software is developed. The second category is “People”, concerning the characteristics, behaviours, and competencies of the leadership and the teams involved in software development. The third category is the “Project Environment”, encompassing the concepts related to project management capabilities, the context for the project that will be undertaken, and its product-related aspects. The relevant factors that support the choice of an Agile approach within each of these categories were identified as a result of the data analysis and are summarised in Table 1.

Table 1. Factors to consider in an Agile approach suitability assessment.

Category	Factor
Organisational Environment	<ul style="list-style-type: none"> • Agile maturity; • Organisational structure; • Change management maturity; • Collaborative work level.
People	<ul style="list-style-type: none"> • Leadership characteristics and behaviours; • Team characteristics, behaviours, and competencies.
Project Environment	<ul style="list-style-type: none"> • Project Management capabilities; • The context for the software development project to be undertaken; • Product characteristics.

The relevant Organisational Environment factors that emerged as a result of the exploratory study involves the Agile maturity, the Organisational structure, the Change management maturity, and the Collaborative work level in the organisation. The Agile maturity may be best evaluated verifying the historic of Agile practices in the organisation. According to Kourounakis et al, it can range from no history at all to the adoption of Agile approaches at a strategic level [10]. Regarding the Organisational structure, the point is to identify if the organisation is structured around projects or primary functions. The more projectized the organisation is, the easier is the scalation and usage of Agile practices according to our findings. In relation to the Change management maturity, the idea is to assess whether change management competency is part of the practices in the organisation or not. If the organisation does not have guidelines for managing changes, a proper application of a development process that is open to embrace frequent changes is more difficult. To better evaluate the aspects related to Collaborative work, we consider that, in the organisation, the level of cooperation, coordination, collaboration, and communication must be analysed. The higher the level of these factors, the more promising is the environment to work with Agile.

In the People category, the main factors to be considered relate to the characteristics, behaviours, and competencies of the leadership and the development teams. Our results corroborate many of these characteristics, behaviours, and competencies found in the Agile literature. Some other non-obvious were unveiled in the research, such as a characteristic required by leaders, which is to know very well the team members and stay close to them to be familiar with important themes and act as quickly as possible to unblock issues. A behaviour that the development team should present also came up from the experiences of the participants. It relates to the seek for the working agreements within the team, which means, the alignment of expectations and values among the team members.

To assess the Project Environment, three factors proved to be essential according to our study: the project management capabilities, the context for the software development project to be undertaken, and the product characteristics. The findings suggest that project management capabilities can be classified from no formal implementation of project management practices to a comprehensive, fully integrated approach to managing projects [11]. The higher the project management capabilities, the easier and more effective Agile can be used in the project in question. Specific characteristics of the project can be verified to check if its context favours Agile or not. We suggest three project characteristics to consider in this case: the degree of uncertainty on the system requirements; the real necessity for co-locating the project team and business representatives during the project; and the need for the project to apply tools and techniques to enhance communication and collaboration. Regarding the aspects related to the software resulting from the project – the product –, our findings recommend

considering the possible impacts of defects, and also the feasibility to adopt an approach where the product can be delivered in parts.

4.2 Aspects that must be considered in the option of the adequate Agile methodology

In the option of the most suitable Agile methodology among the four studied, our findings reinforce the approach of focusing on the differences among the frameworks as a way to uncover the aspects that deserve greater attention. Supported by this idea, we unveiled five aspects to consider in order to perform the assessment (in the context of a software development effort). These aspects are presented in Table 2 by order of relevance, from the most (1) to the least relevant (5), according to the results of our research.

Table 2. Aspects to consider while choosing an adequate Agile methodology.

Relevance	Aspect
1	The emphasis of the methodology.
2	The way the methodology addresses changes in requirements.
3	The way the methodology approaches the delivery.
4	The way the methodology deals with customer engagement.
5	The way the methodology approaches prioritisation.

