Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 197 (2022) 290–300

Procedia Computer Science

www.elsevier.com/locate/procedia

Sixth Information Systems International Conference (ISICO 2021)

Agile project management challenge in handling scope and change: A systematic literature review

Primadhika Marnada*,Teguh Raharjo, Bob Hardian, Adi Prasetyo

Faculty of Computer Science, University of Indonesia, Depok, Indonesia

Abstract

Uncertainty has become a challenge in project management during the age of the Covid-19 pandemic. Project management required a more flexible and dynamic approach; therefore, it is believed that Agile project management could provide a solution to this problem. Agile has been widely used in multiple industry areas as a practical and flexible approach in project management practices. In the project, the scope is considered one aspect that can directly affect the budget and timing. Unfortunately, it seems to be one of the most neglected domains in Agile and conventional. Agile is recognized for its rapid improvement as well as its willingness to embrace change. However, uncontrolled change can potentially delay and make the project overspend. This paper aims to examine the challenges and discover the best practices in handling scope and change in Agile. A systematic literature review (SLR) was carried out through the literature sources, which entailed IEEE Explore, ScienceDirect, Emerald Insight, ProQuest, and Wiley Online. The finding was seven categories, and eighteen issues were found. Then twenty-four best practices were mapped to the challenges. We found the most significant challenge came from People & Organization, User Requirement Prioritization, Over-scope Requirement, and Communication & Coordination. Those areas are considered the most significant challenge because they can create delays and overspend in the project.

Keywords: Agile; scope; change; project management; systematic literature review.

1. Introduction

Agile project management has been widely used in multiple industry areas as a practical and flexible approach in

* Corresponding author. Tel.: +62-81514-882353; fax: -.
E-mail address: primadhika.marnada01@ui.ac.id

1877-0509
10.1016/j.procs.2021.12.143

project management practices. The 14th Annual State of Agile Report shows that [1] Agile adoption has increased by 33% in response to Covid-19, with 60% of respondents claiming that Agile has assisted their speed to market. Agile has proven to be one of the methodologies capable of solving complex issues and adapting rapidly to business changes in the era of agility and rapid transformation by staying close to the customers [2].

In the project, the scope is considered one aspect that can directly affect the budget and timing. However, it seems to be one of the most neglected domains in agile and conventional [3]. Agile is recognized for its rapid improvement as well as its willingness to embrace change. [4]. Change of scope is considered not "creep" because, in Agile, it was accepted and expected as part of the project and will be managed throughout the project. In a more conventional perspective, as in Projects In Controlled Environments (PRINCE2), change is considered a risk [5]. Every change needs to be carefully maintained because it can affect the project's timing and budget, and that is why the scope needs to be defined clearly at the beginning of the project. There are five levels in Agile methodology considered as a process to define project scope, i.e., *product vision planning, story mapping planning, release planning, iteration planning,* and *daily meeting* [6].

Despite having those five levels, the nature of Agile was flexible and dynamic [3], this is because, during the revisit in each iteration planning phase, the customer is involved in the process when defining the scope and definition of done; and keep validating the features of the product to be developed [6]. Features that have a more significant impact on the market need to be developed and deployed first. Those processes make Agile final scope change up to 30% during each iteration [3]. Therefore, it can potentially bring the project into delay and increase the cost of the product. Some organizations with strict regulations in the project and fixed budgets make the scope management in Agile more challenging.

The previous study of scope management in a project shows that scope is one of the importance of the project's knowledge area that, if not appropriately managed, can lead to project failure [3], [7]. This uncontrolled scope was defined in PRINCE2 as scope creep, which will impact the project either budget or timing [5]. Agile handles this scope because, as mentioned before, Agile is more welcome to the changes. This paper aims to study the challenges in handling scope and the contract that occurs in the project based on that description.

2. Literature Review

In project management, the scope is defined as a specification document that includes the work's detail and the process involved to create or deliver the project's purpose [7]. Thus, scope management is the process of defining and controlling what should or should not be included in a project. It also defines the baseline for performance measurements and for controlling the project, and communicating clear responsibilities. To get a better understanding of the definition of the scope in the project management. We conducted the literature review on the most commonly used guidelines and frameworks in project management, *Project Management Body of Knowledge (PMBOK),* and *Projects in Controlled Environment (PRINCE2),* then compared the definition with agile project management.

