Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 157 (2019) 274–281

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), 12–13 September 2019

Systematic Literature Review on Decision-Making of Requirement Engineering from Agile Software Development

Reginald Putra Ghozali[a], Herry Saputra[a], M. Apriadin Nuriawan[a], Suharjito[a], Ditdit Nugeraha Utama[a,*], Ariadi Nugroho[a]

[a]*Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480*

Abstract

This paper presents a systematic literature review of agile software development at decision making method for requirement engineering. Presently, agile software development method is operated to cope with requirements that changes dynamically. This study seeks to find out and discuss what types of method that have been exploited for decision making on managing feasible requirements and challenges of decision making in agile software development. Papers reviewed in this study are published from 2017 to present. Resulting 8 papers that have been identified of presenting decision making methods. Using these papers, 11 methods and 7 challenges of decision making identified. This study contributes a review of requirement management and engineering by providing decision making methods on agile software development and the challenges of decision making for requirement engineering.

Keywords: Systematic Literature Review; Agile Software Development; Decision Making; Requirement Engineering;

* Corresponding author. Tel.: +62-812-8961-4291.
E-mail address: ditdit.utama@binus.edu

10.1016/j.procs.2019.08.167

1. Introduction

In every project, requirements definition is important phase that significantly affects whole project processes and results. Failure to satisfy all these requirements might lead to unhappy customers or even lawsuits [1]. Carelessly accept all needs and wishes of client may lead to problems that caused project to be delayed or cancelled [2]. Requirements not only come from clients, but are able to be from environments, or evaluation of project prototype's performance that came from unsatisfied stakeholders [3]. These phenomena cause the project stakeholders must have good decision-making to satisfy project's goal.

In software development project, many factors are dependent to decisions from project stakeholders. Most of them are capable to change including the requirements. The nature of decision-making is dependent to person's experience, knowledge, capabilities, and cultural background [3]. These factors lead to challenge that the project's members are collective of different people that may lead into different decision-making results. Besides of project's member experiences, there are external factors that also affect decision-making. Either a trend, or other environmental elements.

One of the most recent research [4] used systematic literature review (SLR) to find application of agile techniques for project management in different software projects. Most of the research have identified and portrayed methods of managing project in agile environment. Agile management purpose is to ensure project members to develop software that work well in continuous way [4]. The research concluded that agile project's performance is affected by team's freedom, authority, and capability to satisfy customers' needs [4]. However, the research only clarified what kind of requirements that are going to be managed in agile project.

The purposes of this research are to find challenges of decision-making for requirements and every approaches of decision-making for managing feasibility of user's requirements that have been functioned and suggested. A good requirement needs to be engineered by good decision-making from the stakeholders. And if these requirements defined clearly, it is going to produce the optimum performance of agile software development [3]. These purposes are achieved by executing a SLR on past research papers that discuss about decision-making on project's requirements.

2. Research Methodology

The research conducted in this paper is based on three reasons for performing an SLR. First is to aggregate and unify knowledge from previous researches concerning the related topic. Second is to identify weaknesses from previous researches. And third is to support for finding new research topic by providing base information required [5]. These reasons are why SLR capable to fulfill needs to reach multiple purposes while maintain a strong proving ground.

2.1. Research Questions

The SLR was conducted to answer two central research questions as a guidance to perform the study. Such questions are:

1. RQ1: What kind of approaches of decision-making for managing feasibility of user's requirements?
2. RQ2: What challenges of decision-making for requirements in agile development?

RQ1 focused on finding best approach of decision-making for managing user's requirements that already done by other researchers that closely related to our topic. This involved analyzing and comparing every approach. In actual fact, the technology advance and business needs trigger the user's requirement to evolve every day [6]. It directly affects the achievement of the software development project. Thus the proper method to decide the requirement is definitely required, particularly regarding information and communication quality [7]. While RQ2 involved finding and point out most challenges that faced by agile development projects that related to decision-making for requirements. Surely, this research question mentions strongly that the decision making in managing the user's requirement is practically desirable. As, the requirement evolves daily and progressively [6].

