

The Essence of Software Engineering Framework-based Model for an Agile Software Development Method

Teguh Raharjo¹, Betty Purwandari², Eko K. Budiardjo³, Rina Yuniarti⁴
Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia^{1, 2, 3, 4}

Abstract—Agile development's rapid growth is due to its ability to address complex problems and facilitate a smooth transition from traditional methods. However, no single Agile method can fit every organization, which leads to a lack of adoption guidelines. It triggers this investigation by proposing an Agile development method model based on the Essence of software engineering framework and incorporating the common ground of popular methods such as Scrum, Kanban, Extreme programming, SAFe, LeSS, Nexus, Spotify Agile, Scrum of Scrums, and Disciplined Agile. The Essence of software engineering framework provides an approach for organizations to develop software development methods based on common ground or shared understanding among methods. We enhance this approach for Agile methods, resulting in a model to support organizations in developing their Agile methods and practices. Moreover, Design Science Research (DSR) was employed as a methodology to construct the artifact, demonstration, and evaluation. We demonstrated the model in an Agile product development at a national-wide bank in Indonesia. This investigation enhances Agile methods in SWEBOK's Software Engineering Models and Methods knowledge area, benefiting academics and practitioners. Practitioners can use the model as a reference to implement their Agile projects.

Keywords—Agile; common ground; the essence of software engineering framework; Design Science Research (DSR)

I. INTRODUCTION

At present, the development of the Agile approach is very rapid in organizations [1]. This approach has a significant influence on the growth of businesses and the performance of projects, enabling them to tackle complex problems in an era of rapid disruptions [2]. The concept of Agile innovation teams, aimed at maintaining proximity to customers and swiftly adapting to evolving business conditions, is already well-known to most management levels [3]. Moreover, a captivating study conducted across multiple organizations revealed that companies that cultivate an adaptive Agile culture witnessed revenue growth exceeding the pace multiple times [4]. Therefore, the Agile approach must be scaled up at the enterprise level to handle multiple teams and projects. While organizations are transitioning to Agile, they are also looking to scale up [5]. The scaling approach provides value and benefits to the business operations and supports [6].

The organization could adopt the Agile methods to perform Agile implementation. Although several Agile methods offer the solutions, such as Scrum, Kanban, Extreme programming, SAFe, LeSS, Discipline Agile, Nexus, and Spotify Agile, no single Agile method can fit every organization [7], which leads to a lack of adoption guidelines.

To the best of our knowledge, there is no common ground or shared understanding of the Agile methods or the guideline for the organization to adopt the methods, as shown in Fig. 1. This is the problem we need to solve.

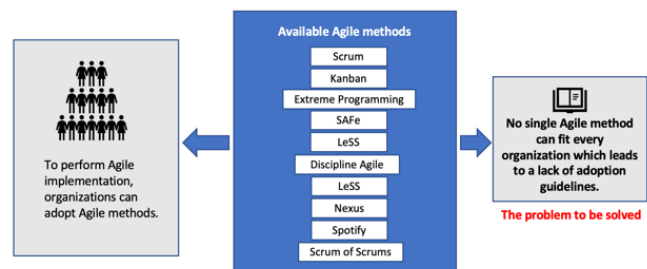


Fig. 1. The problem needs to be solved

No studies currently explore a model for Agile methods in organizations. The Essence provides a common ground for all methods in software engineering [8]. It offers a thinking framework for the team to collaborate; resources for discussing, improving, comparing, and sharing methods and practices; a foundation for defining practices independent of methods.

This research aims to develop a model for an Agile method development based on selected practices, drawing from the Essence framework. The intended scope of this Agile method is up to scaling Agile, as it already encompasses the Agile method within it. This study serves as an academic reference for Agile methods. From the practitioners' perspective, the results can guide the organization to apply the scaling Agile methods. Additionally, the model can be integrated with the hybrid agile management approach [9], and DevOps [10] for organizations seeking a hybrid implementation. These contributions make this study unique.

The remaining sections of this article are structured in the following manner: Following this introduction, the subsequent section offers a brief overview of the related work, covering concepts such as the common ground concept, scaling Agile methods, and the Essence framework. Section III explores the research methodology employed, specifically focusing on design science research. Moving on, Section IV presents the study's findings, which is resulting in the model, namely an Essence-based Agile method development model. Subsequently, in Section V, we present the case study of the software development organization in one bank in Indonesia. Section VI explains the evaluation of our research, respectively. Lastly, Section VII is the concluding section provides a summary and closure to the paper.

II. RELATED WORK

This section discusses related work. First, we review the concept of common ground based on the Essence of software engineering as the reference for this study. Then, we explore the current scaling Agile methods, such as SAFe, LeSS, Nexus, Disciplined Agile, Spotify Agile, and Scrum of Scrums. We focus on the scaling methods, including the individual Agile methods, such as Scrum, Kanban, and Extreme Programming. We only select the most popular scaling Agile methods based on a survey as the main comparison [5, 11]. The other methods will fill the gaps in the practices implemented in the case study.

A. The Common Ground Concept in the Software Engineering

The concept of the common ground in this study refers to the common ground in software engineering introduced by Ivar Jacobson [8]. He introduced the Essence kernel for software engineering. The common ground is developed by thinking that no software development method is appropriate for everyone. The power of the common ground is to provide a common framework for the team to understand the general concept of the software development methods. The common ground is employed for discussing, improving, comparing, and sharing software engineering methods and practices. This approach and inspiration of the common ground in software engineering are used in scaling Agile methods.