2.1. Scope in Project Management Body of Knowledge (PMBOK)

The Project Management Institute (PMI) developed the Project Management Body of Knowledge (PMBOK) to establish project management knowledge principles. The purpose is to assist a project manager in completing a project by providing an up-to-date knowledge base and framework for project management [8]. The PMI has published six editions of the PMBOK, with the latest edition was the 6th edition published in 2017. The 6th Edition of the PMBOK categorized the processes into ten knowledge areas; *Project Integration Management, Project Scope Management, Project Schedule Management, Project Cost Management, Project Quality Management, Project Resources Management, Project Communication Management, Project Risk Management, Project Procurement Management,* and *Project Stakeholder Management* [8].

In terms of scope management, the 6th Edition of the PMBOK Guide defines the scope as the processes required to ensure that the project includes all the work required to complete it successfully [8]. The scope is defined in the project planning phase and became part of the project plan document. This project plan document acts as guidance in the project because it consists of many details required.

2.2. Scope in Projects In Controlled Environments (PRINCE2)

The Projects in Controlled Environments (PRINCE2) is a project management methodology structured around the experience gained through thousands of projects and the contributions of numerous sponsors, managers, project teams, academics, trainers, and consultants [9]. It was developed in 1989 by The Central Computer and Telecommunications Agency (CCTA) and is now referred to as The Office of Government Commerce (OCGC). In 2013, the ownership then changed to AXELOS, which brought a major update of PRINCE2 and PRINCE2 Agile. In terms of obtaining a successful project, PRINCE2 brought the guidelines and framework then divided them into three categories: *seven principles, seven themes,* and *seven processes* [5].

In PRINCE2, the scope was defined as what the project will deliver [5]. There should be an explicit agreement between the project manager and the customer about the project's scope before starting the project. The project manager should understand what is and is not in the project's scope and not deliver beyond this scope because it can delay the project, overspend, and create uncontrolled changes [5]. Those processes should be followed because the contract is also based on the agreed-upon scope. Therefore, there should be a clear understanding between the project manager and the customers. Any significant change in the middle of the project might lead to a delay and an overspend.

2.3. Scope in Agile Project Management

The Agile approach to the project's scope differs from PMBOK and PRINCE2. Agile is a welcomed fresh idea and the fresh ideas at later stages of the project. Therefore, volatility in the project scope is accepted [7]. During each iteration phase, the product owner and the customer collaborated to control and verify the scope. This process defines which features are accepted or rejected that are completed during the iteration phase. That process is natural because the Agile approach entails close interaction with the client throughout the project's development. These processes were done to avoid rework, unexpected delays, overspend, and customer dissatisfaction [10].

However, this approach potentially brings a new risk to the project. Changes occur more frequently during the development phase, and there is a possibility of undesired changes at the end of the iteration. These continuous changes can increase the risk in the project. Moreover, the Agile risk mitigation approach is not considered an essential point of view [10]. Fig. 1 shows the example of potential risk on Agile where each sprint contained potential hazards such as *scope creep, unrealistic expectations, lack of cooperation,* and *lack of communication*.

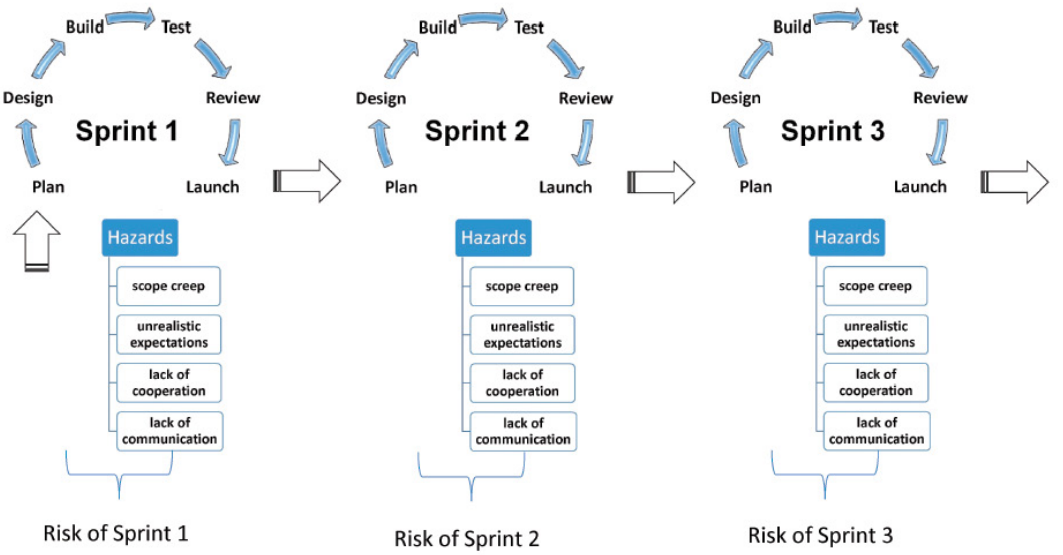

Fig. 1. Example potential risk on agile [10]

Despite this requirement, relatively few literary works transcend the Agile constraint on change and scope management. As such, this research is being performed to investigate and respond to the following research questions (RQ):

RQ1: What were the challenges to handle scope and change in agile project management?