2.2. The Search Process

Search process involved standard index in science related literature on online databases. All papers included were within January 2017 to present. Online digital libraries included are: IEEE Xplore, ACM, and Springer Link. The search string format used in the search criteria was:

- (agile OR scrum OR kanban OR "extreme programming" OR lean)
- (decision making) AND (requirement engineering)

The first search string used in the "find at least one in contexts or titles" from advanced search of digital libraries. It used to find all papers that related to agile development. Second search string used in "find exactly same" from advanced search. This search string used for filtering every research paper that are not related with decision-making and requirement engineering.

2.3. Study Selection

To concentrate the center of the focus, the selection of studies was minimized by using a set of inclusion and exclusion criteria [8]. We tried to include as many research papers as possible to compare every feasible approaches that have been used and carefully avoid research papers that not related to software development and agile development. The inclusion criteria is: Paper related of agile development about decision making on requirements from 2017 to present, based on case studies and/or experience reports and/or researches. Three years are common review operated to see the trend usually operated via several researchers, e.g. [9] and [10].

The exclusion criteria are:

1. Paper focused on agile development about decision making on requirements but not on software project.
2. Studies that did not discuss decision making for requirement engineering in agile development.
3. Full book paper or non-English paper.
4. Paper introducing tools without specific function described.

Table 1. Quality Assessment Questions.

Number	Question
1	Is this a research paper?
2	Is the paper covered all related topics?
3	Are the aims defined clearly?
4	Are the findings defined clearly?
5	Is the paper provide examples /illustrations/ implementations of methods used?

The selection started with searching directly into digital libraries by using only the search strings without modifying search criteria. Then, apply inclusion and exclusion criteria to reduce research papers founded. These papers then filtered by analysing based on title and abstract to find all related papers that capable to answer research questions. Finally, remaining papers filtered based on full text analysis and quality assessment [11]. The quality assessment questions are shown in table 1. While the full text analysis is using thematic descriptive analysis approach to find out if the paper fulfilled all related topics and then carefully read full text of the paper.

3. Result of Study

Whole results of selection are shown in Fig. 1. At first, we committed search on the digital libraries using only the search string. Which returned 491 papers from ACM digital library, 10717 papers from IEEE Xplore digital library, and 54,714 papers from Springer Link digital library. Resulting total of 65,922 papers of first stage of searching. After that, we applied the inclusion criteria and exclusion criteria, resulting 1,358 papers from ACM, 101 paper from IEEE Xplore, and 16 papers from Springer Link. This second stage provide us with 1,475 papers. Third stage of searching

is selection based on title and abstract, resulting 20 papers. Lastly, we filtered our search based on full text analysis and quality assessment resulting 8 papers that capable to answer our research questions.

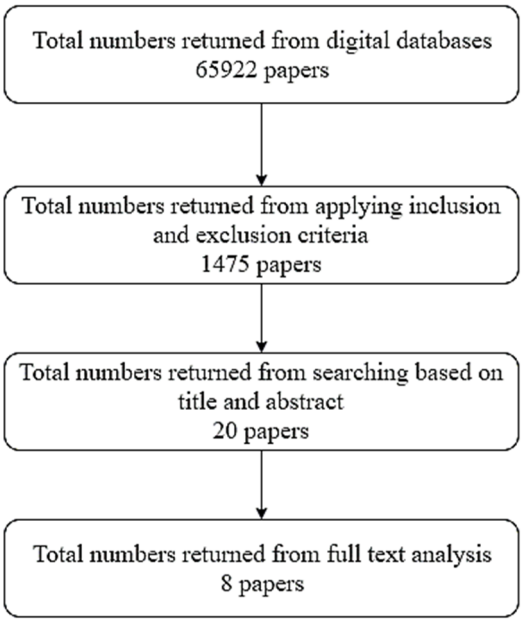

Fig. 1. Flow of Paper Selection Process

Table 2. List of Selected Papers.