B. Scaling Agile Methods

The popular Agile methods, including Scrum [12] and Extreme Programming [13], work well for the small project team. When there is a need to scale up a project at the enterprise level, the complex situation is hard to handle [14]. Project Management Institute [7] categorizes Agile methods as the scaling approach and team method, as demonstrated in Fig. 2. Some thinkers, such as Dean Leffingwell and Scott Ambler, developed several enterprise Agile methods, namely Scaled Agile Framework [15] and Disciplined Agile Delivery [16]. There are no formal names for these methods. The literature declared them as enterprise Agile framework [11], scaling methods and approach [11], scaling framework [5], and scaling Agile methods [17]. The Scaled Agile Framework (SAFe), Large-Scaled Scrum (LeSS), Disciplined Agile Delivery (DAD), and Nexus are some of the popular scaling Agile methods and frameworks [5, 10].

Most scaling Agile methods exercise Scrum, Kanban, and Extreme Programming as part of their team methods. On the other hand, integrating Scrum and Kanban is employed based on the needs [18]. These methodologies offer flexibility and adaptability to different projects. For instance, Scrum provides a structured framework with defined roles and ceremonies. Kanban, on the other hand, is well-suited for projects that require continuous delivery. Research in [19, 20, 21, 22, 23] has been performed on this scaling approach for Agile implementation in organizations.

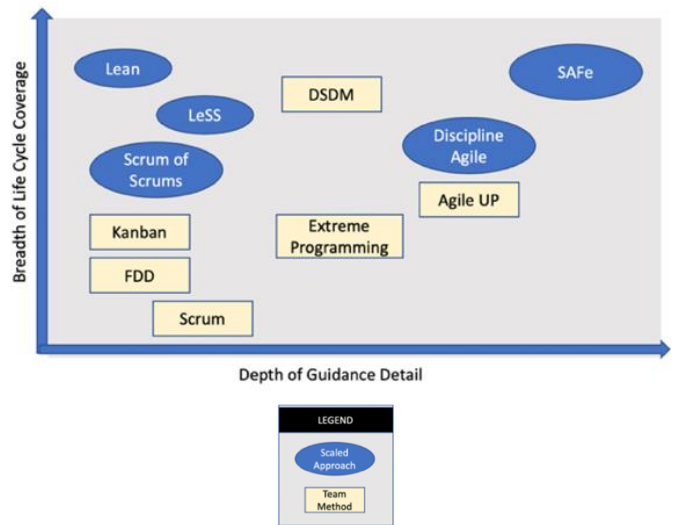


Fig. 2. Classification of agile based on scaling level [7].

C. The Essence – The SEMAT Kernel

The Essence is the standard of software engineering that provides a universal language for defining methods and practices [24]. It was established by the Object Management Group (OMG) [25] and received strong support from the Software Engineering Methods and Theory (SEMAT). The Essence was built based on solid theory, proven principles, and best practices. It consists of methods, practices, kernel, and Language. Fig. 3 depicts that methods comprise many practices. In this context, a practice is a repeatable approach to performing activities with a specific objective. The kernel elements are used to elucidate the practices, and these kernel elements are specified based on the Language.

SEMAT was developed with the purpose of addressing certain challenges within today's software engineering domain, characterized by the presence of immature implementation. These issues comprise the absence of a universally acknowledged theoretical foundation, the abundance of numerous methods and their variations, and the disconnect between industry practices and academic research.

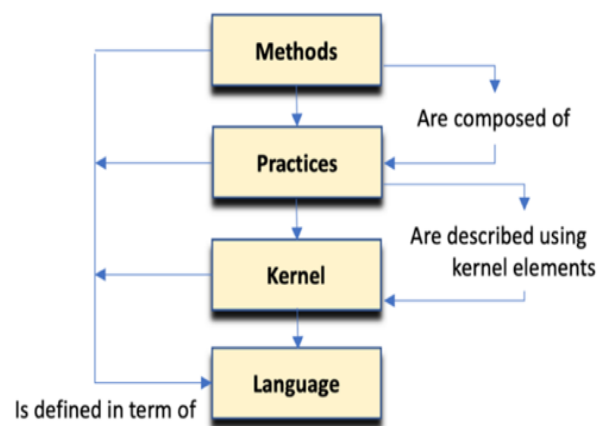


Fig. 3. The Essence method architecture [24].

The kernel contains three areas, including customer, solution, and endeavor. The Essence defines kernel Alphas as the things to work with. Within the customer domain, the team is required to comprehend stakeholders and potential opportunities. The team should address the requirements and software system concerns in the solution area. There are things to perform in the area of endeavor. They include work, team, and way of working.

D. The Previous Studies

Several studies compared the scaling Agile methods [16, 26], while there was another study [27] that made a method of selecting Agile methods in their project implementation. The study [17] tried to analyze the method differences and similarities for DAD, SAFe, LeSS, Spotify, Nexus, and RAGE, while [26] performed the comparison based on the underlying practices. This study differs from the previous studies, where this study develops the common ground based on the common ground of software engineering [8]. The common practices from [26] can be used in more detail for this research.

III. RESEARCH METHOD

Design Science Research (DSR) was employed as a research method for this study [28], as demonstrated in Fig. 4. The DSR methodology aims to solve the organization's problem by producing artifacts. The process begins with the identification problem, as discussed in Section I.

The next process defines the solution, design and development, demonstration, evaluation, and communication. This study aims to get the common ground from the current scaling Agile methods. The comprehensive literature review from current methods of scaling Agile methods was employed as the baseline for the basic common ground. The previous study performed the same [29, 30]. The Essence SEMAT kernel was the primary reference for developing the common ground [25].

