Scaled Agile Process Improvement Recommendations with CMMI 2-Based Agile Scaling Model: A Case Study of the Indonesia National Single Window Agency

I Made Aditya Pradnyadipa Mustika, Betty Purwandari, Alex Ferdinansyah
Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia

***Abstract*—The Indonesia National Single Window System (SINSW) is a platform developed by the Indonesia National Single Window Agency (LNSW) using Agile methodology, specifically the Scrum framework. Several challenges were identified during its development, including deviations from the Scrum Guide and the absence of formal, regular events necessary for effective team coordination and alignment. These issues revealed gaps in Scrum implementation and broader difficulties associated with scaling Agile practices beyond the team level. Therefore, this study applied a scaling Agile model based on Capability Maturity Model Integration (CMMI) 2 to evaluate the agency's existing Agile process and recommend targeted improvements to Agile practices. The evaluation involved qualitative interviews with key stakeholders, including the Project Management Officer, System Analyst, and developers. The interviews were subsequently quantified using the Key Process Area (KPA) rating framework. The findings led to actionable recommendations to optimize the Agile process, improve team collaboration, and support SINSW's success.**

***Keywords*—Scaling Agile; CMMI; Scrum; KPA rating; software engineering**

I. Introduction

Recently, digital transformation has significantly influenced government administrative processes, particularly in customs services. A Single Window System is an integrated digital platform that allows businesses, traders, and government agencies to submit and process information through a single entry point. It is widely used in trade facilitation, customs clearance, and regulatory approvals to streamline processes, reduce paperwork, and improve efficiency [1]. Indonesia's single window system is called the Indonesia National Single Window System (SINSW). The management and operation of SINSW are regulated by the Minister of Finance [2], who assigns the Indonesia National Single Window Agency (LNSW) the responsibility for system governance and development.

SINSW serves as a critical, unified web-based integration hub that streamlines national trade and logistics. The operational scale of this environment is significant, as it consolidates 42 distinct application systems, known as Single Submission modules. These modules facilitate data exchange and synchronization across 18 different Government Ministries and Agencies. To manage this complexity, the development effort is distributed among six dedicated development teams working concurrently to ensure a seamless "single-entry point" for all stakeholders. This multi-agency architecture creates a high-stakes development environment where technical alignment and cross-team synchronization are paramount to maintaining the integrity of Indonesia's digital trade infrastructure.

Despite its strategic importance, the implementation of SINSW faces challenges in meeting development goals, particularly in fulfilling its sprint goals. Although the implementation is imperfect, LNSW utilizes Scrum to manage SINSW development projects. Several issues hinder the full realization of Agile principles, including the lack of structured coordination events and inconsistencies in Scrum implementation. Reports from the second semester of 2023, specifically across eight observation sprints [3], indicate that many sprint goals were not met, with unfinished tasks carried over into subsequent sprints, delaying system development and deployment.

The issue of coordination and alignment between teams in SINSW development underscores challenges beyond team-level Agile implementation. The issue highlights the intricacies of scaling Agile practices in a large, multi-stakeholder environment. Agile methodologies at the team level emphasize iterative development, flexibility, and responsiveness. However, scaling Agile across multiple teams introduces organizational dependencies, workflow synchronization challenges, and cross-functional misalignment [4], [5]. The complexities often lead teams to focus on their deliverables rather than collaborate effectively with other teams. Moreover, silos are one of the key inhibitors of Agile transformation at scale [6].

In the specific context of LNSW, these theoretical challenges manifest through several critical processes. General scaling misalignment is evidenced here by inconsistent Standard Operating Procedures and a lack of uniform process assets. Technical coordination is frequently compromised by ad-hoc synchronization of dependencies. The absence of documented Success Criteria also complicates the identification of architectural issues and diminishes the effectiveness of Agile workflows. Furthermore, technical estimation remains limited to time-based metrics that fail to account for resource availability and complexity.

This study aims to evaluate the current scaling of Agile implementation in SINSW development, identify key bottlenecks, and propose improvements to enhance Agile implementation in large-scale government digital transformation projects. This research will examine two primary problems. The first is the absence of formal and regular cross-team coordination events, which hinders alignment and efficiency; the second is the deviation from Scrum best practices, which diminishes the effectiveness of Agile workflows. Ignoring best practices in the scrum process can lead to increased risks, poor team collaboration, ineffective communication, and suboptimal product delivery. Ultimately, it hampers project success, reduces stakeholder satisfaction, and may lead to higher costs and longer development timelines [7]. To address these challenges, this study seeks to answer the following research questions:

RQ1: What are the evaluation results of the Scaled Agile process in the SINSW development?

RQ2: What recommendations can be proposed to improve the Scaled Agile process in the SINSW development?

This research improves Agile implementation for large-scale, multi-stakeholder government projects through a unique Indonesian case study. By using the Capability Maturity Model Integration (CMMI) 2-Based Agile Scaling Model to quantify gaps, the study provides a validated, phased roadmap for digital transformation rather than generic advice. Ultimately, this research aims to assist the LNSW in achieving greater agility and efficiency in managing Indonesia's digital trade facilitation infrastructure, thereby contributing to the broader goal of streamlining public service delivery through technology-driven governance.