The emphasis of the methodology is the aspect that most differentiates the four methodologies. In Scrum, the project requires mainly a managerial framework, with a focus on productivity, ensuring the software is built quickly and with minimum complexities. An iterative and incremental approach is required to periodically release software increments. Extreme Programming applies an iterative and incremental approach as well, but the highest priority is the quality of interim and final software. To deliver it, the methodology focuses on applying sophisticated software engineering practices and works with shorter development cycles. Kanban, in turn, focuses on improving the workflow and refraining from producing a surplus throughout full transparency of work and real-time communication of capacity. For this, it is key to have a clear view of the development process and address the issues as fast as possible. Lean Software Development shares principles similar to Kanban. However, its main concern lays on to streamline the development process, cutting away all activities that do not directly contribute to the final software. Lean Software Development does not demand to work with fixed-length development cycles, but it relies on an experienced team to apply the best software engineering practices to deliver the project.

Another distinguishing aspect among the methodologies studied is the way each methodology addresses changes in requirements during the development process. In Extreme Programming, changes are expected, but usually, the related implementations and modifications in the system are not performed during an ongoing development cycle. If a big or exceptional change is needed, the project team typically stops the current cycle and starts a new one with new requirements. In Scrum, changes in requirements are also expected, but while working with fixed-length iterations, implementing these changes during an iteration is accepted only when strictly necessary. The team members will make their best effort to maintain the planned workload throughout the reprioritisation of features. Lean Software Development and Kanban share similar characteristics regarding changes in requirements. For both, frequent changes in requirements and scope are the norm. The main difference lays in the way each one reacts to it. In Lean, these changes are typically accepted, implemented, and delivered as fast as possible because one of its priorities is to deliver value to the customer as soon as possible. In Kanban, changes are accommodated in the backlog according to the priority and executed when there is available capacity in the team.

The way the methodologies approaches the delivery may be divided into two, the ones which privilege a more planned manner, and the ones that require the delivery to be ad-hoc or as fast as possible. We can affirm that to work with Scrum or Extreme Programming, the development effort requires a process that enables frequent delivery of parts of the software in a planned way. The scope of the releases mostly contains business features required by the customer. There is flexibility for changing requirements, but the implementation of these changes must keep a balanced backlog, minimizing risks for the continuity of the delivery and the quality of the software. On the other hand, to use Kanban or Lean Software Development, the project requires an approach that enables the delivery of parts of the

software as fast as possible or when required by the customer. There must be flexibility for the implementation and rapid availability of changes. It is essential to have an optimized and constantly revised workflow, minimizing risks of bottlenecks or performing tasks that do not directly contribute to the software.

The way the methodology deals with customer engagement is another aspect that makes possible to differentiate these methodologies. In Extreme Programming, business representatives are actively engaged in software development daily, integrating the project team. In Scrum, the customer frequently interacts with the project team and participates in key project activities, like planning sessions and reviews of the work done, providing frequent feedback on software increments. In contrast, the results indicate that in Lean Software Development and Kanban, the customer or business representatives are mainly involved in the development process to identify value streams and to contribute with the optimization of the workflow.

The way the methodology approaches prioritisation in an Agile development effort is another aspect that allows distinguishing the methodologies. Although the different approaches might not be strictly applied as described here, in general, they are close to these definitions. In Scrum, the customer is the one in charge of setting priorities among the features within a sprint (iteration). However, the project team performs the work in the sequence they deem to be adequate for the objectives set for the sprint. In Extreme Programming, the team works on the features planned for the iteration in the order defined by the customer, and the project team cannot change this order. In Kanban, the prioritisation of the job is established by whoever is in charge of it (the development team or the customer), according to the needs of the business. Here the main concern is pulling work and often reprioritise the remaining tasks to keep a paced workflow. In Lean Software Development, like in Kanban, the prioritisation is done by whoever is responsible for this, but the objective is to avoid unnecessary work and to prioritise tasks that directly add value to the resulting software.

5 CONCLUSIONS

In this paper, we explored the factors that favour the use of an Agile approach in software development projects – part 1 – and the main aspects that lead development teams to choose one Agile methodology among Scrum, Extreme Programming, Kanban, or Lean Software Development – part 2 – aiming to support the decision-maker in the process of selecting an appropriate development approach.

Regarding the first part of the study we conclude that there are nine factors, divided into three categories, that we should consider in the decision process for an Agile approach: the Agile maturity (1), the organisational structure (2), the change management maturity (3), and the collaborative work level (4), with regard to the organisational environment; the characteristics, behaviours and competencies of leaders and teams concerning the people factors (5, 6); the project management capabilities (7), the context for the software development project (8), and the software characteristics (9) in relation to the environment for the project.