We conducted this research using DSR as the comprehensive research method, from the initial stage of problem identification to the evaluation stage. Each phase of this DSR research is explained in its respective section. The research was conducted in the environment of one of the private banks in Indonesia, as described in Section V. We utilized data and information available at the case study location, including their current methodologies and practices. Research validation methods, as described in Section VI, were employed in this study, including internal and external validation through focus groups and interviews.

IV. RESULT AND DISCUSSIONS

This research provides an Agile development model based on the Essence of software engineering. This section is part of

the process in the DSR that defines and develops artifacts. The model contains the common ground or shared understanding among the popular Agile and scaling Agile methods discussed in the previous section.

A. An Essence-based Agile Method Development Model

We develop a model for an Agile software development method based on the Essence in Fig. 5. The model was derived from the Essence methods architecture in Fig. 3. The organization can develop its own Agile methods based on the selected Agile practices. We emphasize the scope of the Agile methods using the scaling Agile methods, comprising the Agile team methods, such as Scrum, Kanban, and extreme programming. For example, the organization develops its practices by adopting the practices such as Scrum practice, Spotify practices for the organization, and Kanban for the workflow. They also can adopt the Scrum of Scrums practice for coordination. The organization can also implement some practices from Extreme Programming for pair programming and code ownership.

B. The Common Ground of Scaling Agile Methods

We establish a common ground for scaling Agile methods, as scaling Agile methods already include Agile methods at the team level, such as Scrum, Kanban, and Extreme Programming, which are widely used by organizations. This study proposed the common ground in Fig. 6, which is classified by principles, process, work product, organization, and implementation approach.

At the minimum level, the scaling Agile team only contains the Scrum and integration teams. LeSS, Nexus, Scrum of Scrum have similar project team structures with it. Nexus has the following roles: The Nexus Integration Team Member, Nexus Integration Team Scrum Master, Nexus Integration Team Product Owner, and Nexus Integration Team itself [31].

The common ground of scaling the Agile method contains the Agile practice. It refers to the level of Agile implementation [7]. Team Agile is the Agile practice and method at the team level, such as Scrum, XP, Kanban, FDD, and DSDM. Enterprise agility contains a larger Agile team in the organization collaboratively together to develop a single large product. On the other hand, business agility embraces an Agile mindset and principles that encompass all areas, extending beyond product development to encompass domains like personnel management, servant leadership, design of organization, and financial planning [13]. This study refers to the previous research and literature on the general practice of Agile [7, 32, 33, 34].

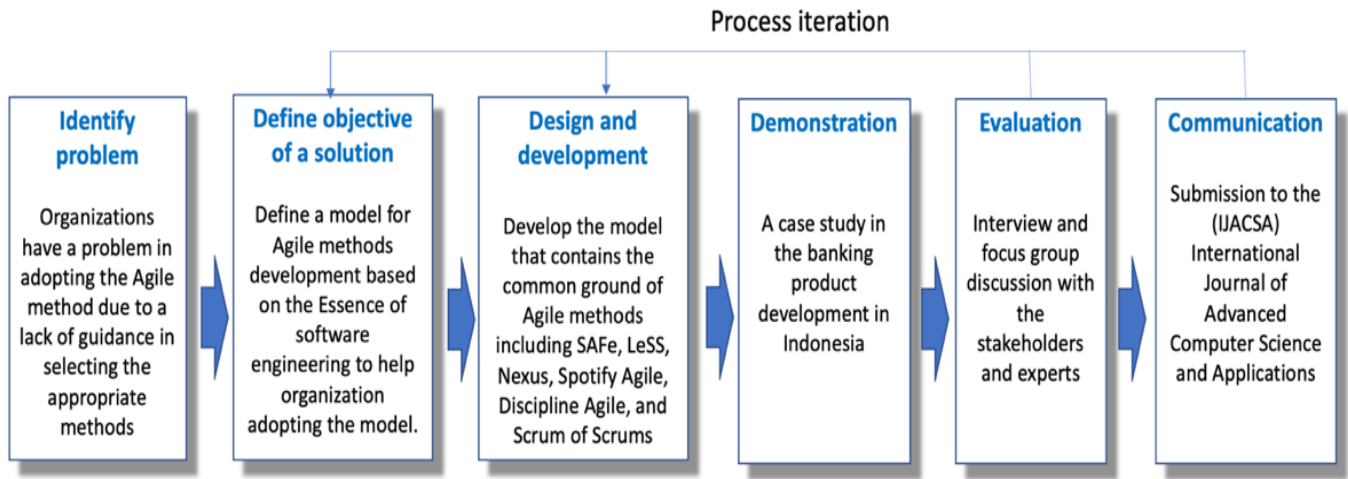


Fig. 4. The DSR's research methodology [28].

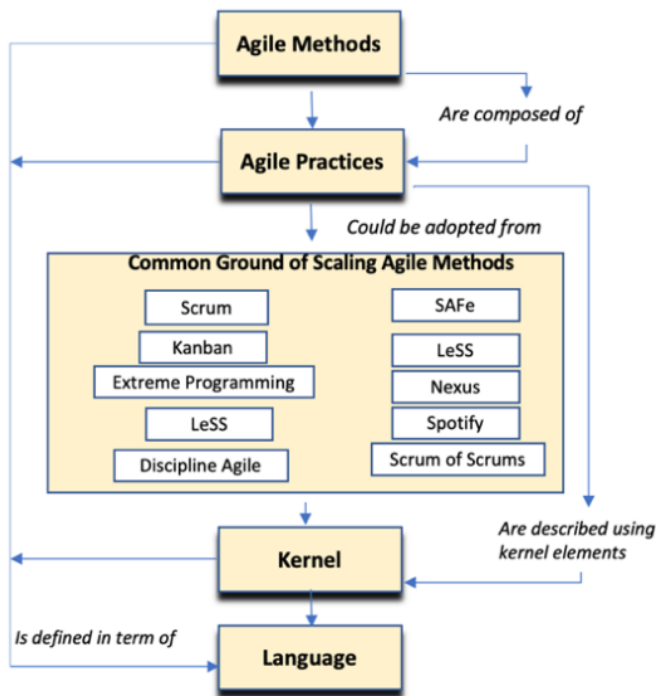


Fig. 5. The DSR's research methodology [28].