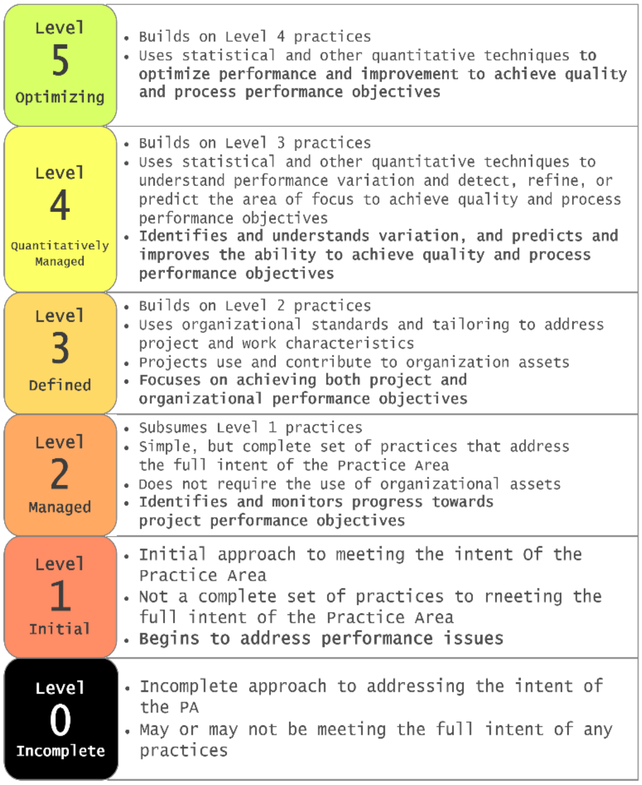
II. Literature Study

The primary challenge in digital transformation is balancing Agile flexibility with rigorous oversight. Ferdinansyah and Purwandari [8] argue that while Agile provides adaptability, models like CMMI are essential for an organization to minimize risk and ensure process stability. Raharjo and Purwandari [9] further classify these challenges through the lens of PMBOK, identifying stakeholder management as a critical governance hurdle.

As organizations transition from individual teams to a broader scaled environment, these challenges become significantly more complex. Managing dependencies across multiple teams requires specialized coordination that goes beyond standard Scrum ceremonies. Conboy and Carroll [16] identify nine core challenges in large-scale frameworks, noting that communication breakdowns and rigid procedures are frequent obstacles. These challenges often worsen in environments with high architectural coupling, where a change in one software module can trigger unexpected failures in others. Consequently, Dikert et al. [17] emphasize that success depends on how well development teams synchronize with other units, such as Quality Assurance and Operations. This synchronization ensures that integrated modules meet shared quality standards and security protocols before deployment.

The organizational structure of a transformation determines its long-term sustainability and scalability. Sońta-Drączkowska and Krogulec [18] suggest that managerial tensions at the organizational, top management, and team levels must be resolved through clear structural alignment. This alignment is essential because large organizations frequently experience a cultural conflict between traditional top-down hierarchies and decentralized Agile decision-making. Furthermore, Fissalma et al. [19] demonstrate that, in the Indonesian context, localized or customized frameworks are often more effective than commercial solutions such as SAFe or LeSS. This effectiveness stems from the framework's ability to account for local regulatory requirements and specific organizational cultures within Indonesian agencies.

To provide the necessary stability and minimize risk within these complex structures, models like CMMI are employed. CMMI remains a leading framework for software process improvement [10], [11]. Since the release of version 2.0 in 2018, it has formally integrated Agile principles to ensure that standardized workflows enhance, rather than hinder, flexibility and responsiveness [12], [13], [14]. The CMMI capability levels range from Level 0 (Incomplete) to Level 5 (Optimizing), as shown in Fig. 1. Using the CMMI 2.0 framework helps organizations strengthen their processes and scale their agile practices, with a focus on performance [15]. To operationalize this for the LNSW, the CMMI 2-Based Scaling Agile Model is utilized, which leverages the Essence Language Key to define 29 essential Scaled Agile elements. This model provides a structured benchmark for assessing the capability of Scaled Agile adoption within the organization [16]. Rather than addressing the entire framework, this study focuses on high-impact elements specifically chosen to target the issues identified within LNSW's Agile implementation. The relationship between these selected elements and their corresponding CMMI 2 Practice Areas is shown in Table I.



Level 5 Optimizing
- Builds on Level 4 practices
- Uses statistical and other quantitative techniques **to optimize performance and improvement to achieve quality and process performance objectives**

Level 4 Quantitatively Managed
- Builds on Level 3 practices
- Uses statistical and other quantitative techniques to understand performance variation and detect, refine, or predict the area of focus to achieve quality and process performance objectives
- **Identifies and understands variation, and predicts and improves the ability to achieve quality and process performance objectives**

Level 3 Defined
- Builds on Level 2 practices
- Uses organizational standards and tailoring to address project and work characteristics
- Projects use and contribute to organization assets
- **Focuses on achieving both project and organizational performance objectives**

Level 2 Managed
- Subsumes Level 1 practices
- Simple, but complete set of practices that address the full intent of the Practice Area
- Does not require the use of organizational assets
- **Identifies and monitors progress towards project performance objectives**

Level 1 Initial
- Initial approach to meeting the intent of the Practice Area
- Not a complete set of practices to rneeting the full intent of the Practice Area
- **Begins to address performance issues**

Level 0 Incomplete
- Incomplete approach to addressing the intent of the PA
- May or may not be meeting the full intent of any practices

Fig. 1. CMMI 2 capability level for practices [15].

TABLE. I. RELATION OF SELECTED SCALED AGILE ESSENCE ELEMENTS BASED ON LNSW ISSUES TO CMMI 2 PRACTICE AREA [16].

Scaled Agile Essence Element	Classification	CMMI 2 Practice Area
Agile Master	Pattern	GOV, II, PCM, MPM, CAR, DAR, OT, RSK, MC
Scaled Agile Master	Pattern	GOV, II, PCM, MPM, CAR, DAR, OT, RSK, MC
Team Coordination Meeting	Activity	RSK, PCM, MC, TS, RDM
Scaled Coordination Meeting	Activity	RSK, PCM, MC, TS, PLAN, RDM
Success Criteria	Work Product	VV, PLAN, PAD, EST, RDM, DAR, MC
Team Retrospective	Activity	CAR, PR, MPM, MC, TS, RSK, PQA, DAR, PCM

III. METHODS

This research employed a case study approach focusing on the development process of SINSW at the LNSW. The study integrated mixed methods, combining qualitative and quantitative data collection to provide a comprehensive evaluation. The research analyzed real-world problems and proposed solutions to improve them [17]. The study followed a structured applied research framework [18] comprising four stages, as depicted in Fig. 2.