RQ2: What were the best practices to handle those challenges in agile project management?

3. Research Methodology

This research uses a systematic literature review (SLR) to compile a precise, methodical, exhaustive, and comprehensive assessment of Agile's scope and change by doing the scientific mapping. SLR was used to illustrate the researcher's review and highlight the critical research field [11]. Thus, it enables the researcher to comprehend how the scientific field progresses and synthesizes disparate pieces of knowledge.

Kitchenham proposes this methodology in 2004 and 2005 to adopt evidence-based software engineering (EBSE) and the user systematic literature review to support EBSE [12]. This methodology aims to search, identify, and validate multiple variables on related research, then synthesize the result empirically. The process to do SLR was divide into three processes, planning, implementation, and reporting. Fig. 2 shows the details of those three processes to do the SLR. These three processes will be discussed separately in each sub-section.

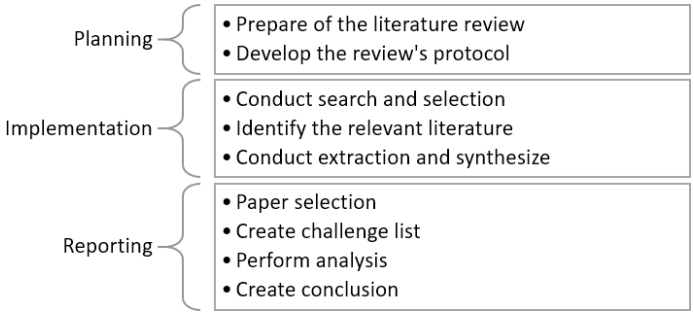

Fig. 2. SLR methodology

3.1. Planning The SLR

The first step in SLR is to conduct SLR planning. This step involves developing the review's protocol. The strategy involved defining a specific keyword and conducting a methodical search on IEEE Explore, ScienceDirect, Emerald Insight, ProQuest, and Wiley Online. The results were filtered to match the related research topic. The keyword phrase for this research topic should contain the following terms: challenge, agile, scope, and changes. The first keywords used in the boolean search to locate and identify the challenge are as follows. ("ISSUE" OR "PROBLEM" OR "CHALLENGE" OR "LIMITATION" OR "OBSTACLE").

Since the research topic relates to Agile project management, the second keyword to be searched was "AGILE." The following keyword has any relation with the scope and changes specifically in an information technology project. Then we conduct the search with the following keyword as the boolean search ("PROJECT" OR "CHANGE" OR "SCOPE" OR "PLAN" OR "MANAGEMENT" OR "SPRINT" OR "INFORMATION SYSTEM" OR "INFORMATION TECHNOLOGY" OR "METHOD" OR "METHODOLOGY"). Then we combine from first, second, and third keywords into the following keyword ("ISSUE" OR "PROBLEM" OR "CHALLENGE" OR "LIMITATION" OR "OBSTACLE") AND "AGILE" AND ("PROJECT" OR "CHANGE" OR "SCOPE" OR "PLAN" OR "MANAGEMENT" OR "SPRINT" OR "INFORMATION SYSTEM" OR "INFORMATION TECHNOLOGY" OR "METHOD" OR "METHODOLOGY"). These keywords became the review protocol for this SLR, which will be used to locate pertinent literature in IEEE Explore, ScienceDirect, Emerald Insight, ProQuest, and Wiley Online.

The inclusion (IN) and exclusion (EX) criteria were used to restrict the search results. The study has to have been published between January 2016 and January 2021 (IN1), be written in English (IN2), be published in international journals and conferences (IN3), and be focused on Agile methods or appropriate search criteria for the research subject (IN4). To avoid being omitted from this SLR, the linked study can not be a presentation, opinion, or suggestion article (EX1). The focus is confined to the challenges inherent in managing change and scope in Agile project management (EX2).

3.2. Implementation of the SLR

The second step of the SLR is searching and selecting relevant literature, identifying it, and extracting and synthesizing it. Fig. 3 shows the process of those three processes of implementation of SLR.