Table I summarizes the common ground for popular scaling Agile methods, SAFe, LeSS, DAD, Nexus, Spotify, and Scrum of Scrums. We classify the practices based on the common ground: principle, process, organization, work product, and implementation approach.



Fig. 6. The high level of the common ground scaling agile method.

C. The Guidance to use the Model

An organization can implement the Agile method using the guidance in Fig. 7 adopted from [35]. The model, built on current theories and practices, allows practitioners to enhance their work by adopting newer practices. The common ground is a guideline for selecting suitable scaling Agile tools and practices. It can also be expanded as needed for project implementation.

TABLE I. THE SUMMARY OF COMMON GROUND FOR SCALING AGILE METHODS

SAFe	LeSS	DAD	Nexus	Spotify	Scrum of Scrums
Principles All methods follow the Agile Manifesto for mindset, values, and principles. SAFe, LeSS, and DAD enhance Agile Manifesto to their methods based on the scaling Agile need. Other methods don't specifically define their principles.					
Mindset, values, principles	Principles	Principles	Follow the Agile manifesto and Scrum value	Follow the Agile manifesto	Follow the Agile manifesto and Scrum value
Process All methods have their process. They utilize Scrum for the Agile team. SAFe use Scrum, Kanban, DevOps, and XP. LeSS, Nexus, Spotify, and Scrum of Scrum mainly utilize Scrum for their methods. DAD utilizes Scrum but incorporates more Agile methods.					
Agile product delivery, enterprise solution delivery	The first method is for up to eight teams, and the second is for up to a few thousand people on one product.	Disciplined Agile Delivery as the bottom process, Disciplined DevOps, Disciplined Agile IT (DA, and Disciplined Agile Enterprise (DAE).	Following the Scrum framework with the integrated elements	Following the Scrum and enhancing the methods	Scrum, Scrum of Scrum
Organization Scrum, coordination, and support teams are the minimal team structure for scaling the Agile method. Each method has the predefined team structure					
Agile Team, Product and Solution team, and Business Owner. Agile team; Teams of Agile team: business, product management, hardware, software, quality, testing, compliance, operation, security. They operate within the context of Agile Release Train	LeSS Structure contains Teams, Scrum Master, Feature teams, organizing by customer value, organization structure, and communities.	Primary Roles: Team Lead, Product Owner, Team Member, Architectural Owner, Stakeholder. Supporting roles: Specialist, Independent Tester, Domain Expert, Technical Expert, Integrator.	Nexus roles: Nexus Integration Team, Product Owner in the Nexus Integration Team, Scrum Master in the Nexus Integration Team, Nexus Integration Team Members.	Squads, Chapters, Tribes, and Guilds.	Scrum of Scrum
Business Requirements and Work Product Each Agile team develops a single work product. The whole team produces the integrated work product.					
Program increment	Product backlog, Potentially Shippable Product Increment	Release solution	Product Backlog, Integrated Increment	Work product	Work product
Implementation approach The implementation scope starts from the single project up to the enterprise level.					
Essential SAFe, Large Solution, Portfolio, Full	LeSS, Less Huge	DAD as the bottom process, Disciplined DevOps, DAIT, and DAE.	Single product development	Single product development	Single product development

V. CASE STUDY

The case study is a part of the demonstration process in the DSR. It was conducted at a national bank in Indonesia. It was actively engaged in a digital transformation initiative to develop a business ecosystem and a digital ecosystem framework based on digital technologies (refer to Table II). The project charter planning document expressed a desire for the application to drive an increase in the Current Account Saving Account (CASA) by USD 1.3 billion. However, despite observing a rise in transaction volume from 2020 to 2021, the cumulative transaction volume only amounted to USD 282,000 by July 2021. Consequently, this discrepancy posed a significant challenge as the target in CASA through application implementation was not attained.

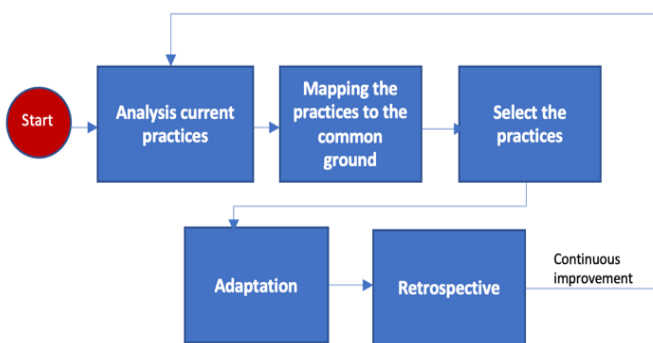


Fig. 7. The guidance for implementation.