A. Research Definition

This stage involved two key steps: understanding the problem and identifying research questions. The process began by comprehensively understanding the issue and its surrounding context. In understanding the problem phase, researchers conducted interviews and analyzed internal documents to identify gaps between the current state and expected outcomes. These gaps helped pinpoint root causes, shaping the study's primary focus. Using a fishbone diagram, researchers formulated key research questions based on the identified root causes in the identification phase [19]. The questions defined this study’s objectives and expected contributions.

B. Research Design / Plan

The Research Design/Plan stage focuses on developing a structured research approach based on the previously identified key research questions. This stage consisted of three essential steps: choosing the design and data collection approach, inventorying resources, and developing research instruments. Each step ensured the research methodology aligned with the objectives and provided a comprehensive framework for evaluating Scaled Agile implementation.

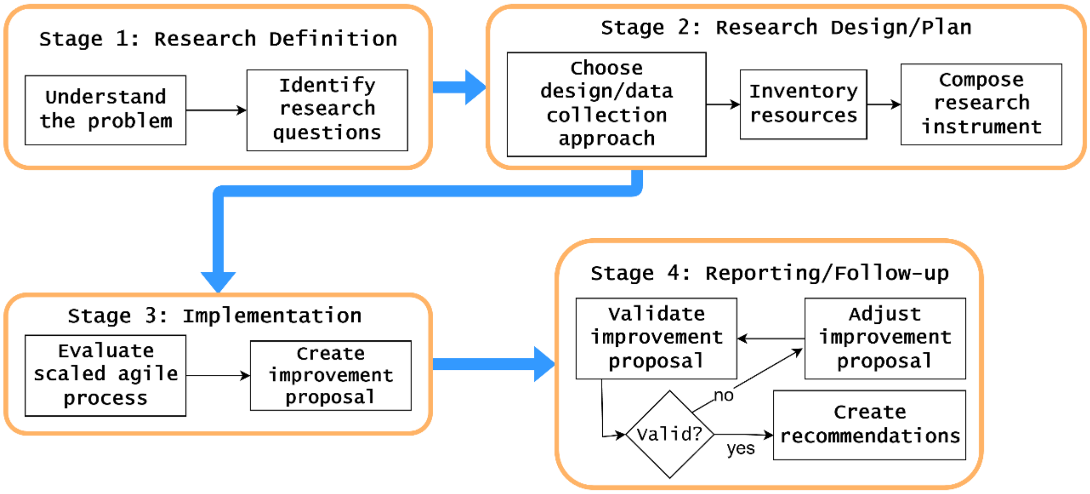

Stage 1: Research Definition
Understand the problem
Identify research questions
Stage 2: Research Design/Plan
Choose design/data collection approach
Inventory resources
Compose research instrument
Stage 3: Implementation
Evaluate scaled agile process
Create improvement proposal
Stage 4: Reporting/Follow-up
Validate improvement proposal
Adjust improvement proposal
Valid?
no
yes
Create recommendations

Fig. 2. Applied research framework [18].

The first step, choosing the design and data collection approach, involved conducting a literature review to explore previous studies and applying the Compare, Contrast, Criticize, Synthesize, and Summarize (3C+2S) analysis method. This process helped establish a theoretical framework and research methodology for subsequent steps. Through the Compare and Contrast phases, it was identified that while existing models provide frameworks for capability measurement, they often lack the specific organizational context of large-scale government agencies managing multi-stakeholder infrastructures. The Criticize phase highlighted a significant gap in prior SINSW evaluations. The gap often focuses on isolated services rather than the organization-wide coordination between all 42 integrated systems. Consequently, the Synthesize and Summarize phases led to the selection of the CMMI 2-Based Scaling Agile Model as a suitable foundation for this study.

Next, the inventorying resources integrated literature reviews and interviews to gather essential inputs for the study. The researchers combined the theoretical framework and methodology from the previous step with additional insights, including organizational expectations, key roles in the development process, and the organization's current state of Scaled Agile implementation. The output of this step was a comprehensive resource inventory listing the necessary elements for understanding the organization’s Scaled Agile implementation and identifying areas for improvement.

The final step, composing research instruments, involved refining and validating the collected resources to create practical tools for data collection. The process included defining key Scaled Agile Framework elements, formulating interview questions based on the CMMI 2 practice area, and selecting appropriate appraisal tools. It ensured that the research instruments aligned with the study’s objectives and enhanced the reliability and depth of the evaluation.

Table II lists the appraisal participants, who were purposively selected as subject-matter experts with cross-team oversight. While the sample size is small, these individuals provide a holistic view of the 42 integrated systems. Specifically, the PMO (RespA) manages governance for all six teams; the System Analyst (RespB) serves as Product Owner for two teams; and the developers (RespC, RespD) serve as Technical Leads across different units. Their collective leadership roles ensure a representative analysis of systemic gaps within the LNSW.

Meanwhile, Table III outlines the key focus issues in the Scaled Agile implementation at the LNSW and maps them to their respective Scaled Agile essence elements. These identified issues primarily relate to process coordination and organizational structure rather than direct technical or deployment failures. While these procedural gaps can create coordination overhead, they do not indicate a lack of technical capability or a failure in the underlying deployment infrastructure.

C. Implementation

The Implementation stage consists of two key steps derived from the Research Design phase: evaluating the Scaled Agile process and formulating improvement recommendations. This stage aimed to assess the current state of Agile implementation within the organization and develop strategies to enhance its effectiveness in line with industry’s best practices.

The first step involved evaluating the Scaled Agile process using the CMMI 2-Based Scaled Agile Model [16]. Then, the evaluation results were quantified using the Key Process Area (KPA) Rating to assess the Agile implementation's capability and identify areas for improvement [20]. This structured approach utilizes a questionnaire to measure the implementation of Agile practices, with the final capability level determined by a percentage-based Eq. (1):

$$KPA\ Rating = \frac{\Sigma(Y_n) + \frac{1}{2}\Sigma(P_n)}{\Sigma(T_n) - \Sigma(NA_n)} \times 100 \qquad (1)$$

where, each notation is explained as follows:

- Yn: Number of "Yes" responses (Practice has been fully implemented)
- Pn: Number of "Partially" responses (Practice has been partially implemented)
- Tn: Total number of questions
- NAn: Number of "Not Applicable" responses (Practice is not relevant)

By applying this equation, the organization's adoption of Scaled Agile practices was mapped to specific capability levels based on the percentage ranges shown in Table IV. This evaluation provided a benchmark for assessing how well the organization had optimized its methodologies and pinpointed key gaps or inefficiencies for future improvement.

