Ordering the Product Backlog in Agile Software Development Projects: A Systematic Literature Review

Ana Silva[§], André Silva[§], Thalles Araújo[§], Renan Willamy[¶], Felipe Ramos[¶],
Alexandre Costa[¶], Mirko Perkusich[¶], Ednaldo Dilorenzo[§]
[§]Federal Institute of Paraíba
PB-264
Monteiro, Paraíba, Brazil, 58500-000
{anasilva.ifpb, tr.andreluis, thalleshenrique.na}@gmail.com, ednaldo.dilorenzo@ifpb.edu.br
[¶]Federal University of Campina Grande
Campina Grande, Paraíba, Brazil, 58429-140
{renanwillamy, feliperamos, antonioalexandre, mirko.perkusich}@copin.ufcg.edu.br

***Abstract*—Recently, agile software development methods have attracted the attention of academic and industrial domains. Unlike traditional approaches, agile methods focus on rapid delivery of business value to customers through fast delivery of working code. Therefore, requirement prioritization is considered a crucial process in this context, since there is the need to identify which requirements must be developed first, based on stakeholders preferences and taking into account business and technical challenges such as dependencies of tasks, developers skills, limitation of budget, etc. In Scrum, the most popular agile method, requirement management is done by maintaining the product backlog, which contains the list of functionalities that must be developed. Although some works conducted systematic reviews on the topic of agile requirement engineering, to the best of our knowledge, there is no study that reviewed the specific subject of ordering the product backlog in agile projects. Therefore, in this work, we conducted a Systematic Literature Review of studies on this topic published up to (and including) 2016 through a database search, to identify and analyze factors and techniques used to accomplish the task of ordering the product backlog in agile projects. In total, we evaluated 1556 papers, of which 13 reported on the subject of study.**

Keywords*—Systematic Literature Review; Product Backlog; Agile Software Development; Requirements Management; Requirements Prioritization.

I. Introduction

Due to the need of flexibility in software projects, agile software development (ASD) methods (e.g., Scrum and Extreme Programming) are becoming more popular in the last years [3]. ASD promises some benefits compared to traditional software development methods, including the delivery of business value in short iterations, following an incremental and empirical development process [10].

Moreover, ASD methods promotes the constant communication (e.g., face to face communication) and a development process open to changes [7]. Therefore, requirements are initially defined with customers, but are continuously refined [7]. However, as well as in traditional methods, a relevant activity of ASD methods is requirement engineering (RE) [10].

In the context of Scrum, the most popular agile method, requirement management is accomplished by maintaining the product backlog [7]. Therefore, to ensure a rapid delivery of business value to costumer, it is necessary to identify which requirements in this list must be developed first (ordering of requirements), i.e., to order the product backlog taking into account business and technical factors [6].

The ordering of requirements is treated as a complex multi-criteria decision making process [1], since it aims to aid the early implementation of core requirements based on the preferences of relevant stakeholders, but considering the challenges associated with software development such as limitation of budget and resources, technical knowledge of the software team, etc. Thus, it is considered to be a hard task to define and maintain [6].

Additionally, in the context of agile, requirement ordering is a continuous task, since requirements are constantly changing during iterations [10]. However, requirement engineering in agile is still informal and based on the knowledge of individuals. For example, in Scrum, the Product Owner (PO) is responsible for the elicitation and the ordering of the product backlog list [3]. Therefore, these issues can be mitigated by applying techniques to assist individuals involved in the decision-making process of ordering the product backlog.

Thus, the ordering of requirements is a crucial process in ASD projects, since it is one of the main processes to produce value quickly [6]. However, although some previous works mapped the subject of agile requirement engineering [3], [10], [7], to the best of our knowledge, research on the ordering of product backlog in ASD projects has not been systematically reviewed yet. Therefore, the main goal of our work is to conduct a Systematic Literature Review (SLR) to identify and analyze factors and techniques used to order the product backlog in ASD projects. To accomplish this work, we followed the guidelines proposed by Kitchenham and Charters [8].