TABLE II. ADOPTED PRACTICES FOR THE CASE STUDY

The Common Ground	Current Practices	Proposed Practice
Principles	<ul style="list-style-type: none"> - Currently, the company already has a procedure guide in the form of a Circular of Information Technology Workflow Procedures, which regulates the Agile methodology 	<ul style="list-style-type: none"> - Apply and explore the principle of SAFe and LeSS for organizations (SAFe dan LeSS) - Spotify - squad, tribes, chapter, guild
Organization	Roles: Tribe Sponsor Product, Tribe Sponsor Technology, Tribe Leader Product, Tribe Leader Technology, Product Owner, Scrum Master, Programmer, Quality Control/Tester, Business Analyst, System Analyst, QA Analyst, Document Writer, Application Security Engineer, Manajemen Risiko, Operation Engineer, Subject Matter Expert (UI/UX), Subject Matter Expert (Data Analyst), Subject Matter Expert (Arsitektur IT)	<ul style="list-style-type: none"> - Adopted SAFe-scaled Agile principles (SAFe) - Implementing the role of a quad, tribes, chapter, and guild as a whole in the Agile process (Spotify) - Implementation of a team based on the Whole Team (XP)
Business Requirement	<ul style="list-style-type: none"> - Business Requirements are explained in the Product Roadmap activity - Development needs are stated in the Business Model Canvas (BMC) document - Furthermore, BMC is discussed in the Discovery Session activity 	<ul style="list-style-type: none"> - Application of product backlog understanding to the entire team through Program Increment Planning (SAFe) - Adopt product backlog from Scrum, Sprint Planning, Definition of Ready, and Definition of Done - Using visualize tools
Process	<ul style="list-style-type: none"> - Currently, the scrum practice is not fully implemented, which is only sprints and scripts - Other scrum practices, such as sprint review and sprint retrospective, are not implemented 	<ul style="list-style-type: none"> - The discipline of doing Daily Scrum - Implementation of sprint backlog activity, output backlog refinement - Adopt Joint Product-level Sprint Review and Joint Retrospective
Implementation Approach	<ul style="list-style-type: none"> - Project level, Tribe leader product, and tribe leader technology actively coordinate - Program level, The development team does not focus only on developing one application but parallel developing other applications - Business level, The business team doesn't focus on just one product developer 	Adopt LeSS Practices <ul style="list-style-type: none"> - Implementation of a team-based organization - Cross-functional theme - Not implementing the "Resource-Allocation" phase

A. Analysis of Current Condition

From the interview analysis and source documents, the researchers focused on a root problem in Agile software development projects: the application of Agile practices that were not fully implemented. The organization needed to develop its practices to improve the software development delivery process. Current practices were not appropriately defined.

We adopted the Essence-based Agile Model (Fig. 5) and mapped its components. They included common ground, principles, organization, business requirements, process, work products, and implementation approach. For the principles, Agile provisions had not been fully implemented following the procedure guidelines. There were vital responsibilities, performance indicators, and authority on the Agile team which had not been implemented. There was an Agile implementation process that did not use supporting tools, namely Jira and/or Confluence. There were Scrum practices that were not implemented.

B. Proposed Practices

As per the Essence guidance to develop the team's methods, we selected the proper practices from the popular Agile methods, such as Scrum, SAFe, Nexus, and LeSS, for our methods. We mapped the nominated practices to the common ground component.

VI. EVALUATION

This section describes the model's validation process and the case study's implementation. It is part of the evaluation process in the DSR. The evaluation process was carried out on the practices proposed in the organization to ensure that these practices would be suitable for implementation in a methodology. The evaluation involved 16 internal and external stakeholders through a process of Focus Group Discussion (FGD) or interviews, as well as filling out questionnaires. The questionnaire used a Likert scale with five measurements: (1) Strongly Disagree (2) Disagree (3) Neutral (4) Agree (5) Strongly Agree.

A. Internal Evaluation

We involved eleven Agile team members: a leader, a scrum master, and nine team members. In the internal evaluation process, one-to-one interviews were conducted with the team leader. Besides, a questionnaire was filled out by all Agile team members regarding appropriate practices. Each proposed practice was asked for confirmation using a Likert scale. The question was, "Are the principles of practice <practice name> applicable?". The results of distributing the questionnaires are shown in Fig. 8.

B. External Evaluation

Five experts with experience in Agile implementation participated in the external evaluation using a one-to-one interview and filling out a questionnaire. The experts' profiles participating in the expert judgment process are described in Table III.

The results of internal and external evaluations were carried out by analyzing the interviews and questionnaires. The transcripts from interview data were analyzed using the Dedoose tool, while the Likert scale questionnaire data were analyzed with Microsoft Excel.

C. Internal Questionnaire Data Analysis and Processing

Results from the questionnaires distributed to the Agile team and the experts are demonstrated in Table IV. An average calculation was carried out, and the overall evaluation value was obtained. It was agreed that the practice value was above 3 to apply.