Number	Publication Year	Paper Type	Topics	Method(s) Used
[S1]	2018	Conference	Requirement Allocation	Mixed Integer Programming Model
[S2]	2018	Conference	Visualization for Decision Making	Goal Model
[S3]	2018	Conference	Decision Knowledge	Decision Knowledge
[S4]	2018	Conference	Requirement Design Model	Change Vector
[S5]	2017	Conference	Requirement Elicitation & Prioritization	Cumulative Voting, Numerical Assignment, analytical hierarchy process, MoSCoW
[S6]	2018	Conference	Requirement Prioritization	Flowchart Model
[S7]	2017	Conference	Requirement Prioritization	Cumulative Voting, Weighted Sum, Planned Poker
[S8]	2017	Conference	Requirement Decomposition	Extreme Programming, Scrum, Scrum with Kanban

Each paper analyzed to fulfill all research questions mentioned before. 8 papers that were selected from previous process. The papers listed in Table 2. These papers are listed with their proposed method for decision-making of requirement engineering. Mostly topics that we founded from these papers are about requirement prioritization. While some of them are about visualization of requirement design.

From paper [S1], it described about next release problem. Next release problem defined in this paper is about what will be implemented in the next release as the decision affected by several factors. The Agile method using SCRUM with mixed integer programming model [12] for decision making of requirements. The decision making is also used for other categories like business and legal problem. These requirements are valued based from business value, minimum and maximum team members needed, cost, and time needed for finishing requirements. Results from these computations will be the base for taking decision. This paper focused to fulfill customer satisfaction, reduce time development, increase buffer time to consider new releases, and reduce cost on development. Case study used in this

paper with 3 companies and 9 different teams. Research used to compare decision making with manual computation and with modelling tools. The criteria used in the research are to find better requirements for future release with more time, maximization of customers' satisfaction, and optimum cost reduction. Research showed that manual results were worse than modelling tools [12].

Paper [S2] is about decision making using goal models, where the model gave clear representation of different alternatives that can be met by high-level objectives. These alternatives have each of their own qualities that might lead to confusion for reasoning. This paper presented the model as the effective method to reasoning decisions for requirement engineering [13]. The model proposed is represented with bar and pie chart, and traditional tree-map. These difference visualization lead to different performance. And these representations tested with university students. The test asked the participants to do work on daily occurrences to measure accuracy, efficiency, and confidence. Bar and pie chart gave better performance to define hard-goal, which is accurately defined to achieved, and soft-goal, that inaccurately defined that become the decision criteria. Each of hard-goal gave different effect to each criteria defined [13].

Paper [S3] is about decision knowledge. Decision knowledge is about developer's knowledge about decisions and their justification. The decision knowledge wanted to be captured explicitly from daily programming tasks and become assets for future reference [14]. This action is needed to be performed repeatedly to ensure the information is not outdated, incomplete, or hard to be accessed. The decision knowledge included a knowledge of the systems (code, design, and test cases) that called system knowledge, and the knowledge of developments and software progress called project knowledge [14]. The paper defined the knowledge meta-model with decision documented model. Then using scenario that fitted with the knowledge needed. The decision knowledge also has triggers that defined in this paper to capture and use this decision knowledge. They are packaging distributed decision knowledge, making tacit decision explicit, and considering consistency between decisions [14].

Paper [S4] is about change vector to visualize changes that will be happen from selected design. The practice is conducted iteratively to evaluate different design options. Each design will be given weight based on dimensions that will be fulfilled by the design. The paper gave illustration from selecting requirement design based on quality and cost [15]. The paper described how to model these changes into a vector, then analyze the nonfunctional requirements that will be affected by the design, while also giving the alternatives of changes for different time.

Paper [S5] is a review paper about impact and challenges from requirements elicitation and prioritization from agile SCRUM. The proposed model used cumulative voting that differentiate requirement based on limited value given into stakeholders. Then results from each stakeholder were aggregated into single value for said requirements. Second step is using numerical assignment that grouped the requirements based on influencing factors. These groups were separated based on distributed points on the criteria. Third step using analytical hierarchy process that gave rank based on importance if compared with other requirement groups to produce 10 top requirements. Lastly, the requirements filtered using MoSCoW (Must have Should have Could have Won't have) method that based on human opinions that based from experience, desire, costs, etc. to create fixed requirements list [16]. The model validated using iThink software to simulate the flow of the proposed model. The model expected to increase requirements' quality [16].