DOI reference number: 10.18293/SEKE2017-007

This paper is structured as follows. In section 2, we discuss previous literature reviews in the subject of study. In Section 3, we present the protocol of our systematic review process. In Section 4, we present our findings. In Section 5, we discuss the results. In Section 6, we present our conclusions and future work.

II. Related Work

Some literature reviews were conducted on the topic of agile requirement engineering [7], [10], [6], [3]. We summarize them as follows.

Inayat et al. [7] conducted a systematic literature review focusing on practices and challenges of requirements engineering in the context of ASD. The review identified 17 practices of RE in agile, five challenges found in traditional methods which were overcame by agile requirements engineering, and eight challenges resulting from the practice of RE in agile. The authors suggest that the subject needs further investigation and more empirical results to improve its understanding.

Schön et al. [10] conducted an SLR on the topic of agile RE with a focus on the participation of stakeholders and users in the process of RE. The authors investigated existing approaches to involve stakeholders in this stage and which approaches are used to present the user's perspective during these process. They concluded that agile RE is a complex research field with cross-functional influences and that their study brings an overview of requirements management in ASD.

Heikkilä et. al [6] conducted a mapping study on the research literature of RE in ASD with focus on the benefits of this process in agile and the reported problems and corresponding solutions. The authors concluded that the agile RE definition is still vague. Among the main benefits, stand out: better requirements understanding, responsiveness to change and rapid delivery of value. Among the reported problems, stand out: user story requirements format, prioritization of requirements and imprecise effort estimation. Finally, the authors concluded that, in general, studies' evaluations are weak (i.e., need more effort in empirical evaluation) and the subject of study needs more research.

Unlike previously mentioned works, which cover the primary issues of agile RE, we focus on the identification and analysis of factors and techniques applied in the ordering of product backlogs, which is a more specific issue of agile RE.

In the area of software requirements prioritization, Achimugu et al. [1] conducted an SLR to identify and analyze existing prioritization techniques, their limitations, processes, and taxonomies. The authors concluded that existing prioritization techniques present limitations and their applicability in complex and real setting has not been reported yet. Moreover, it was considered that existing techniques need improvements. Although the work is related to ours, the authors did not focus on agile, and hence, few works were returned on this topic, which has been attracting the attention of an increasing amount of research studies and it is gaining popularity in the industry [6]. Furthermore, the authors did not evaluate the main factors used by the techniques in the ordering process, as reported in our work.

III. Review Method

In this research, we performed a Systematic Literature Review following the method defined in [8]. An SLR aims to identify, critically evaluate and integrate relevant, high-quality studies addressing one or more research questions.

A. Research Questions

As previously mentioned, the main goal of this research is to identify and analyze factors and techniques used in the ordering of product backlogs in ASD projects presented in the literature. Therefore, we formulated the following research questions (RQs):

RQ1: Which factors are considered to order the product backlog items?

RQ2: How are the factors identified in RQ1 measured?

RQ3: Which techniques are used to order the product backlog?

RQ4: Which evidences show that the techniques identified in RQ3 are efficient?

B. Data Sources and Search String

We conducted a database search in the following sources: ACM, Engineering Village, ISI Web of Science, ScienceDirect, SpringerLink, Scopus and Wiley. These data sources were chosen based on its relevance in the software engineering domain. Aiming to start to answer the research questions defined in section III-A, we formulated the following search string:

> *(prioritization OR prioritisation OR prioritizing OR prioritising OR prioritize OR prioritise OR prioritized OR prioritised OR priority OR order OR ordered OR ordering) AND (requirement OR functionality OR requisite OR prerequisite OR user story OR "user stories" OR backlog OR issue) AND (practice OR practices OR techniques OR technique OR process OR processes OR tactic OR tactics OR method OR methods OR strategy OR strategies OR factor OR factors OR component OR components) AND (software AND (agile OR gil) AND (scrum OR xp OR (crystal AND (clear OR orange OR red OR blue)) OR dsdm OR fdd OR "feature driven development" OR (lean AND (development OR desenvolvimento)) OR Kanban OR "extreme programming" OR "programao extrema" OR devops))*

The search results are presented in Table I.