Based on these findings, the next step was to formulate improvement recommendations to align Agile adoption with the organization’s expectations and long-term goals. These recommendations were derived from an extensive literature review, with CMMI-2 as the primary reference to ensure a structured, systematic approach to process enhancement. Additionally, best practices from popular Scaled Agile frameworks and models helped refine the proposed solutions.

TABLE. II. PROFILE OF THE RESOURCE PERSON FOR THE APPRAISAL PROCESS.

Code	Position/Role
RespA	Section chief of Information Systems Governance and Program Management, also acts as the SINSW Project Management Officer
RespB	Section chief of Information System Design and Evaluation, also acts as the SINSW System Analyst
RespC & RespD	Developer 1, Developer 2.

TABLE. III. SELECTED SCALED AGILE ESSENCE ELEMENTS BASED ON FOUND ISSUES.

Issues in the Scaled Agile Process at the LNSW	Scaled Agile Essence Elements	Resource person for appraisal
The lack of formal, regular events for cross-team coordination.	Scaled Coordination Meeting	RespB, RespC, RespD
Daily Scrum events are not conducted daily.	Team Coordination Meeting	RespB, RespC, RespD
Retrospectives do not function as standalone events and are still combined with reviews.	Team Retrospective	RespB, RespC, RespD
The Scrum Master role is not well-defined, and no formal responsibility is assigned for Scrum Master duties.	Agile Master & Scaled Agile Master	RespA, RespB, RespC, RespD
The Definition of Done (DoD) for increments does not include testing.	Success Criteria	RespB, RespC, RespD

TABLE. IV. LEVEL INTERPRETATION OF KPA RATING [20]

Level	Range	Interpretation
Fully Achieved	>85% – 100%	Agile practices have been systematically implemented with strong evidence of achievement.
Largely Achieved	>50% – 85%	Agile practices have been largely implemented, although some areas may need improvement.
Partially Achieved	>15% – 50%	Agile implementation is still in its early stages, with inconsistencies and limited adoption.
Not Achieved	0% – 15%	There is insufficient evidence of Agile practices being implemented within the organization.

D. Reporting / Follow-up

The reporting/follow-up stage consists of three key steps: validating the improvement proposal, refining recommendations, and formulating conclusions and suggestions based on the final validation results. The validation process involved the Subject Matter Experts (SMEs) to ensure the proposed improvements' accuracy, relevance, and practical applicability. It was followed by LNSW stakeholders' reviews of the recommendations to assess their alignment with organizational needs and strategic decisions. This step determines feasible proposals that can be effectively integrated into the existing processes.

Following the formulation of an improvement proposal, a structured two-tier validation process was conducted to ensure the accuracy and practical applicability of the recommendations. First, a technical validation was performed with the Subject Matter Experts (SMEs) to verify that the proposed recommendation aligned with Agile best practices. Second, an organizational validation was conducted with LNSW stakeholders to ensure the recommendations remained feasible within existing organizational constraints and strategic goals. If these experts identified any gaps or inconsistencies, necessary adjustments were made, and the validation cycle was repeated. This iterative approach ensured that the proposed improvement aligned with organizational constraints, Agile best practices, and long-term scalability goals.

IV. Results and Discussion

The appraisal process involved evaluating key Scaled Agile essence elements, identifying implementation gaps, and mapping findings to CMMI 2 practice areas. Additionally, insights from the Subject Matter Experts (SMEs) and key stakeholders were incorporated to refine the evaluation results and recommendations. Based on this analysis, the study derived key findings and developed targeted recommendations to address the research questions as follows:

A. RQ1: What are the Evaluation Results of the Scaled Agile Process in the SINSW Development?

The evaluation of the Scaled Agile process in the SINSW development, conducted as part of the implementation stage, reveals significant gaps in Agile adoption, as demonstrated in Table V. The Scaled Coordination Meeting and Team Retrospective elements show a 0% achievement rate, which is caused by the fact that the events are currently not being implemented. The lack of formally scaled events forces teams to resolve dependencies and unclear tasks through ad-hoc discussions, leading to a fragmented development cycle. Furthermore, the 19.24% achievement in Team Coordination Meetings correlates with reports that daily stand-ups often focus solely on progress updates rather than proactive risk management. Consequently, the delivery cadence is frequently disrupted, as evidenced by the high volume of unfinished tasks being carried over into subsequent sprints.

The capability of role definitions also impacts operational efficiency. The Agile Master (48.15%) and Scaled Agile Master (47.92%) are only partially completed, highlighting a lack of clearly defined responsibilities and structured facilitation of Agile processes. The only element to achieve a high level of status is the Success Criteria (52.11%). However, it still falls within Capability Level 0 (CL0), indicating that the overall capability of Agile implementation remains at the lowest capability level. Without these formal structures, the organization experiences backlog growth driven by rework and a loss of institutional knowledge, as solution alternatives and issue resolutions are rarely documented. These findings suggest the need for structured governance, more explicit role definitions, and formalized Agile events to enhance scaled Agile adoption within LNSW.