Paper [S6] is about user requirements prioritization in agile project. The priority is based from business and process flow. From the user stories collected, these stories converted into UML diagram or flowchart. Then assign these priorities based on process flow where the topmost process become most important if compared with below process. The priority also aggregated by priority from product owner based on business value. These two priorities aggregated by giving appropriate weight. This last result converted into product backlog with their prioritization [17]. The model explained using library management system as example. Where user stories such as registration or searching book for users, are defined first. Then the given stories converted into flowchart. Next, the priorities defined by product owner that based on business value are defined. The final priorities are defined by aggregating these priorities rank [17].

Paper [S7] is about priority requirement. The model used based on user stories rank that produced from cumulative voting method. Then all the requirements weighted with weighted sum method. To find priority to fulfill the requirements. To find out the effort and complexity of requirements, a method called planned poker was used. Where a set of prime number cards are given to each stakeholder, these stakeholders gave their vote to define their doubts based on the prime number used. If all the stakeholders give same number, that number become value of the user-story, if the number different, these stakeholders gave their opinions why they gave these numbers, and then after discussion, the scoring is repeated. This process will be repeated for each requirement [18]. Case study is used to show

feasibility from this proposed method. The study involved about university management system from Lovely Professional University students that provide information that helpful for faculties and students. Result shown that students prioritized login to the system as top priority, and left lecturer's feedback as lowest priority [18].

Paper [S8] is about research to decompose requirement used by Scrum, Extreme Programming, and Scrum with Kanban. Requirements are decomposed based on input options, data types, operations of the requirements (CRUD), complexity, workflow, etc. The paper written an empirical study to compare requirement decomposition with controlled experiment of four groups of master students. The experiment used project idea used for contest for entrepreneurs with project design created by designer that did not have any experience in software development. The output resulted that extreme programming decomposed more stories than others, then scrum with Kanban, with following the one using scrum method. The lowest performance was done by unstructured process team. For the development effort, it was inverse if compared with the amount of story decomposition, resulting lowest effort for extreme programming while highest effort for unstructured process team [19].

4. Discussion of Research Questions

RQ1. What kind of approaches of decision-making for managing feasibility of user's requirements?

Based from papers reviewed in data extraction, we found different types of method used by different people. These methods are described in table 3. Different kind of methods described by these papers have their own focus points to improve requirements' quality. Paper [S1] used mixed integer programming model that work as top-down approach while second paper [S2] using goal model that work as bottom-up approach. The decision knowledge used in paper [S3] defined about method to generate knowledge for decision-making. The design selection proposed by paper [S4] is interesting to track requirements history. Paper [S5] offer combination of decision-making methods of cumulative voting, numerical assignment, analytical hierarchy process, and MoSCoW. The aim is to produce good quality of requirements with clear definition. Another paper [S7] also used combination of cumulative voting, weighted sum, and planned poker to rank the requirement priorities. Paper [S6] also suggested method to ranking requirements using flowchart. Final paper researched [S8] is defining requirement decompositions of different agile methods.

Table 3. Approaches of Decision-Making for Managing Feasibility of User's Requirements.

Number	Model	Description
1	Mixed Integer Programming Model	A method where mixed weighted values are used for weighting the requirements. These values are calculated from various aspects in requirements, i.e. costs, team members needed, time, or business value. As the base for taking decision in requirement allocation.
2	Goal Model	Model for summarize the high-level objectives of the development that can be met based on the criteria selected.
3	Decision Knowledge	Method to extract the knowledge of developer to be applied in decision making. The knowledge also included justification to support the validity of the knowledge. Justification can be either experience or personal opinions.
4	Change Vector	Method to visualize the outcome of selecting certain requirement design that will be executed for development. This method also described how the requirement design affect certain defined criteria.
5	Cumulative Voting	Requirement ranking method by distributing virtual values to stakeholders. The stakeholders distributed these values to each requirement. The result of distributed value will be aggregated with other stakeholders selection for each requirement. Then ranked based on highest score given to the requirements.
6	Numerical Assignment	Grouping method by assigning numerical value of certain criteria to the requirement, then group them by finding similar highest value of the given criteria.
7	Analytical Hierarchy Process	Ranking method that compared one requirement with other requirements then gave rank based on importance in the development.
8	MoSCow	Must have Should have Could have Won't have, a method to filter requirements based on their importance. Mostly filtered based on human experience and opinions.
9	Flowchart Model	Model used to rank priority of requirements based on topmost tasks that will be executed by the software.
10	Weighted Sum	Method to find priority to meet requirements based on aggregated weights of tasks for each requirement.
11	Planned Poker	Method to give rank for requirements by using set of numbers that will be selected by stakeholders. When every stakeholder gave same value to the requirements, that value will be assigned for that requirement, if not, the stakeholders hold short discussion to create same view points for the requirement and execute the voting again.