TABLE I: Number of papers returned from database search.

Data Source	Results
ACM (http://dl.acm.org/)	22
Engineering Village (www.engineeringvillage.com)	43
ISI Web of Science	35
ScienceDirect (www.sciencedirect.com)	1,173
SpringerLink (www.scopus.com)	-
Scopus (www.scopus.com)	86
Wiley (onlinelibrary.wiley.com)	455
Total	1,814

		Reviewer 2		
		Relevant	Uncertain	Irrelevant
Reviewer 1	Relevant	A	B	D
	Uncertain	B	C	E
	Irrelevant	D	E	F

Fig. 1: Categories of agreement or disagreement.

C. Selection Criteria

Formulating a consistent search string does not guarantee a significant set of studies. Due to database query limitations, major part of the returned papers are not related to the research subject. To select only relevant studies, we have divided the selection criteria into three phases: generic exclusion criteria, basic criteria and advanced criteria.

Generic exclusion criteria. To eliminate the most irrelevant papers we decided to discard i) papers not published in English or Portuguese languages; and ii) published in non-peer reviewed publication channel such as books, thesis or dissertations, tutorials, keynotes, etc.; and iii) duplicated.

Basic Criteria. This phase was performed by two reviewers, randomly chosen, and it consisted of reading the title and abstract of the remaining papers from previous phase (generic exclusion). Each paper was evaluated and classified following the procedure presented in Ali et al. [2], as:

- *Relevant:* papers related to agile software development and to product backlog management;
- *Irrelevant:* papers not related to agile software development or not related to product backlog management;
- *Uncertain:* the available information on the title and abstract was inconclusive or insufficient to classify as relevant or irrelevant.

According to Ali et al [2], there are six categories of agreement or disagreement between the reviewers, as shown in Figure 1.

Categories A or B mean that at least one reviewer evaluated the paper as relevant and it is included. Category B occurs when one reviewer is uncertain about the relevance of the paper. To minimize the risk of discarding a significant study, the paper is included. Afterwards, in the next selection criteria step all doubts about the paper's relevance are clarified with a further evaluation.

Category C means that no concrete decision was made by any of the reviewers and further investigation is needed. In this case, both reviewers, independently, conduct an adaptive reading, which is composed of three steps: 1) read the Introduction; 2) if not having agreement in 1, read conclusion; 3) if not having agreement in 2, use the keywords to evaluate their usage to describe the context of the paper. This practice helps to make a decision.

Categories D and E are results from disagreement and the reviewers are asked to discuss what reasons led them to their respective decisions. After that, a consensus is expected and a new category (A, C or F) classification must be done.

Papers in category F are excluded, as both reviewers agreed on their irrelevance.

Advanced criteria. In this selection phase, a further investigation is performed, once a full-paper evaluation is required. We used the same criteria established in basic criteria, but reading the full-text. Each paper was evaluated by two reviewers, a data extractor and a data checker. The data extractor performs a Quality assessment (Section III-D) and a Data extraction (Section III-E). After the data is extracted, the data checker reviews to confirm if it is corrected, following the advice presented in Brereton et al. [4] and Staples and Niazi [12].

D. Quality assessment

In this stage, each paper was evaluated by a data extractor and a data checker, in the following manner: the data extractor fills the data extraction form, and then, the data checker confirmed that the data on the extraction form were correct. Each category was evaluated on a boolean scale (i.e., 0 or 1). The assessment checklist follows 11 criteria presented by Dyba and Dygsoyr [5].