For the second part we can conclude that there are five main aspects that we should assess while choosing one Agile methodology (among the four studied): the emphasis of the methodology (1); the way the methodology addresses changes in requirements (2); the way the methodology approaches the delivery (3); the way the methodology deals with customer engagement (4); and the way the methodology approaches prioritisation (5).

Therefore, we can say that our major goal – and great contribution of this study – has been reached, which is to clarify what is important to consider in the choice of an Agile methodology, helping the decision-maker in the option for an appropriate development approach.

As a suggestion for the continuity of this study, aiming to expand the present results, we recommend conducting exploratory studies with groups whose experiences with Agile resulted in unsatisfactory outcomes. It would provide a better view of when not to use Agile. Another approach that could be explored in future researches is expanding the group of Agile methodologies studied, including the hybrid ones, such as the Spotify Squad framework or SCRUMBAN.

REFERENCES

- [1] J. Sutherland, *Scrum: The Art of Doing Twice the Work in Half the Time*. New York, NY: Random House Business Books, 2015.
- [2] M. Griffiths, *PMI-ACP® Exam Prep: A Course in a Book for Passing the PMI Agile Certified Practitioner (PMI-ACP) Exam*. Minnetonka, MN: RMC Publications, 2015.

- [3] CollabNet VersionOne, "The 13th Annual State of Agile Report," CollabNet VersionOne, Alpharetta, GA, 2019.
- [4] M. Griffiths, "Agile Suitability Filters," 2007. [Online]. Available: https://leadinganswers.typepad.com/leading_answers/files/agile_suitability_filters.pdf.
- [5] PMI, *Agile Practice Guide*. Newtown Square, PA: Project Management Institute, 2017.
- [6] P. H. Järvinen, "Research Questions Guiding Selection of an Appropriate Research Method," in ECIS 2000, 3-5 July, Vienna, Austria, 2000.
- [7] L. F. Rodrigues, A. Oliveira and H. Rodrigues, "Main gamification concepts: A systematic mapping study," 17 July 2019. [Online]. Available: <https://www.heliyon.com/article/e01993>.
- [8] R. C. Martin, *Agile Software Development, Principles, Patterns, and Practices*. London, UK: Pearson Education, 2003.
- [9] Leximancer Pty Ltd, "Leximancer User Guide, Release 4.5," 12 April 2018. [Online]. Available: <https://doc.leximancer.com/doc/LeximancerManual.pdf>.
- [10] N. Kourounakis, A. Maraslis, I. Velitchkov, L. Sequeira and O. Beghain, *Agile @EC Guide*. v.1.3.2 ed., Brussels: European Commission Advisory Services, 2015.
- [11] M. E. Mullaly and J. Thomas, "Re-thinking project management maturity: perspectives gained from explorations of fit and value," in PMI® Research Conference: Defining the Future of Project Management, Washington, DC, 2010.
- [12] M. O. Ahmad, J. Markkula and M. Ovio, "Kanban in software development: A systematic literature review," in 39th Euromicro Conference on Software Engineering and Advanced Applications, Santander, Spain, 2013.
- [13] K. Beck, *Extreme Programming Explained: Embrace Change*. Boston, MA: Addison-Wesley Professional, 2004.
- [14] B. Boehm and R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed*. Boston, MA: Pearson Education, Inc, 2003.
- [15] E. Hossain, M. A. Babar and H.-y. Paik, "Using Scrum in Global Software Development: A Systematic Literature Review," in Fourth IEEE International Conference on Global Software Engineering, Limerick, Ireland, 2009.
- [16] C. R. Kothari, *Research Methodology: Methods and Techniques*, 4th Edition ed., New Delhi, India: New Age International, 2008.
- [17] M. Poppendieck and T. Poppendieck, *Lean Software Development: An Agile Toolkit*. Boston, MA: Addison-Wesley Professional, 2003.
- [18] J. Widman, S. Y. Hua and S. C. Ross, "Applying Lean Principles in Software Development Process – A Case Study," *Issues in Information Systems*, pp. 635-639, 2010.
- [19] J. Stapleton, *DSDM, Dynamic Systems Development Method: The Method in Practice*, Boston, MA: Addison-Wesley, 1997.