TABLE. V. Scaled Agile Essence Elements Appraisal Result

Scaled Agile Essence Elements	KPA	Interpretation	Capability Level
Scaled Coordination Meeting	Level 1 - 0%	Not Achieved	**CL0**
Team Coordination Meeting	Level 1 - 19,24%	Partially Achieved	**CL0**
Team Retrospective	Level 1 - 0%	Not Achieved	**CL0**
Agile Master	Level 1 – 48,15%	Partially Achieved	**CL0**
Scaled Agile Master	Level 1 – 47,92%	Partially Achieved	**CL0**
Success Criteria	Level 1 – 52,11%	Largely Achieved	**CL0**

B. RQ2: What Recommendations can be Proposed to Improve the Scaled Agile Process in the SINSW Development?

Following appraisal and validation with SMEs and stakeholders, the proposed recommendations aim to strengthen Agile implementation through three strategic areas. First, elaborating on Agile workflows ensures consistency across the LNSW. Second, increasing understanding of Agile value fosters the effective adoption of best practices. Finally, utilizing appropriate tools streamlines processes and enhances collaboration. The recommendations address the gaps identified in the appraisal, enabling the organization to effectively align its Scaled Agile practices with industry standards.

1) Agile Workflow elaboration: The Agile workflow needs to align with the current operational conditions and cultural dynamics of the LNSW. Given the organization's compact and straightforward structure and the existing allocation of development teams, adopting the Nexus Framework was recommended as a strategic approach for Scaled Agile practices. This framework is particularly well-suited for organizations with multiple Scrum teams collaborating on a single product, as it provides a lightweight, scalable approach to managing cross-team dependencies and ensuring alignment throughout the development process [21].

The Agile Workflow Elaboration consists of three parts. First, Agile Role Mapping defines team responsibilities and accountability. Second, Agile Event Adjustment refines ceremonies, such as coordination meetings and retrospectives, to improve dependency management. Finally, the Definition of Process Assets establishes the standardized tools and templates required for consistent execution.

a) Agile role mapping: The Agile role mapping aligns existing LNSW positions with Agile roles using the CMMI 2-Based Scaled Agile Model to provide greater clarity on responsibilities. Table VI shows role mapping, illustrating how traditional roles can be successfully translated into an Agile structure. SMEs emphasize that existing structural roles do not require any changes. Still, individuals in predefined roles within LNSW can assume Agile roles without altering their formal positions within the organization's hierarchy. However, responsibilities may vary depending on their Agile role, requiring individuals to adapt their approach accordingly.

The findings suggest that Agile roles do not have to originate solely from predefined roles in the LNSW but can emerge from team members or individuals from other units with the required expertise and capabilities. The LNSW will further review this approach to determine how Agile roles can be effectively integrated, using references from the Scrum Guide [22] and the Nexus framework [23] to establish a more flexible and adaptive role distribution within the organization. This approach ensures that Agile practices can be embedded within the LNSW structure without disrupting its operational hierarchy while fostering a culture of collaboration, accountability, and continuous improvement.

b) Agile event adjustment: In Agile Event Adjustment, it is recommended that team coordination meetings be improved to enhance communication and alignment within teams. Additionally, the introduction of scaled coordination meetings and team retrospectives is essential, as these events have not yet been conducted in the LNSW. These recommendations will be discussed in greater detail in the points below:

- Scaled Coordination Meeting

Scaled Coordination Meeting is a structured coordination event designed to align multiple Agile teams working on the same product or project. This meeting ensures that cross-team dependencies, impediments, and communication gaps are effectively managed [16]. Unlike regular daily stand-ups within a single Agile team, a Scaled Coordination Meeting focuses on inter-team synchronization, helping teams identify blockers early and resolve dependencies proactively.

The meeting should not exceed 15 minutes to maintain efficiency and involve only essential participants. Ideally, attendees include representatives from each team, such as Agile Masters, Scaled Agile Masters, and Product Owners, with additional roles joining as needed. The meeting structure can follow these key points [24]:

- Team Updates – Each team briefly shares:
    - What has been completed since the last meeting?
    - What will be worked on before the next meeting?
    - Are there any issues, blockers, or dependencies that require attention?
- Dependency Management – Discuss newly discovered dependencies and strategies to resolve them.
- Issue Resolution – Escalate unresolved issues and schedule follow-up discussions as necessary.
- Priority Adjustment – Ensure that all teams remain aligned with current priorities and accommodate any priority changes.

TABLE. VI. MAPPING OF PROPOSED AGILE ROLES WITH EXISTING LNSW JOB FUNCTIONS.

Agile Role	Existing LNSW Job Function
Product Owner	System Analyst (SA)
Scaled Product Owner	Business Analyst, Business Analyst Supporting
Agile Master	Project Management Officer (PMO) or System Analyst (SA)
Scaled Agile Master	Project Management Officer (PMO)
System Architect	Enterprise Architect
Development Team	Developer, Tester

Implementing Scaled Coordination Meetings at the LNSW should be done gradually, starting with meetings as needed rather than immediately holding them daily. This approach prevents unnecessary overhead while enabling teams to benefit from structured cross-team communication. By integrating this meeting, inter-team dependencies can be identified early, minimizing the risk of delays caused by unexpected blockers. Additionally, the meeting enhances risk management by allowing teams to escalate potential impediments and collaborate on resolutions before they become critical obstacles.

SMEs highlighted the importance of keeping Scaled Coordination Meetings flexible, emphasizing that the LNSW can combine structured meetings with ad hoc discussions among teams and Product Owners. While daily meetings may not be necessary, SMEs recommend holding formal synchronization sessions weekly or biweekly involving all Scaled Product Owners and Product Owners. It ensures that coordination and dependencies across teams are well managed, allowing the LNSW to adapt to its existing organizational dynamics. However, proper documentation is crucial for these meetings to be effective, ensuring that key discussion points and resolutions are formally recorded for future reference. By balancing structured coordination with flexibility, the LNSW can improve inter-team alignment, enhance transparency, and optimize Agile execution across projects.

- Team Coordination Meeting

Team Coordination Meeting, often called stand-up, is a key Agile practice that aligns team efforts and ensures a shared understanding of daily priorities. The primary goal of this meeting is to enhance transparency, foster collaboration, and facilitate rapid issue resolution by allowing every team member to share updates, discuss progress, and raise potential blockers [22]. To maintain focus and efficiency, Team Coordination

Meetings must be time-boxed to a maximum of 15 minutes, with attendance required for all development team members and the Agile Master, who typically facilitates the session. Additional roles, like the Product Owner (PO), may join as needed. The structure of the meeting should follow these three key points [25]:

- What did you do yesterday? – Each team member briefly shares what they completed the previous day.
- What will you do today? – Team members outline their focus for the day.
- Are there any blockers or impediments? – Team members highlight issues they are facing. While blockers do not need to be resolved within this meeting, they should be documented and followed up on in a separate discussion if required.