E. Data extraction

During this stage, we used a spreadsheet editor to record information. For each paper, we extracted general information such as title, year and publication channel, and data related to the RQs. The following data were extracted from the papers:

(i) type of article (journal, conference),

(ii) name of the publication channel,

(iii) year of publication,

(iv) agile method,

(v) product domain,

(vi) application domain,

(vii) team size,

(viii) team distribution,

(ix) research type (based on the classification presented by Wieringa et al. [15]: validation research, evaluation research, solution proposal, philosophical papers, opinion papers or experience papers.),

(x) research question type (based on the classification presented by Shaw [11]: method or means of development; method for analysis or evaluation; design, evaluation, or analysis of a particular instance; generalization or characterization; or feasibility study or exploration.),

(xi) empirical research type (based on the classification presented by Tonella et al. [13]: experiment, observational study, experience report, case study or systematic review.),

(xii) research validation (based on the classification presented by Shaw [11]: analysis, evaluation, experience, example, persuasion or blatant assertion.),

(xiii) factors (RQ1),

(xiv) how factors were measured or identified (RQ2),

(xv) techniques (RQ3),

(xvi) evidences of techniques efficiency (RQ4).

IV. Results

In this section, we present the results for the SLR process and for the research questions as well. In Table II, we present the list of included studies. In Figure 2, we present the amount of papers per year. In Figure 3, we show the distribution of papers per type of publication channel. In Figure 4, we show the percentage of the included papers per type of agile method studied. In Figure 5, we show the aggregated results of the quality assessment.

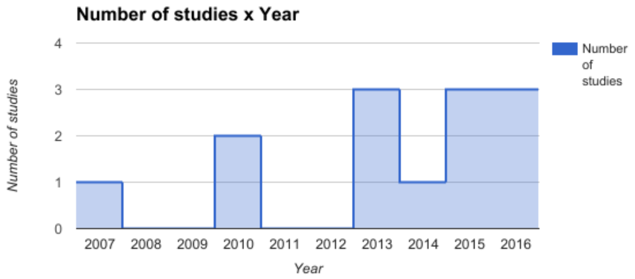

Fig. 2: Number of studies x year.

In Figure 6, we present the number of factors identified in the review (**RQ1**). We identified eight factors in our study, i.e., business value (P1, P2, P4, P5, P6, P7, P8, P9, P10, P12, P13), effort/cost (P1, P2, P4, P5, P8, P9, P13), dependency (P1, P2, P3, P4, P6, P9, P11, P13), risk (P1, P2, P9), volatility (P1, P2), technical debt (P2), human resources (P1), and schedule (P1). Additionally, we show the information related to the measurement of the factors in Table III (**RQ2**). By analyzing Table III, it is possible to conclude that effort, dependency, business value, and risk had measurement procedures presented in the works. On the other hand, the measure of four factors – volatility (P1 and P2), technical debt (P4), human resources (P1), and schedule (P1) – were not explained in the papers.

In table IV, we show the techniques to order the backlog identified in our review (**RQ3**) and the evidences that show

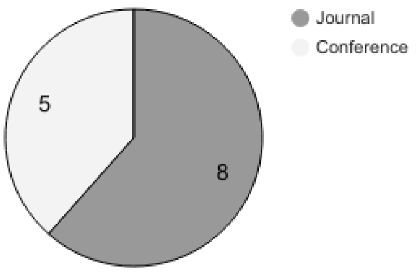

Fig. 3: Number of studies per type of publication channel.

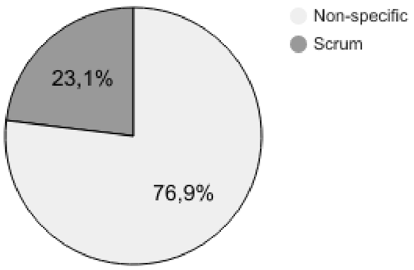

Fig. 4: Percentage of agile methods investigated in the studies.