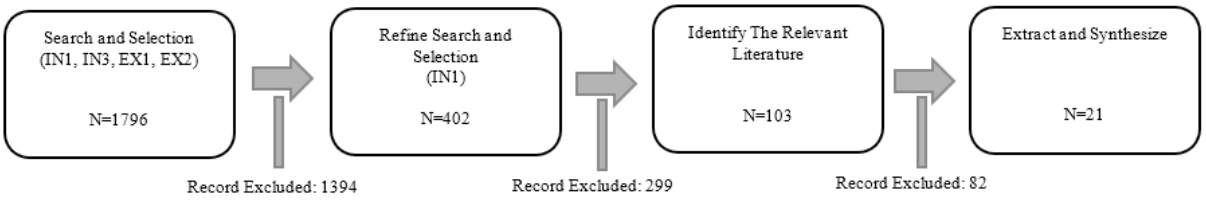

Fig. 3. Extraction process

The *Search and selection* process was initiated by performing searches from IEEE Explore, ScienceDirect, Emerald Insight, ProQuest, and Wiley Online. This phase involved the application of inclusions IN2 and IN3, as well as exclusions EX1 and EX2. This phase resulted in the discovery of 1,796 publications and conferences. The second step in this phase is to conduct inclusion IN1, and the resulting list of 402 manuscripts of journals and conferences was filtered. Identifying Relevant Literature began with a keyword filter to narrow the scope, change, and difficulty. The resulting list of 103 publications and conferences was filtered.

The Extract and Synthesis were carried out manually by examining each paper relevant to the research topic. This investigation was carried out manually by reading the relevant text contained within the documents. As a result of this step, 21 articles were selected for publication.

3.3. Reporting The SLR

The final process of the SLR was to complete the SLR's reporting. The paper extracted in the previous steps will be further analyzed in these steps. The previous extraction process narrowed the field to 21 journals, and conferences were chosen to analyze the challenge of managing change and scope and identify best practices for overcoming the obstacle. Table 1 summarizes the distributed studies between 2016 and 2021, while Table 2 details the journals.

Table 1. Distribution of study per year

Year	Frequency	Percentage
2016	2	9,52
2017	3	14,29
2018	7	33,33
2019	7	33,33
2020	2	9,52
Total	21	100

Table 2. Extracted journals or papers

No	Title	Year	Code
1	Agile Challenges and Chances for Open Source Lessons Learned from Managing a FLOSS Project	2019	[13]
2	Agile development as a change management approach in software projects Applied case study	2016	[14]
3	An Empirical Investigation on Requirements Change Management Practices in Pakistani Agile Based Industry	2019	[15]
4	Calculating Completeness of Agile Scope in Scaled Agile Development	2017	[3]
5	Challenges in agile software development A systematic literature review	2018	[16]
6	COOC An Agile Change Management Method	2019	[17]
7	Correlation of change size to fault-proneness A real-life case study	2016	[18]
8	Harmonizing IT Frameworks and Agile Methods Challenges and Solutions for the case of COBIT and Scrum	2020	[19]
9	Identification and Prioritization of Agile Requirements Change Management Success Factors in the Domain of Global Software Development	2020	[20]
10	Managing Change in Agile Software Development a Comparative Study	2018	[21]
11	Methodology for Determining Agile Product Scopes in Development Projects	2018	[22]
12	On Interfaces to Support Agile Architecting in Automotive an Exploratory Case Study	2019	[23]
13	Prioritizing User Requirements for Agile Software Development	2018	[24]
14	Quantitative planning and risk management of Agile Software Development	2017	[25]
15	Reducing the Costs of Engineering Design Changes Through Adoption of a Decision Support and Knowledge Management System Early in the Design	2019	[26]
16	Requirement change taxonomy and categorization in agile software development	2018	[27]
17	Toward successful agile requirements change management process in global software development: a client-vendor analysis	2019	[28]
18	A supporting tool for requirements change management in distributed agile development	2017	[29]
19	Enabling agility in product development through an adaptive engineering change management	2018	[30]
20	The Change In Management Style During The Course Of A Project From The Classical To The Agile Approach	2018	[31]
21	A Novel Framework for Change Requirement Management (CRM) In Agile Software Development (ASD)	2019	[32]

4. Result and Discussion

This section discusses the systematic review method in detail. The challenge was mapped based on the corresponding research in relation to scope and change. The procedure that followed was to establish the best practices for resolving the issues. Each of these processes is described in more detail in the following sub-sections.

4.1. Mapping the Challenges on Related Studies

The first part outlines the challenges of associated research previously and therefore answers RQ1. It was analyzed and discovered that 18 issues were detected and divided into seven categories, as Table 3 shows.