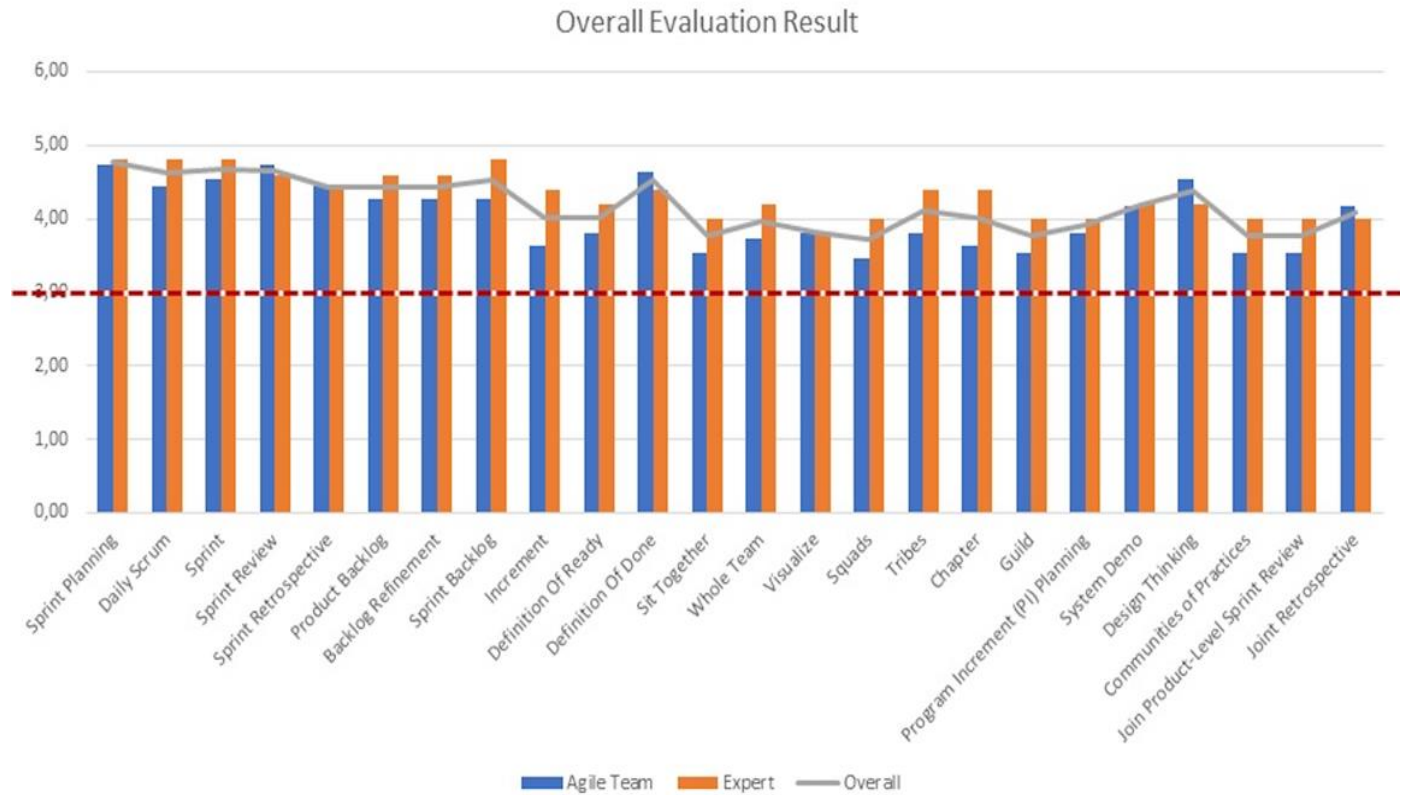


Fig. 8. Evaluation result.

TABLE III. THE PROFILES OF THE EXPERTS

Code	Profession	Experience	Specialization	Certification
N1	Consultant	25 Years	Agile Software Development, Agile DevOps, Project Management, IT Governance	COBIT 5 Foundation certification, Certified Information Systems Auditor, Scrum Master, Professional Agile Coaching, Certified DevOps Foundation, ITIL Foundation - Intermediate Banking Risk Management Certified Level 1-3, dll
N2	Consultant	17 Years	Project Management, Agile Software Development, DevOps	Certified DevOps, ITIL, Certified Agile Coach, Scaled Agile (SAFe) Agilist v5.1, SCRUM Master (CSM), Certified SCRUM Professional (CSP), Certified Kanban System Design (KMP-1), COBIT 5 Foundation, dll
N3	Senior Manager	17 Years	Project Management, Agile Software Development, Business Analyst	Scrum Master Certified (SMC), Project Management Professional (PMP)
N4	Assistant Manager	9 Years	Agile DevOps, Quality Assurance, Agile Software Development	Scrum Master, DevOps Foundation, Quality Management System (ISO 9001:2015)
N5	Senior Manager	13 Years	Quality Assurance, Agile Software Development, Agile DevOps	Certified Agile Tester By International Software Quality Institute (ISTQB) Certified Data Management Professional (CDMP)

TABLE IV. THE QUESTIONNAIRES RESULTS

No	Selected Practice	Agile Team	Experts	Overall
1.	<i>Sprint Planning</i>	4,73	4,75	4,74
2.	<i>Daily Scrum</i>	4,45	4,75	4,60
3.	<i>Sprint</i>	4,55	4,75	4,65
4.	<i>Sprint Review</i>	4,73	4,75	4,74
5.	<i>Sprint Retrospective</i>	4,45	4,5	4,48
6.	<i>Product Backlog</i>	4,27	4,75	4,51
7.	<i>Backlog Refinement</i>	4,27	4,75	4,51
8.	<i>Sprint Backlog</i>	4,27	4,75	4,51
9.	<i>Increment</i>	3,64	4,5	4,07
10.	<i>Definition Of Ready</i>	3,82	4,25	4,03
11.	<i>Definition Of Done</i>	4,64	4,5	4,57
12.	<i>Sit Together</i>	3,55	4,25	3,90
13.	<i>Whole Team</i>	3,73	4,5	4,11
14.	<i>Visualize</i>	3,82	4	3,91
15.	<i>Squads</i>	3,45	4	3,73
16.	<i>Tribes</i>	3,82	4,5	4,16
17.	<i>Chapter</i>	3,64	4,5	4,07
18.	<i>Guild</i>	3,55	4	3,77
19.	<i>Program Increment (PI) Planning</i>	3,82	4,25	4,03
20.	<i>System Demo</i>	4,18	4,5	4,34
21.	<i>Design Thinking</i>	4,55	4,5	4,52
22.	<i>Communities of Practices</i>	3,55	4,25	3,90
23.	<i>Join Product-Level Sprint Review</i>	3,55	4,25	3,90
24.	<i>Joint Retrospective</i>	4,18	4,25	4,22

D. Internal Analysis and Processing of Interview Data

Interview data was processed by compiling transcripts of recorded interviews or FGDs with the Agile team and one-to-one interviews with experts into a text file. Files containing interview transcripts were uploaded to the Dedoose software. Next, a coding process was carried out for the 24 elections by marking the code from the results of the interview transcription in the relevant section. Excerpts of the codes revealed practices and methods mostly emerge from

interviews. Next, a new Software Development Methodology Design was created from the selected practices, proposed practices, and methods.