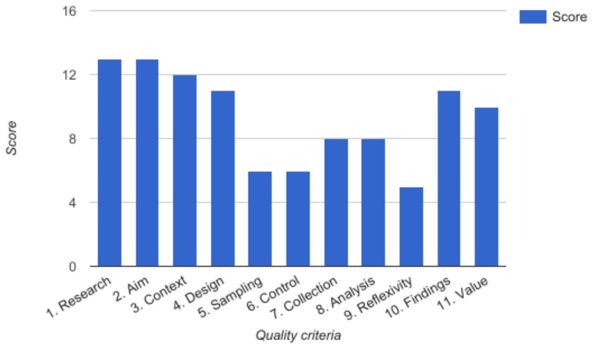

Fig. 5: Score x quality criteria.

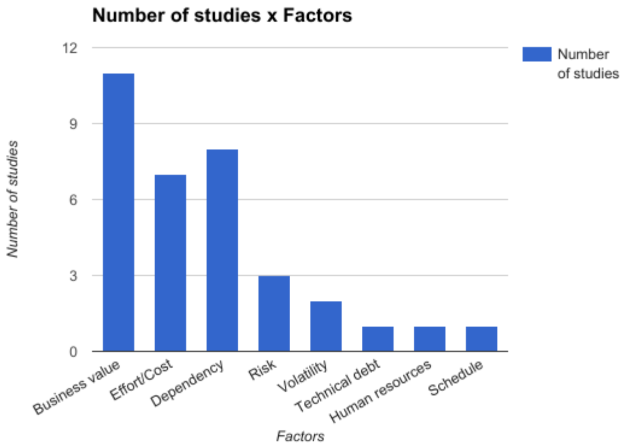

Fig. 6: Number of studies x factors.

TABLE II: Overview of the selected studies.

Paper Number	Authors	Year	Title	Publication Channel
P1	Rami Hasan AL-Taani and Rozilawati Razali	2013	Prioritizing Requirements in Agile Development: A Conceptual Framework	International Conference on Electrical Engineering and Informatics
P2	Maya Daneva, Egbert van der Veen, Chintan Amrit, Smita Ghaisas, Klaas Sikkel, Ramesh Kumar, Nirav Ajmeri, Uday Ramteerthkar and Roel Wieringa	2013	Agile requirements prioritization in large-scale outsourced system projects: An empirical study	Journal of Systems and Software
P3	Marina Trkman, Jan Mendling and Marjan Krisper	2016	Using business process models to better understand the dependencies among user stories	Information and Software Technology
P4	Michel dos Santos Soares, Jos Vrancken and Alexander Verbraeck	2010	User requirements modeling and analysis of software-intensive systems	Journal of Systems and Software
P5	Jos M. Chaves-Gonzlez, Miguel A. Prez-Toledano and Amparo Navasa	2015	Software requirement optimization using a multiobjective swarm intelligence evolutionary algorithm	Knowledge-Based Systems
P6	David P. Harvie and Arvin Agah	2017	Targeted Scrum: Applying Mission Command to Agile Software Development	IEEE Transactions on Sofware Engineering
P7	Julian M. Bass	2015	How product owner teams scale agile methods to large distributed enterprises	Empirical Software Engineering
P8	Thomas Michael Fehlmann and Eberhard Kranich	2014	Early software project estimation the six sigma way	XP 2014
P9	Weam M. Farid and Frank J. Mitropoulos	2013	NORPLAN: Non-functional requirements planning for agile processes	IEEE SoutheastCon
P10	Helena Holmstrm Olsson and Jan Bosch	2015	Towards continuous validation of customer value	XP 2015
P11	Arturo Gomez, Gema Rueda and Pedro P. Alarcn	2010	A systematic and lightweight method to identify dependencies between user stories	XP 2010
P12	Balasubramaniam Ramesh, Lan Cao and Richard Baskerville	2007	Agile requirements engineering practices and challenges: an empirical study	Information Systems Journal
P13	Jan Vlietland, Rini van Solingen and Hans van Vliet	2016	Aligning codependent Scrum teams to enable fast business value delivery: A governance framework and set of intervention actions	Journal of Systems and Software

TABLE III: Data related to the factors measurement.