Table 3. List of validated challenges

No	Category	Issue	Reference
1	People & Organization [17]	No specific role to handle and control scope & change	[13], [17], [30]
		No clear procedure to control scope & change	[17], [30]
		Unclear roles and responsibilities among team members	[20], [30]
		Time-consuming bureaucracy	[17]
2	User Requirement Prioritization [24]	No specific procedure to define the level of prioritization of the change	[21], [15], [24], [30]
3	Over-scope Requirement [16]	Unreasonable and unsystematic change requests	[16], [32]
		Objectives are poorly defined	[17], [15]
		Unclear completeness in each sprint	[3]
		No document for maintaining version history	[15], [32]
		Low knowledge of the product	[26]
4	Tools & Process [3]	No tools to control version history	[15], [29]
		No tools to control the scope	[3], [30]
		Unavailability of a newer technology to handle a change request	[3]
5	Product Backlog [3]	No clear phase on each iteration	[17], [15], [3]
6	Communication & Coordination [17]	Low participation rate from member and stakeholder	[17], [20], [32]
		Lack of communication during changes	[17], [15], [20], [28], [30], [32]
7	Culture and Behavior [17]	Team member unaccustomed to change	[17]
		Numerous aversions to change	[17]

From Table 3, we selected the first category from the SLR as people and organization. Le Grand and Rebecca (2019) assert that agile change management entails a greater degree of human engagement [17]. The first problem in this area is that there is no designated position for managing and controlling scope and change [13] [17], and there is no clear process for controlling scope and change [17]. We are aware that some organizations lack a dedicated function and process for managing and controlling the scope and change of research. These problems get more complicated since companies "ideally" need IT Service Management (ITSM) to manage and track any product modifications. Due to the lack of maintenance, the organization will have no record of the modifications made to the systems. It has the potential to cause the project to be delayed or overspent. The second problem was a lack of clarity about the roles and duties of team members [20]. Defining roles and duties is a significant challenge since some companies lack sufficient resources, causing roles and responsibilities among team members to overlap. The last problem we found was lengthy bureaucracy [17]. Some organizations, such as the government, adhere to strict regulations when it comes to project changes. While Agile's nature is adaptable and embraces change as an opportunity for development, it requires a prompt reaction from stakeholders to any changes. As a result, synchronizing this with bureaucracy has become very difficult.

The second category of the challenge is user requirement prioritization [24]. The issue in this category was the absence of a standardized procedure for establishing the priority of the change [24] [15] [21]. This issue may result in uncontrollable change throughout the process, putting the project at risk of being disrupted

The third category is the requirement that is too broad in scope. According to Fitriani, Rahayu, and Sensuse (2018), the over-scope requirement was identified as a significant challenge among 30 Agile software development challenges [16]. That is because unreasonable and unsystematic change [16] can put the development team in a pickle, resulting in task overlap on the project. The following issue we identified was that objectives were poorly defined [15] [17] and that each sprint's completion status was unclear [3]. Agile, on the other hand, embraces scope changes during sprint planning.

On the other hand, uncertain objectives that lack a methodology for determining the scope's completeness can result in sprint inconsistency. The most recent issue in this category concerned the absence of a document for tracking version history [15] and a lack of product knowledge [16]. These two issues were identified as a challenge because some organizations do not maintain an up-to-date version history, making tracing the app's history difficult. New team members who have no prior knowledge of the product will have difficulty comprehending it.

The fourth challenge category is Tools & Process. According to Amjad et al. (2017), some scope-related issues in Agile are caused by a lack of scope definition tools and methods, which results in project failure [3]. We identified a lack of tools for controlling scope [3] and a lack of tools for controlling version history [15] as issues. Maintaining the project's scope was deemed critical. A lack of tools can have a direct impact on the budget and schedule. Another issue we identified was the absence of a newer technology capable of handling change requests [3]. This could potentially result in difficulties completing the project's scope on time and within budget.

The fifth category in the recent studies was Product Backlog. According to Fitriani, Rahayu, and Sensuse (2018), another challenge in agile software development was maintaining consistency in distributed teams' design standards [3]. Additionally, the lack of a defined phase for each iteration [3] [15] [17], such as a sudden change in the middle of the iteration, can present a problem when completing the product backlog.

The sixth category extracted from the validated paper was communication & coordination. Le Grand and Rebecca (2019) mentioned in their research that the more we move forward in the project, the more communications actions must occur, and the more the project must be known [17]. Two main issues were a low participation rate from members and stakeholders [17] [20] and lack of communication during changes [17] [15] [20]. These two issues can create misunderstanding and reduce the awareness of why the changes are being applied.