Discussing about decision making, there is no right decision; academically we only talk about the objective decision. All methods mentioned in Table 3 were already operated well in specific decision problem area. However, we proposed that the researchers should enlarge interconnected parameters more (not only number, but also quality of parameter); as the quality of decision selected alternatively is totally depends on parameters included. Furthermore, the decision model for answering the problem, particularly the problem of user requirement managing in one project, the model verification and validation should be performed technically.

RQ2. What challenges of decision-making for requirements in agile development?

Challenges that founded from these papers are different based on the topic covered in each paper. We summarized most general challenges founded in each research, and also included few papers that already filtered from full text analysis but relevant to this research question. All of the challenges are written in table 4. Mostly all papers we researched have problem in discovering whole important requirements needed at the beginning of project [S5]. This is usually caused by lack of experiences or capabilities to define requirements. However, both developers and team members might be having minimum motive to share their knowledge [S3]. On the other side, market trends that lead the changes can be unpredictable. The trends capable to change quickly, causing customers' needs must be fulfilled as fast as possible [S1]. Few other challenges that described by other papers are different experiences and views from stakeholders that lead into conflict when giving decision. This will reduce development time if the decision is not appointed. Which might cause loss of quality and integrity of developers [**16**]. The knowledge of project's big picture defined by customers or stakeholders that will be passed to developers might be wrongly understood if defined poorly. This caused developers to be confused about what they will develop [**16**]. One of the confusion sources is unpredictable customer involvement. As the developers at usual case are not known by the customers, and vice versa [20]. The effort will be similarly unpredictable. Stakeholders' authority might become obstacle to define requirements. This problem called High Paid Person Opinion (HiPPO) that may be restrict requirements or even change requirements on the development because the stakeholder personal opinion [21].

Table 4. List of Challenges of Decision-Making for Requirements in Agile Development.

Number	Challenges
1	Difficulty to discover whole requirements needed in the early phase of development.
2	Minimum motive from developers / team members to share their knowledge for requirement engineering.
3	Market trends that changes quickly.
4	Different experiences and view points from each stakeholder that always lead to debate, causing reduction to development time.
5	Knowledge of the big picture from project goal that can be wrongly understood from stakeholders to development teams, or to customers that need considerable efforts.
6	Customer involvement that may be gave unpredictable efforts done by teams.
7	High Paid Person Opinion (HiPPO) that may be restrict requirements or even change requirements on the development.

5. Limitations

The minimum amount of papers that reviewed in this paper might cause validity of the findings to be questioned. We assumed that short gap of published paper year (2017 to present) will show most state-of-the-art methods or models for decision making in requirement engineering. However, in reality this caused the minimum amount of relevant papers founded and filtered. This also caused by the strict quality assessment that reject most papers that did not fulfill all the assessment questions.

6. Conclusion and Future Works

We conducted systematic literature review on decision making for requirement engineering in agile software development. For the response to the research question RQ1, 11 methods or models are used to engineering requirements. These methods or models can be combined according to necessity information needed to define

requirements. There are 7 challenges that found out from papers reviewed. Future research could be held to improve validity of these findings by including more digital libraries or expand the inclusion criteria.