Factor	Measure	Study
Effort	Person hours to complete a delivery story. Vendor uses empirical data about the productivity of their teams, if feasible.	P2
	Story points.	P8
	COSMIC function points.	P8
	PERT-based User Story Points scheme.	P9
Dependency	The dependence between user stories is identified by associating user stories with business process model activities through a method called BuPUS (Business Process User Story method).	P3
	Precedence, combination, exclusion or modification.	P5
	Dependency on key and dependency on service.	P11
Business value	A given priority level and a weight assigned to the client.	P5
	Contribution of a story card to one of the business drives on a weighted scale.	P8
	Quantitative and qualitative customer feedback techniques.	P10
Risk	Project management and requirements quality metrics, estimate and dependency risk are used to calculate the risk score.	P9

the used techniques are efficient (**RQ4**). By analyzing Table IV, it is possible to identify that only nine studies (P3, P4 P5, P6, P8, P9, P10, P11, and P13) specified the technique used to order the product backlog, the remaining three works did not give details about the used technique (P1, P2, and P12). Additionally, two studies did not show the efficiency of the used technique (P10 and P11).

V. Discussion

In this section, we discuss the results about the four research questions (see Section III-A), which explore the ordering of product backlog in ASD projects.

The first paper was published in 2008, i.e., research on this area is relatively recent. By analyzing Figure 5, we concluded that most of the papers present important findings and research value for the area, but the average quality of studies is low, considering sampling, selection of control groups, and reflexivity, which indicates that works on this topic need more empirical rigor.

We identified 13 relevant papers (see Table II), and eight different factors (see Figure 6), as follows:

Effort/Cost. P2 defines the effort based on the number of hours a person need to complete a delivery store. P8 uses the story points and the COSMIC function points to measure the effort. P9 calculates the effort based on a PERT user story points scheme. P1, P2, P5, and P13 do not give details on how the effort was measured in their work.

Dependency. P3 defines dependency between user stories by associating user stories with business process model activities through a method called BuPUS. On the other hand, P5 did not measure the dependency, but classifies it as: precedence, combination, exclusion or modification., in which the type of classification influences the resulting ordering. P11 classified the dependency into two types: dependency on key and dependency on service. P1, P2, P3, P6, P9, P11 and P13 do not give details on how the dependency was measured in their work.

Business value. P5 defines business value based on a given priority level and a weight assigned to the client. In P8, the business value is measured considering the contribution of

TABLE IV: Data related to the techniques.

Technique (RQ3)	Evidences of efficiency (RQ4)	Study
The execution orders are managed by flow and gateways elements.	According to an experiment performed with 127 undergraduate students, their solution increase the execution order and integration dependencies among user stories by the giving context.	P3
Use SysML diagrams and constructions which helps in better organizing requirements to assist on the prioritization of requirements.	They apply the solution to a list of requirements extracted from the literature and evaluate it given the IEEE Recommended Practices for Software Requirements Specification, showing better representation.	P4
Multi-objective search-based approach based on the artificial bee colony (ABC) algorithm	The authors present the results using different quality indicators and compare them with the results of other approaches published in the literature. They argue that their proposed solution is the algorithm which gives the best results regarding this multi-objective indicator for every problem instance (8 instances compared).	P5
Modifications to agile software development based on inspirations from mission command	The proposed solution was tested during a semester-long agile software engineering course designed for both graduate and upper level undergraduate students in Computer Science and Computer Engineering at the University of Kansas. They concluded that it improved planning and prioritization of requirements and developing the overall project architecture.	P6
Buglione-Trudel matrix tool	The solution was applied to a sample set of requirements.	P8
Risk-driven algorithm to prioritize and plan an improved requirements implementation sequence	The solution was applied to the requirements of the European Union procurement system just to show the different results that the algorithms calculated.	P9
Qualitative/quantitative Customer-driven Development (QCD) validation cycle	Unspecified	P10
Dependencies Identification Method	Unspecified	P11
Product Owner Group (POG), which discusses and decides about the priority of each feature on the feature backlog.The POG is headed by the Epic Product Owner.	The intervention actions (IA) resulted in an average delivery time reduction from 29 days to 10 days.	P13

a story card to one of the business drives on a weighted scale. On the other hand, P10 defines business value based on quantitative and qualitative customer feedback techniques. P1, P2, P4, P6, P7, P9, P12 and P13 do not give details on how the business value was measured in their work.