The selection of these practices has been adapted through one product development sprint cycle, with the outputs of each stage described in Table V. A retrospective process was continued to review the practices and make continuous improvements. Therefore, the team can develop even better methods.

TABLE V. PRODUCT DEVELOPMENT CYCLE AND PRACTICES

No	Phase	Practice	Origin Method	Output
1.	Agile Discovery Session	Program Increment Planning	SAFe	Business Model Canvas (BMC), Product Backlog
		Product Backlog	Scrum	
		Visualize	Kanban	
2.	Agile Sprint Planning	Sprint Planning	Scrum	Sprint Backlog, Kanban board
		Definition of Ready	Scrum	
		Definition of Done	Scrum	
		Visualize	Kanban	
3.	Agile Sprint	Sprint	Scrum	Increment (Product/Function) SDLC Documents(Unit Test (UT), System Integration Test (SIT), Code Review, Apps Security Test, Migration (MIG), Deployment (DEP), Deployment verification, User Manual, Pentest)
		Daily Scrum	Scrum	
		Product Backlog	Scrum	
		Backlog Refinement	Scrum	
		Sprint Backlog	Scrum	
		Sit Together	Scrum	
		Whole Team	XP	
		Squads	Spotify	
		Tribes	Spotify	
		Chapter	Spotify	
		Guild	Spotify	
4.	Agile Review & Retrospective	Sprint Review	Scrum	Retrospective Documents
		Sprint Retrospective	Scrum	
		System Demo	SAFe	
		Joint Retrospective	LeSS	
Proposed Practice				
5.	Agile Sprint	Clean Code	LeSS	Increment
		Unit Testing	LeSS	
		Test Automation	LeSS	
		Test Driven Development	LeSS	
		End To End Testing	Scrum	
Proposed Method				
6.	Agile Discovery Session		Design Thinking	Design Thinking Report
7.	During coordination		Scrum of Scrums	Sprint Backlog

VII. CONCLUSION

This study aims to develop a model for Agile methods based on the Essence of software engineering framework. The resulting model is a significant outcome of this research, as it encompasses the common ground or shared understanding of scaling Agile practices from well-known methods such as SAFe, LeSS, Nexus, Disciplined Agile, Spotify Agile, and Scrum of Scrums. The intended scope of this Agile method is up to scaling Agile, as it already encompasses the Agile method within it. The common ground includes the principle, business requirement, process, work product, organization, and implementation approach. By following this model, organizations can benefit from valuable guidance in developing their Agile methods, as outlined below:

- Organization can define their initial practices from the current Agile methods
- The common ground can classify the practices to group the practices for more manageable to maintain
- The sample implementation was demonstrated in the case study

The model was implemented in the Agile product development in a national bank in Indonesia. Agile experts participated in reviewing the model. The selected practices, as based on the model, were evaluated during the focus group discussion and interview with the relevant stakeholder in the case study. The model of this study and the common ground can be implemented in the other organization to develop their methods in their Agile development project.

This study has significant implications for both academic research and practical applications. In terms of academic research, it introduces new literature concerning the scaling of Agile methods. A notable contribution is the Essence-based model, which establishes a shared foundation in the realm of scaling Agile. For practitioners, this study enhances their understanding through relevant case studies. The model can serve as a valuable reference for practitioners to make informed decisions, customize, and adapt Agile methods to suit their organizational needs.

Further research can be explored to enhance the model into a more detailed framework. Another study to build the framework can be performed with extensive expert judgment and more case studies. The framework can include more practices and specifications as supporting references.

As the model was derived from the content analysis from relevant references, it may have limitations in actual practices. The grounded theory may complement this model. Other studies in the same field may have different perspectives for building the model.

ACKNOWLEDGMENT

We appreciate the research support provided by the Faculty of Computer Science, Universitas Indonesia, by funding this study through the PUTI 2020 program under grant contract number NKB-559/UN2.RST/HKP.05.00/2020.