At LNSW, Daily Meetings are not yet conducted routinely, often exceeding the intended duration and focusing primarily on progress updates rather than proactive risk management. To maximize the benefits of these meetings, it is recommended that the LNSW implement structured daily stand-ups with explicit time constraints and discussion points. It ensures that meetings remain productive and aligned with Agile principles. Implementing a well-structured coordination meeting at the LNSW can provide several key benefits. First, it allows teams to identify risks early, ensuring that challenges are recognized and addressed before they escalate. Second, it improves workflow management, keeping the team aligned with sprint goals and optimizing work progression. Finally, this event is a primary mechanism for monitoring and controlling sprint execution. It enables teams to track their commitments, ensure tasks remain on schedule, and make real-time adjustments as necessary. By establishing consistent and effective coordination meetings, the LNSW can significantly enhance team coordination, improve sprint execution, and strengthen Agile adoption across its projects.

- Team Retrospective

Team Retrospective is a crucial Agile event that allows teams to reflect on their processes at the end of each sprint. While the Sprint Review focuses on product outcomes, the Sprint Retrospective evaluates team collaboration and process improvements. This event is essential for continuous learning and adaptation. It ensures that teams do not repeat past mistakes and can consistently optimize their workflow.

Team Retrospectives provide an opportunity to explore what worked well, what did not, and what can be improved in the next sprint [25]. Team Retrospectives should involve the entire team to gain diverse insights and perspectives. According to Scrum guidelines, the event should not exceed three hours for a one-month sprint [22]. Given that LNSW follows a two-week sprint cycle, the recommended duration for retrospectives is no more than 1.5 hours. Key aspects that should be documented during a Team Retrospective include:

- Sprint Summary – Evaluating sprint conditions and whether the sprint goal was achieved.
- What Went Well? – Identifying positive aspects of the sprint that should be continued.
- What Didn't Go Well? – Addressing challenges or blockers encountered during the sprint.
- Improvement Actions/Action Items – Listing prioritized improvements and assigning accountability.
- Team Feedback – Gathering comments, suggestions, and critiques about team collaboration and sprint performance.
- Follow-up on Previous Improvements – Reviewing the past retrospective action implementation

The LNSW needs to establish a structured Retrospective process that is consistently practiced by analyzing ongoing workflows, identifying inefficiencies, and implementing corrective actions to improve team performance and product quality. Regular retrospectives allow teams to proactively address bottlenecks, reduce technical debt, and ensure alignment with organizational goals. Additionally, this event fosters a culture of transparency, accountability, and collaboration, enabling teams to refine their processes, enhance quality assurance, and strengthen risk management.

SMEs also emphasized the importance of Retrospectives in Agile implementation. They recommended that Agile Masters and Product Owners conduct their own process evaluations alongside team-wide sessions. This approach can provide a broader perspective on team dynamics. However, SMEs stressed that team awareness is critical to a successful Retrospective. It aimed to develop members' understanding of why the event is essential and to evaluate team processes proactively to improve them. By increasing awareness and participation, the LNSW can ensure that retrospectives drive meaningful change, reinforcing a culture of continuous improvement and Agile excellence.

c) Definition of process assets: To enhance Agile implementation and ensure structured, efficient, and scalable development processes, the LNSW needs to establish transparent and standardized process assets. These assets serve as guidelines and frameworks that enable teams to work cohesively, maintain quality standards, and improve collaboration. The key process assets that require definition at the LNSW include the Team Working Agreement, Definition of Ready (DoR), and Definition of Done (DoD).

- Team Working Agreement

A Team Working Agreement is a collaboratively created document that outlines the guidelines, rules, and expectations agreed upon by a team to enhance collaboration, productivity, and alignment. It serves as a framework for how the team works together and can be updated based on team consensus [26]. This agreement covers communication methods, meeting schedules, and decision-making processes, ensuring that all team members operate under a shared understanding. Additionally, it plays a crucial role in onboarding by providing new team members with an explicit reference for team expectations and workflows.

During the validation process, SMEs emphasized that working agreements should exist not only at the team level but also at the organizational level. An organization-wide working agreement should define standards applicable across all teams at the LNSW. One example is the standardization of story points for effort estimation, ensuring consistent evaluation of task complexity. Moreover, the agreement should be collectively discussed and approved by all teams at the team and organizational levels to ensure broad acceptance and adherence.

- Definition of Ready and Definition of Done

The Definition of Ready (DoR) and Definition of Done (DoD) are critical components of Agile development that ensure clarity, efficiency, and quality in the software development lifecycle. DoR establishes the criteria a user story or backlog item must meet before entering a sprint, ensuring that tasks are well-defined, feasible, and actionable [25]. A well-structured DoR prevents ambiguities, reduces rework, and eliminates delays by identifying potential issues early in the process [26]. Without a clear DoR, teams may work on poorly defined tasks or lack necessary dependencies, leading to inefficiencies and sprint disruptions.

Similarly, the Definition of Done (DoD) ensures that a completed task meets quality, consistency, and transparency standards before being considered finished. It defines the requirements for design, development, testing, and documentation, ensuring that deliverables are production-ready and aligned with business expectations [26]. A well-enforced DoD minimizes technical debt, reduces defects, and enhances product reliability [25]. Without a structured DoD, teams risk delivering incomplete or low-quality work, resulting in rework, project delays, and misalignment with stakeholder requirements. During discussions, SMEs emphasized the need for the LNSW to incorporate DoR and DoD into an organization-wide working agreement to ensure alignment and awareness across all teams. Although the LNSW has existing criteria for determining when a task is ready to be worked on or considered complete, team awareness remains low, resulting in inconsistent implementation. Implementing standardized DoR and DoD at the organizational level is essential to improving workflow efficiency, reducing project risks, and ensuring effective Agile adoption across the LNSW.