Risk. in P9, risk is defined as score based on project management and requirements quality metrics, estimate and dependency risk. P2 and P9 did not give details on how the risk was measured in their work.

Volatility. P1 and P2 uses the requirements volatility as a factor to order the product backlog, but they do not give details on how it was measured in their work.

Technical debt. In P2, technical debt is used as a factor to order the product backlog, but there is no detail about its measurement. Although the authors report that it implies the amount of architecture-redesign related work that accumulates over a period of time.

Human resources. P1 uses the human resource as a factor to order the product backlog, but there is no detail about its measurement.

Therefore, business value (11 studies), dependency (8 studies) and effort (7 studies) are the most reported factors, which shows the relevance of considering this factors in the product backlog ordering. Additionally, in the literature, Usman et al. [14] conducted an SLR on the state of the practice on effort estimation in ASD. Furthermore, Racheva et al. [9] conducted an SLR on how business value is created by agile projects.

Moreover, 9 studies give details about the techniques used in the product backlog ordering (see Table IV). None of them used the same approach. In P3, the execution orders are managed by flow and gateways elements. The authors argue that their method improve the understanding about the execution order and integration dependencies among user stories, based on an experiment performed with 127 undergraduate students. In P4, they used a SysML Requirements diagram, which details requirements relationships, and hence, aids in requirements prioritization. They apply the solution to a list of requirements extracted from the literature and evaluate it, given the IEEE Recommended Practices for Software Requirements Specification. P5 uses a multi-objective search-based approach based on the artificial bee colony (ABC) algorithm. The authors compare their approach to other ones published in literature and argue that their proposed solution reached the best results, considering 8 instances compared.

P6 uses Product Backlog grooming dialogues, which are formalized through Lines of Effort (LOEs) as visual representations of the software client's priorities and desire end state. The proposed solution was tested during a semester-long agile software engineering course designed for both graduate and upper level undergraduate students at the University of Kansas. They concluded that it improved planning and prioritization of requirements and developing the overall project architecture. In P8, the technique used was the Buglione-Trudel matrix tool, which provides agile teams with immediate feedback whether their priorities meet customer needs. The solution was applied to a set of requirements. P9 uses a risk-driven algorithm to prioritize and plan an improved requirement implementation sequence. The approach was validated through visual simulation and a case study, which indicates that the use of the proposed solution results in a decrease on the overall duration of the implementation.

P10 uses quantitative and qualitative customer feedback techniques to accomplish the task of product backlog ordering. In this study, no evidences of the technique efficiency are shown. In P11, a dependency identification method is used. In this study, no evidences of the technique efficiency are shown. P13 presents a governance framework and set of intervention actions. One of the intervention action is the Product Owner Group (POG), which discusses and decides about the priority of each feature on the feature backlog. As a result of the interventions, the average delivery time was reducted from 29 days to 10 days.

Therefore, we can conclude that there are many approaches to order the product backlog, but there is no consensus about which one achieves the best results. Additionally, in some works the efficiency of the techniques was not evaluated. Yet, in most cases the evaluations were carried out in academic environment, which may not faithfully represent real industry environment.

VI. Threats to Validity

As well as in all SLR studies, a common threat to validity regards to the covering of all relevant studies. Therefore, to mitigate this problem, we formulated a comprehensive string, which covered keywords and their synonyms and applied it to different database searches to cover main Software Engineering conferences and journals. Another threat faced regards to researchers' opinion, which can influence in the results of the study. To mitigate this problem, each paper was evaluated by two reviewers, which should agree on the inclusion or exclusion of the paper.