The last category in the related studies was culture and behaviour. According to Le Grand and Rebecca (2019), one of the cartographies in change was culture and behaviour. Agile nature was continuous changes and improvement. It became a challenge in terms of scope and change because some company members were unaccustomed to change [17] and have numerous aversions to change [17].

4.2. Mapping of The Best Practices

This section was conducted to answer RQ2. The best practices from the extracted paper were collected and analyzed, then mapped to sub-section 4.1's challenges. The detail was shown in Table 4 and Table 5 (cont'd)

Table 4. List of validated best practices

No	Category	Solution	Reference
1	No specific role in handling and controlling scope & change	Define a precise level of roles in handling scope and change	[17], [28]
		Do proper training to the employee for a change	[15], [28]
		Develop an Agile Mindset Organization	[31]
	No clear procedure to control scope & change	Implement the COOC Method	[17], [28]
		Define clear procedure in change	[21], [28], [29], [32]
	Unclear roles and responsibilities among team members	Assign the accurate roles and responsibilities to the right team	[20], [19], [13], [3], [21], [28]
	Time-consuming bureaucracy	Conduct training and workshops to the organization element using the COOC Method	[17], [28]
2	No specific procedure to define the level of prioritization of the change	Establish the change prioritization	[24], [15], [21], [22], [23], [3], [16], [19], [20]

Table 5. List of validated best practices (cont'd)

No	Category	Solution	Reference
3	Unreasonable and unsystematic change requests	Commitment to the sprint planning	[15], [3], [20]
	Objectives are poorly defined	Product Owner and critical stakeholders assess the progress of product deliverable and provide feedback	[16]
	Unclear completeness in each sprint	Measure the quality and completeness of scope	[17]
	No document to maintaining version history	Establish document to maintain version history	[15]
	Low knowledge of the product	Collaborate and share knowledge within the project team	[13]
4	No tools to control version history	Implement tools for traceability and supporting document	[15], [19], [29]
	No tools to control scope	Implement planning tools to support changes	[3], [17], [23], [25], [29]
	Unavailability of a newer technology to handle a change request	Reevaluated schedule and design a new model or prototype to achieve change scope of an Iteration	[3]
5	No clear phase on each iteration	Clarify the component (goals, resource, schedule, task) in each iteration planning	[3]
		Changes that occur during iterations are Prioritized	[15]
		Clear checkpoint on each iteration	[19]
6	Low participation rate from member and stakeholder	Make training and workshop more playful than simple training	[17]
		Build trust among team member's	[3], [32]
		Maintain frequent communication to the team and stakeholders	[3], [14], [15]
7	Team member unaccustomed to change	Implement the COOC Method	[17]
	Numerous aversions to change	Implement the COOC Method	[17]

From Table 4, we understand that multiple solutions can be implemented in more than one category of issues. From the organizational perspective, we knew now that an Agile mindset should be developed within the organization. Some best practices such as defining a precise level of roles and conducting proper training to manage the change could be conducted. Creating a clear procedure or framework; also consider it essential to face the change and manage the scope. Best practices, such as COOC, also can be implemented for multiple issues. Le grand and Rebecca (2019) introduce the Concept of Continuous Change (COOC) as an exciting concept and approach to handling change management in Agile. In the concept, change management is defined into 3 phases: *define phase, experiment phase,* and *anchor phase*. Where in each of the phases, consist of a set of method or cycle.

We understand that due to a lack of procedure, some companies challenge implementing change management. According to Raza and Waheed (2018), a clear procedure in the change request flow was necessary. Apart from the procedure, the definition of clear roles also considers an essential aspect of handling change. We emphasize priority; nine studies have shown the same best practices as prioritizing changes per related studies. The impact of the associated constraints, such as schedule, budget, technical aspects, business aspect and priority stakeholders, should be carefully assessed.

The scope and requirements were subsequently emphasized. An unreasonable and unsystematic change request can result in a poorly defined goal. The product owner and key stakeholders should evaluate the product progress achieved and provide feedback at the end of the sprint. Amjad et al. (2018) also indicated it was necessary to measure the quality and completeness of the scope [3]. Additionally, the studies demonstrate the importance of adequately documenting version history to avoid the over-scope requirement. The team should share its knowledge with other team members to close the team's knowledge gap. However, and perhaps most significantly, the team should commit to the sprint planning outcome [15].

Apart from the sprint planning, an ambiguous iteration phase can make it challenging to maintain the project's scope. According to Amjad et al. (2018), we must clarify each component (goals, resources, schedule, and task) during

iteration planning. Additionally, as with the SDLC, it was recommended to include a clear checkpoint in each iteration phase [19].