REFERENCES

- [1] PMI. 2017. PMI's PULSE of the PROFESSION 9TH Global Project Management Survey
- [2] McKinsey&Company. 2018. Agile compendium. October (2018)
- [3] D. Rigby, J. Sutherland, and A. Noble. 2018. Change management - Agile at Scale. Retrieved from <https://hbr.org/2018/05/Agile-at-scale>
- [4] B. Davis. 2013. Agile Practices for Waterfall Projects Shifting Processes for Competitive Advantage. J.Ross Publishing
- [5] Scrum.org. 2019. Scrum Master Trends. (2019)
- [6] Rigby, K. Darrell, J. Sutherland., and A. Noble. 2018. Agile at Scale. Harvard Bus. Rev. (2018). Retrieved from <https://hbr.org/2018/05/Agile-at-scale>
- [7] Project Management Institute. 2017. Agile Practice Guide. Project Management Institute, Inc
- [8] I. Jacobson, P.W. Ng, P.E. McMahon, I. Spence, and S. Lidman. 2012. The Essence of software engineering: The SEMAT kernel. Queue 10, 10 (2012), 40–51.
- [9] D. Car-Pusic, I. Marovic, and G. Bulatovic. 2020. Development of a Hybrid Agile Management Model in Local Self-Government Units. Technical Gazette. <https://doi.org/10.17559/TV-20190205140719>
- [10] S. M. R.Masud, M. Masnun, and M. A. Sultana, "DevOps Enabled Agile: Combining Agile and DevOps Methodologies for Software Development", International Journal of Advanced Computer Science and Applications (IJACSA), Volume 13, No. 11, 2022.
- [11] VersionOne. 2019. The 13th annual STATE OF AGILE Report - 2018. CollabNet | VersionOne 13, (2019)
- [12] Scrumguide.org. 2017. The Scrum Guide. Retrieved from <https://www.scrumguides.org/scrum-guide.html>
- [13] K. Back. 2005. Extreme Programming Explained Embrace Change. Addison-Wesley Professional.
- [14] D. Rose. 2018. Enterprise Agility. Jon Wiley & Sons, Inc.
- [15] D. Leffingwell. 2020. Scaled Agile Framework. Retrieved from <https://www.scaledagileframework.com>
- [16] Project Management Institute. 2023. Disciplined Agile. Retrieved from <https://www.pmi.org/disciplined-agile>
- [17] M. Alqudah and R. Razali. 2016. A review of scaling agile methods in large software development. Int. J. Adv. Sci. Eng. Inf. Technol. 6, 6 (2016), 828–837. DOI=<https://doi.org/10.18517/ijaseit.6.6.1374>
- [18] M. Yilmaz and Connor. 2016. A Scrumban integrated gamification approach to guide software process improvement: a Turkish case study. Technical Gazette. DOI: 10.17559/TV-20140922220409
- [19] T. Dyba and T. Dingsoyr. 2015. Agile Project Management: From Self-Managing Teams to Large-Scale Development. Proc. - Int. Conf. Softw. Eng. 2, (2015), 945–946. DOI=<https://doi.org/10.1109/ICSE.2015.299>
- [20] M.M. Jha, R.M.F. Vilardell, R.M.F and J, Narayan. 2016. Scaling agile scrum software development: Providing agility and quality to platform development by reducing time to market. Proc. - 11th IEEE Int. Conf. Glob. Softw. Eng. ICGSE 2016 (2016), 84–88. DOI=<https://doi.org/10.1109/ICGSE.2016.24>
- [21] M. Laanti and M, Kangas. 2015. Is Agile Portfolio Management Following the Principles of Large-Scale Agile? Case Study in Finnish Broadcasting Company Yle. Proc. - 2015 Agil. Conf. Agil. 2015 (2015), 92–96. DOI=<https://doi.org/10.1109/Agile.2015.9>
- [22] M. Paasivaara and C. Lassenius. 2011. Scaling scrum in a large distributed project. Int. Symp. Empir. Softw. Eng. Meas. (2011), 363–367. DOI=<https://doi.org/10.1109/esem.2011.49>
- [23] T. Dingsøy, K. Rolland, N.B. Moe, and E.A. Seim. 2017. Coordination in multi-team programs: An investigation of the group mode in large-scale agile software development. Procedia Comput. Sci. 121, (2017), 123–128. DOI=<https://doi.org/10.1016/j.procs.2017.11.017>
- [24] OMG. 2018. kernel and Language for Software Engineering Methods (Essence). Object Manag. Gr. Versión 1.2 (2018), 300. Retrieved from <https://www.omg.org/spec/Essence/1.2>
- [25] Semat.org. 2020. Software Engineering Methods and Theory. Retrieved from [semat.org](https://www.semat.org)
- [26] S. Theobald, A. Schmitt, and P. Diebold. 2019. Comparing Scaling Agile Frameworks Based on Underlying Practices. Springer International Publishing. DOI=<https://doi.org/10.1007/978-3-030-30126-2>
- [27] M. K. Alqudah, R. Razali and M. K. Alqudah, "Agile Methods Selection Model: A Grounded Theory Study" International Journal of Advanced Computer Science and Applications (IJACSA), Volume 10, No. 7, 2019.
- [28] K. Peffers, T. Tuunanen, Rothenberger, M.A., and Chatterjee, S. 2007. A design science research methodology for information systems research. Journal of Management Information Systems 24, 45–77. DOI=<https://doi.org/10.2753/MIS0742-1222240302>
- [29] R. Almeida, J.M. Teixeira, M. Mira da Silva, and P. Faroleiro. 2019. A conceptual model for enterprise risk management. J. Enterp. Inf. Manag. 32, 5 (2019), 843–868. DOI=<https://doi.org/10.1108/JEIM-05-2018-0097>
- [30] A.C. Amorim, M. Mira da Silva, R. Pereira, and M. Gonçalves. 2020. Using agile methodologies for adopting COBIT. Inf. Syst. xxxx (2020), 101496. DOI=<https://doi.org/10.1016/j.is.2020.101496>
- [31] Scrum.org. 2023. Nexus Guide. Retrieved from <https://www.scrum.org/resources/nexus-guide>
- [32] Axelos. 2018. Prince2 Agile. The Stationery Office Ltd. Kindle Edition., United Kingdom for The Stationery Office. Retrieved from [axelos.com](https://www.axelos.com)
- [33] Goodpasture, J. 2016. Project Management the Agile Way Making it Work in the Enterprise SECOND EDITION. J. Ross Publishing.
- [34] Project Management Institute. 2021. A Guide to the Project Management Body of Knowledge (PMBOK® Guide) (7th ed.). Project Management Institute.
- [35] T.J. Gandomani and M.Z. Nafchi. 2015. An empirically-developed framework for Agile transition and adoption: A Grounded Theory approach. J. Syst. Softw. 107, (2015), 204–219. DOI=<https://doi.org/10.1016/j.jss.2015.06.006>