2) *Increasing understanding of agile value*: Organizational culture, knowledge management, and process alignment/collaboration are significant challenges in Agile software development and its implementation with CMMI [10], [27]. Achieving the expected capability level in the LNSW demands more than adopting Agile frameworks. Formal Agile training is required to instill a shared understanding of Agile principles and best practices. Risk and Opportunities Management must also be embedded within Agile processes to ensure teams proactively identify obstacles, mitigate potential threats, and leverage growth opportunities

a) Formal agile training: To effectively implement and internalize Agile values, the LNSW must provide structured, formal training in Agile methodologies, particularly Scrum. This training equips individuals and teams with a deep understanding of the iterative, collaborative, and flexible principles that underpin Agile project management. The LNSW can enhance team alignment, improve workflow efficiency, and ensure a sustainable and scalable Agile transformation by developing core competencies. The key competencies required for Agile practitioners at the LNSW are derived from established frameworks such as the Scrum Guide [22], the Agile Practice Guide [28], and the CMMI 2-Based Scaling Agile Model [16].

The LNSW should implement a structured Agile training program in phases, starting with Agile Masters and Scaled Agile Masters as key facilitators of Agile adoption, adherence to processes, and team guidance. Once trained, they can mentor other roles, ensuring continuous knowledge-sharing and consistent Agile practices. SMEs support this approach, emphasizing that these roles must understand and apply Agile effectively while addressing team needs. A well-structured training program with ongoing mentoring will help the LNSW build a strong Agile culture, enhance collaboration, and improve overall implementation.

b) Risks and opportunities management: While traditional software development methodologies often incorporate well-defined risk management frameworks, Agile lacks a standardized, comprehensive risk management structure, requiring teams to integrate risk mitigation strategies within Agile processes [29]. It is important to have risk awareness at both the project and iteration levels, ensuring that risks are identified and addressed proactively through continuous tracking and resolution efforts. Agile teams dynamically manage risks through events such as Sprint Planning, Daily Meetings, Sprint Reviews, and Retrospectives. Still, these practices alone may not provide a structured and consistent risk-handling approach across projects [30]. In scaled Agile, risk management is further embedded into Scaled Cycle Planning using models such as ROAM (Resolve, Own, Accept, Mitigate) from SAFe [31]. It ensures that risks are identified early and addressed before they escalate into major blockers.

However, Agile practitioners, including SMEs, emphasize that risk management tools and models serve only as facilitators; the actual effectiveness of risk management lies in how well Agile teams integrate these practices into their workflows. To achieve this, each Agile event plays a specific role in risk identification, mitigation, and resolution, ensuring that teams systematically address risks throughout the development lifecycle. The following breakdown outlines how Agile events contribute to structured risk management:

- Cycle Planning: The Product Owner and Development Team review the backlog to identify uncertainties or dependencies that might pose risks. They discuss ambiguous or complex tasks and explore strategies such as breaking them into smaller, manageable units to minimize risk.
- Team Coordination Meeting: Continuous risk monitoring occurs as team members provide updates and highlight unexpected blockers or dependencies. The

Agile Master facilitates rapid resolution by reallocating tasks, escalating issues, or adjusting priorities.

- User Review: Stakeholders assess deliverables and encounter risks, ensuring that similar challenges are mitigated through feedback and process adjustments in future sprints.
- Team Retrospective: Teams evaluate how well risks were handled and identify improvements to prevent recurring challenges, fostering continuous learning and process refinement.
- Scaled Cycle Planning: Teams collaboratively identify and categorize risks using the ROAM model, ensuring that all risks are acknowledged and appropriately addressed. Fully resolved risks are under "Resolve", while those requiring ongoing attention are "Owned" by specific team members. This structured risk assessment enhances transparency and accountability within Agile teams.

Effective risk management goes beyond mitigating threats and opens the door to innovation and process optimization. A structured risk assessment framework enables teams to identify weaknesses, dependencies, and uncertainties, explore alternative solutions, and implement proactive improvements [29]. Rather than viewing all risks as obstacles, proper evaluation helps teams differentiate between pure threats and risks that present growth opportunities. Maintaining comprehensive documentation of identified risks and their corresponding mitigation strategies also builds a valuable knowledge base, helping teams recognize recurring patterns and leverage past insights to drive future improvements. To maximize these opportunities, risks should be documented, prioritized, and integrated into backlog tasks or UR/RFCs, ensuring strategic follow-up.

3) Utilization of Tools

a) Scrum/Task board: A Scrum/Task Board is a visual tool that displays the status of user stories or tasks within a sprint. It plays a crucial role in maintaining transparency, a core Scrum principle, enabling team members and stakeholders to track progress in real time. Scrum boards can be physical boards for co-located teams or digital task boards using Agile project management tools for distributed teams [26]. Rubin [25] further emphasizes the importance of task boards by describing them as dynamic representations of the evolving sprint backlog, where backlog items are broken down into tasks and visually moved through different workflow stages as work progresses.

For the LNSW, implementing a Scrum/Task Board can enhance transparency by aligning it with the team's workflow stages as column headers. It allows team members and stakeholders to track user stories and tasks in real time, ensuring clear visibility of sprint progress. The board also improves team communication by providing a shared view of ongoing work, keeping everyone aligned with sprint priorities. Additionally, it is a central focus during coordination meetings, helping teams monitor progress, address challenges, and plan the next steps effectively. This visual management tool allows LNSW to streamline workflows, improve collaboration, and strengthen Agile implementation. Fig. 3 shows the recommended stages for use on the board.