Additionally, communication and coordination were discussed. Throughout the changes, it was critical to maintaining frequent communication with the team and stakeholders. One of the aspects that can be considered best practices was the development of trust between team members. Establishing trust among team members is critical for fostering a collaborative environment, ensuring individual rights, and ensuring equal participation throughout the project [17]. Another way to improve communication and coordination is to conduct more playful training and workshops than standard training, thereby strengthening the project team's bonds [17].

The final point was the critical nature of utilizing tools to manage change and scope. According to Batool and Inayat (2019), one method for maintaining the change history was to use a version control tool to document the change history for future maintenance projects [15]. Jira and Microsoft TFS can be used to support this. The benefit of using tools is to control the scope that removes the bottleneck and improves throughput by monitoring and controlling progress [3]. The TinyPM, ScrumDesk, Agile for Trac, and VersionOne are all tools that can be used as planning tools.

5. Conclusion

The research was conducted to find the challenges in handling change and scope. Then find the best practice for those challenges. An SLR was conducted, then seven categories and eighteen issues were found without reducing the other challenge's value. We know that the most significant challenge came from *People & Organization*, *User Requirement Prioritization*, *Over-scope Requirement*, and *Communication & Coordination*. Those areas are considered the most significant challenge because they can create delays and overspend in the project.

The study's implication was to inform organizations or practitioners who need to find the challenge in handling scope & change in Agile and best practices to handle those challenges, especially for Agile during the pandemic nowadays. Although Agile has many challenges, this research is focused only on finding challenges in handling scope and change.

Future Work

Future research can be conducted to add mitigation and seek more solutions and techniques as the best practices. This study only retrieved the relevant literature from *IEEE Explore, ScienceDirect, Emerald Insight, ProQuest,* and *Wiley Online* within the last five years of publication. The papers or relevant literature should be retrieved from other publication databases into a broader publication period to broaden this research topic's study in the future.

References

[1] Digital.ai. (2020) "14th Annual State of Agile Report." [Online]. Available: https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494.

[2] Raharjo, T., and B. Purwandari. (2020) "Agile project management challenges and mapping solutions: A systematic literature review." *Proceedings of the 3rd International Conference on Software Engineering and Information Management*: 123–129.

[3] Amjad, Sunnia, Naveed Ahmad, Tanzila Saba, Adeel Anjum, Umar Manzoor, Muhammad A. Balubaid, and Saif Ur Rehman Malik. (2018) "Calculating Completeness of Agile Scope in Scaled Agile Development." *IEEE Access* **6**: 5822–5847.

[4] Sliger, M. (2010) "Goodbye, scope creep--hello, agile!" *PMI® Global Congress* 2010, North America, Washington, DC. Newtown Square, PA: Project Management Institute. [Online] Available: https://www.pmi.org/learning/library/project-scope-creep-agile-methodologies-6552.

[5] AXELOS. (2017) "Managing Successful Projects With Prince2 - 2017." *AXELOS Global Best Practice*.

[6] Hays, F. (2016) "A guide to the SCRUM BODY OF KNOWLEDGE (SBOK™ Guide): A comprehensive Guide to Deliver Project using Scrum.". *SCRUMstudy*. [Online] Available: http://www.cs.vsu.ru/~svv/spm/SBOK_Guide_3rd_edition_English_Sample.pdf.

[7] Rehman, I. U., S. Ullah, A. Rauf, and A. A. Shahid. (2010) "Scope management in agile versus traditional software development methods." *Proceedings of the 2010 national software engineering conference*: 1-6.

[8] PMBOK. (2017) "PMBOK Guide - 6th Edition." *Project Management Institute*.

[9] Matos, S., and E. Lopes. (2013) "Prince2 or PMBOK – A Question of Choice." *Procedia Technology* **9**: 787–794.

[10] Anes, V., A. Abreu, and R. Santos. (2020) "A new risk assessment approach for agile projects." *2020 International Young Engineers Forum (YEF-ECE)*: 67–72.

[11] Chen, C. (2017) "Science Mapping: A Systematic Review of the Literature." *Journal of data and information science* **2 (2)**.

[12] Kitchenham B., and P. Brereton. (2013) "A systematic review of systematic review process research in software engineering." *Information and software technology* **55** (**12**) : 2049–2075.

[13] Muller, M. (2018) "Agile Challenges and Chances for Open Source: Lessons Learned from Managing a FLOSS Project." in *2018 IEEE Conference on Open Systems (ICOS)*: 1–6. doi: 10.1109/ICOS.2018.8632819.