VII. Conclusion and Future Work

In this paper, we conducted a systematic literature review with focus on the ordering of product backlog in agile software development projects to identify and analyze the main factors and techniques used in this process. To reach this goal, we applied a database search approach to identify the most relevant studies in the topic of study. The primary search fetched 1814 results, among them, 258 duplicated, which resulted in 1556 papers. After analyzing basic criteria, 35 papers were selected, and finally, after advanced criteria analysis (i.e., quality assessment and data extraction accomplished), only 13 papers were included in the study. At the end of the process, data from these papers were analyzed to answer research questions formulated.

We identified 8 different factors. Among them, business value (11 studies), dependency (8 studies) and effort (7 studies) were the most reported ones, which indicates that they are valuable factors to be considered in the product backlog ordering process. Additionally, we concluded there is no consensus about which technique to be applied.

Results of this study can be used as baseline by practitioners to start new works, since it presents an overview of the studies in the literature about the ordering of backlog in ASD. Furthermore, based on our results, we concluded this topic needs further investigation with the conduction of empirical studies to assess the results of applying this practice in real environments.

For future work, we intend to complement the database search with forward and backward snowballing approaches [16], by using the 13 included papers in this study as the seed set of the snowballing.

Acknowledgment

The authors would like to thank CAPES for supporting this work.

References

[1] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin. A systematic literature review of software requirements prioritization research. *Information and Software Technology*, 56(6):568 – 585, 2014.

[2] N. B. Ali, K. Petersen, and C. Wohlin. A systematic literature review on the industrial use of software process simulation. *Journal of Systems and Software*, 97:65 – 85, 2014.

[3] W. Alsaqaf, M. Daneva, and R. Wieringa. *Quality Requirements in Large-Scale Distributed Agile Projects – A Systematic Literature Review*, pages 219–234. Springer International Publishing, Cham, 2017.

[4] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4):571 – 583, 2007. Software Performance5th International Workshop on Software and Performance.

[5] T. Dyb and T. Dingsyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(910):833 – 859, 2008.

[6] V. T. Heikkil, D. Damian, C. Lassenius, and M. Paasivaara. A mapping study on requirements engineering in agile software development. In *2015 41st Euromicro Conference on Software Engineering and Advanced Applications*, pages 199–207, Aug 2015.

[7] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband. A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, 51, Part B:915 – 929, 2015. Computing for Human Learning, Behaviour and Collaboration in the Social and Mobile Networks Era.

[8] B. Kitchenham and S. Charters. Guidelines for Performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE-2007-01, Schoolf of Computer Science and Mathematics, Keele University, 2007.

[9] Z. Racheva, M. Daneva, and K. Sikkel. *Value Creation by Agile Projects: Methodology or Mystery?*, pages 141–155. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[10] E.-M. Schon, J. Thomaschewski, and M. J. Escalona. Agile requirements engineering: A systematic literature review. *Computer Standards & Interfaces*, 49:79 – 91, 2017.

[11] M. Shaw. Writing good software engineering research papers: Minitutorial. In *Proceedings of the 25th International Conference on Software Engineering*, ICSE '03, pages 726–736, Washington, DC, USA, 2003. IEEE Computer Society.

[12] M. Staples and M. Niazi. Experiences using systematic review guidelines. *Journal of Systems and Software*, 80(9):1425 – 1437, 2007. Evaluation and Assessment in Software EngineeringEASE06.

[13] P. Tonella, M. Torchiano, B. Du Bois, and T. Systä. Empirical studies in reverse engineering: state of the art and future trends. *Empirical Software Engineering*, 12(5):551–571, 2007.

[14] M. Usman, E. Mendes, and J. Börstler. Effort estimation in agile software development: A survey on the state of the practice. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, EASE '15, pages 12:1–12:10, New York, NY, USA, 2015. ACM.

[15] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. *Requir. Eng.*, 11(1):102–107, Dec. 2005.

[16] C. Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, pages 38:1–38:10, New York, NY, USA, 2014. ACM.