References

1. Ebert C. 50 Years of Software Engineering: Progress and Perils. IEEE Software. 2018; 35(5): p. 94-101.
2. Anu V, Hu W, Carver JC, Walia GS, Bradshaw G. Development of a human error taxonomy for software requirements: A systematic literature review. Information and Software Technology. 2018; 103: p. 112-124.
3. Maalej W, Nayebi M, Johann T, Ruhe G. Toward data-driven requirements engineering. IEEE Software. 2016; 33(1): p. 48-54.
4. Lappi T, Karvonen T, Lwakatare LE, Aaltonen K, Kuvaja P. Toward an Improved Understanding of Agile Project Governance: A Systematic Literature Review. Project Management Journal. 2018; 49(6): p. 39-63.
5. Kupiainen E, Mäntylä MV, Itkonen J. Using metrics in Agile and Lean Software Development–A systematic literature review of industrial studies. Information and Software Technology. 2015; 62: p. 143-163.
6. Al-Ta'ani RH, Razali R. Prioritizing Requirements in Agile Development: A Conceptual Framework. Procedia Technology. 2013; 11: p. 733-739.
7. Hess A, Diebold P, Seyff N. Understanding information needs of agile teams to improve requirements communication. Journal of Industrial Information Integration. 2019; 14: p. 3-15.
8. Hoda R, Salleh N, Grundy J, Tee HM. Systematic literature reviews in agile software development: A tertiary study. Information and Software Technology. 2017; 85: p. 60-70.
9. Pezeshki Z, Soleimani A, Darabi A. Application of BEM and using BIM database for BEM: A review. Journal of Building Engineering. 2019; 23: p. 1-17.
10. Rijal P, Uprety D. W382 Material Mortality: A 3 Years Retrospective Review from Eastern Nepal. Obstetrics. 2012; 119(3): p. S828.
11. Schön EM, Thomaschewski J, Escalona MJ. Agile Requirements Engineering: A systematic literature review. Computer Standards & Interfaces. 2017; 49: p. 79-91.
12. de Melo França VJ, Balancieri R, Leal GCL, Rouiller AC. Mixed Integer Programming helping Requirements Allocation for the NRP in SCRUM Teams. In Proceedings of the 17th Brazilian Symposium on Software Quality; 2018; Curitiba, Brazil: ACM. p. 279-286.
13. Liaskos S, Dundjerovic T, & Gabriel G. Comparing alternative goal model visualizations for decision making: an exploratory experiment. In Proceedings of the 33rd Annual ACM Symposium on Applied Computing; 2018; Pau, France: ACM. p. 1272-1281.
14. Kleebaum A, Johanssen JO, Paech B, Alkadhi R, Bruegge B. Decision knowledge triggers in continuous software engineering. In Proceedings of the 4th International Workshop on Rapid Continuous Software Engineering; 2018; Gothenburg, Sweden: ACM. p. 23-26.
15. Tharayil R. Change Vector Tracking in Emergent Design. In Proceedings of the 11th Innovations in Software Engineering Conference; 2018; Hyderabad, India: ACM.
16. Asghar AR, Tabassum A, Bhatti SN, Jadi AM. Impact and Challenges of Requirements Elicitation. In 2017 International Conference on Communication Technologies (ComTech); 2017; Rawalpindi, Pakistan: IEEE. p. 50-55.
17. Sachdeva S, Arya A, Paygude P, Chaudhary S, Idate S. Prioritizing User Requirements for Agile Software Development. In 2018 International Conference On Advances in Communication and Computing Technology (ICACCT); 2018; Sangamner, India: IEEE. p. 495-498.
18. Sheemar H, Kour G. Enhancing User-Stories Prioritization Process in Agile Environment. In 2017 International Conference on Innovations in Control, Communication and Information Systems (ICICCI); 2017; Greater Noida, India: IEEE. p. 1-6.
19. Taibi D, Lenarduzzi V, Janes A, Liukkunen K, Ahmad MO. Comparing Requirements Decomposition Within the Scrum, Scrum with Kanban, XP, and Banana Development Processes. In International Conference on Agile Software Development; 2017. p. 68-83.
20. Schön EM, Winter D, Escalona MJ, Thomaschewski J. Key Challenges in Agile Requirements Engineering. In International Conference on Agile Software Development; 2017: Springer, Cham. p. 37-51.
21. Mattos DI, Bosch J, Olsson HH. Challenges and Strategies for Undertaking Continuous Experimentation to Embedded Systems: Industry and Research Perspectives. In International Conference on Agile Software Development; 2018: Springer, Cham. p. 277-292.