[14] Alawairdhi, M. (2016), "Agile development as a change management approach in software projects: Applied case study." in *2016 2nd International Conference on Information Management (ICIM)*,: 100–104. doi: 10.1109/INFOMAN.2016.7477541.

[15] Batool, K., and I. Inayat. (2019) "An Empirical Investigation on Requirements Change Management Practices in Pakistani Agile Based Industry." in *2019 International Conference on Frontiers of Information Technology (FIT)*,: 7–75. doi: 10.1109/FIT47737.2019.00012.

[16] Hacaloglu T., and O. Demirors. (2018) "Challenges of using software size in agile software development: A systematic literature review." *CEUR Workshop Proc.* **2207**: 109–122.

[17] Le Grand, T., and R. Deneckere. (2019) "COOC: An Agile Change Management Method." in *2019 IEEE 21st Conference on Business Informatics (CBI)*, **2**: 28–37. doi: 10.1109/CBI.2019.10093.

[18] Zeindl, M., M. Seitz, and C. Facchi. (2016) "Correlation of change size to fault-proneness: A real life case study." *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*: 1-6.

[19] Ozkan, N., A. Tarhan, B. Gören, İ. Filiz, and E. Özer. (2020) "Harmonizing IT Frameworks and Agile Methods: Challenges and Solutions for the case of COBIT and Scrum." in *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems, FedCSIS 2020*, **21**: 709–719. doi: 10.15439/2020F47.

[20] Kamal, T., Q. Zhang, M. A. Akbar, M. Shafiq, A. Gumaei, and A. Alsanad. (2020) "Identification and Prioritization of Agile Requirements Change Management Success Factors in the Domain of Global Software Development." *IEEE Access* **8**: 44714–44726.

[21] Raza, S., and U. Waheed. (2018) "Managing Change in Agile Software Development a Comparative Study." in *2018 IEEE 21st International Multi-Topic Conference (INMIC)*,: 1–8. doi: 10.1109/INMIC.2018.8595457.

[22] Schuh, G., C. Dolle, F. Diels, and M. Kuhn. (2018) "Methodology for Determining Agile Product Scopes in Development Projects." in *2018 Portland International Conference on Management of Engineering and Technology (PICMET)*,: 1–9. doi: 10.23919/PICMET.2018.8481926.

[23] Wohlrab, R., P. Pelliccione, E. Knauss, and R. Heldal. (2019) "On Interfaces to Support Agile Architecting in Automotive: An Exploratory Case Study." in *2019 IEEE International Conference on Software Architecture (ICSA)*, **(i)**,: 161–170. doi: 10.1109/ICSA.2019.00025.

[24] Sachdeva, S., A. Arya, P. Paygude, S. Chaudhary, and S. Idate. (2018) "Prioritizing User Requirements for Agile Software Development." *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*: 495–498.

[25] Ghane, K. (2017) "Quantitative planning and risk management of Agile Software Development." in *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*: 109–112. doi: 10.1109/TEMSCON.2017.7998362.

[26] Jonkers, R. K., and K. E. Shahroudi. (2019) "Reducing the costs of engineering design changes through adoption of a decision support and knowledge management system early in the design." *2019 IEEE International Systems Conference (SysCon)*.

[27] Saher, N., F. Baharom, and O. Ghazali. (2017) "Requirement change taxonomy and categorization in agile software development." in *2017 6th International Conference on Electrical Engineering and Informatics (ICEEI)*, **2017-Novem**: 1–6. doi: 10.1109/ICEEI.2017.8312441.

[28] Kamal, T., Q. Zhang, and M. A. Akbar. (2020) "Toward successful agile requirements change management process in global software development: a client–vendor analysis." *IET Software* **14** (**3**): 265–274.

[29] Lloyd, D., R. Moawad, and M. Kadry. (2017) "A supporting tool for requirements change management in distributed agile development." *Future Computing and Informatics Journal* **2** (**1**): 1–9.

[30] Schuh, G., T. Gartzen, S. Soucy-bouchard, and F. Basse. (2017) "Enabling agility in product development through an adaptive engineering change management." *Procedia CIRP* **63**: 342–347.

[31] Gablas, B., E. Ruzicky, and M. Ondrouchova. (2018) "The change in management style during the course of a project from the classical to the agile approach." *Journal of Competitiveness* **10**: 38–53.

[32] Shehzadi, Z., F. Azam, M. W. Anwar, and I. Qasim. (2019) "A Novel Framework for Change Requirement Management (CRM) In Agile Software Development (ASD)." *Proceedings of the 9th International Conference on Information Communication and Management*: 2–6.