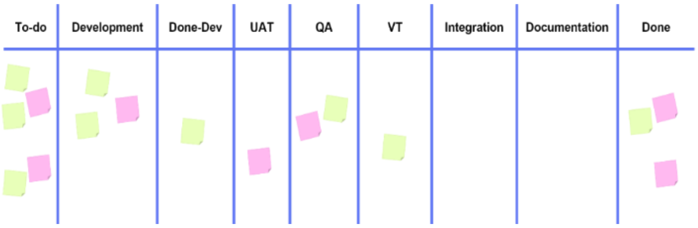


Fig. 3. Recommended LNSW task board illustrating a standardized workflow from To-Do to final documentation.

b) Estimation technique: Traditional estimation techniques, such as Function Points (FP), expert judgment, and Work Breakdown Structures (WBS), face subjectivity issues, reliance on historical data, and a lack of benchmarking [32]. These methods often rely on past project metrics or expert intuition, leading to inconsistent team decisions. In contrast, Agile estimation techniques, particularly Planning Poker, aim to mitigate these issues using a collaborative, consensus-based approach to estimate the effort required for a user story. Planning Poker encourages team discussions, reduces cognitive bias, and aligns workload expectations across team members. While traditional models struggle with evolving requirements, Planning Poker adapts to Agile's iterative nature, ensuring estimations reflect real-time complexities. [25], [26], [33].

Planning Poker is often combined with story points, a metric used in Agile to measure effort, complexity, and uncertainty in completing a user story. Unlike traditional time-based estimation, story points consider technical difficulty, risk, dependencies, and domain knowledge. This method allows teams to estimate work comparatively rather than using fixed durations, making it a more adaptable and reliable approach for Agile planning [33]. SMEs emphasized the need for a standardized definition of story points at the organizational level to ensure consistent task evaluation across teams. A shared understanding of story point values prevents significant discrepancies across teams, ensuring that estimations remain aligned regardless of which team performs the work. This standardization improves estimation accuracy, workload distribution, and cross-team collaboration, fostering better predictability in Agile execution.

For the LNSW, implementing Planning Poker with story points can help teams align on estimation, improve predictability, and enhance sprint planning accuracy. The structured process of Planning Poker involves:

1) Distributing Planning Poker cards or using digital tools for estimation.

2) Select a user story to be estimated, with the Product Owner providing a brief explanation.

3) Individually estimating effort by choosing a story point value.

4) Revealing all estimations simultaneously to avoid bias.

5) Discussing discrepancies if estimations vary significantly, refining assumptions, and re-voting if needed until a consensus is reached.

6) Assigning the final story point value, recording it in the product backlog, and using it for sprint planning.

a) Burndown chart: A Burndown Chart is a visual tool used to track the remaining work in a sprint and clearly represent progress based on estimation metrics such as time, story points, or task count. This tool enhances transparency for teams and stakeholders, offering insights into how efficiently work is completed within a sprint. Additionally, Burndown Charts help teams assess their velocity over time. It is a historical record of team capacity, indicating how many story points a team can typically complete in a sprint [25], [26].

To use a Burndown Chart effectively, teams should update it daily, recording the estimated effort remaining for all unfinished tasks. This real-time visualization helps teams determine whether they are on track to complete the sprint goals. As shown in Fig. 4, if the trend line stays above the ideal trajectory, the team falls behind, prompting further investigation into potential blockers, underestimated tasks, or other challenges. Conversely, if the trend line drops below the ideal trajectory, the team may consider taking on additional work or refining existing tasks [25].

However, LNSW expressed concerns that Burndown Charts could send a negative message if the reported progress does not appear ideal. This concern can be addressed by documenting action items in the team retrospective, which provides context and justification for deviations from the expected trajectory. By analyzing these action items, teams can identify the root causes of delays, justify challenges, and implement corrective measures in future sprints. Through proper documentation and continuous reflection, Burndown Charts can serve as an improvement tool rather than just a performance metric, ensuring constructive discussions and better sprint planning at the LNSW.

7) Proposed phased improvement strategy: To address the gaps identified in the appraisal and attain Capability Level 1, this study outlines the following phased roadmap:

- Phase 1: Standardization (Short-term / High Priority): Establish consistent process assets, including a standardized Team Working Agreement and formalized Definitions of Ready (DoR) and Done (DoD), while utilizing project management tools to institutionalize these standards across all development teams.
- Phase 2: Synchronization (Mid-term / Medium Priority): Institutionalize structured Agile events, specifically Scaled Coordination Meetings and Team Retrospectives, to manage dependencies across the 42 integrated systems and foster continuous improvement. Moreover, begin integrating the mapped Agile roles into daily operations to ensure clear accountability and leadership throughout the synchronization process.
- Phase 3: Optimization (Long-term / Medium Priority): Enhance Agile leadership through formal training for Agile Masters and Scaled Agile Masters, with a focus on cross-team synchronization and coaching. Concurrently, provide foundational Agile workshops for all developers to ensure a shared understanding of iterative delivery and collaborative best practices.

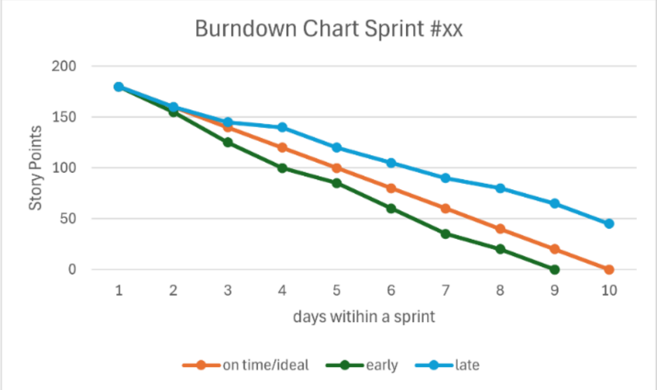


Fig. 4. Sample Burndown Chart illustrating ideal, early, and late progress trends for sprint tracking.

The execution of this strategy follows a realistic, tailored progression to the bureaucratic and technical complexity of a government integration hub. The short-term phase is expected to span 1 to 3 months, focusing on quick wins and administrative alignment that can be implemented without major structural changes. The mid-term phase typically requires 6 to 12 months, as it involves shifting organizational behavior and aligning the sprint cycles of multiple concurrent teams to manage dependencies effectively. Finally, the long-term phase is projected to take 1 to 2 years. This duration accounts for the deep cultural shift required for leadership coaching and the time needed to establish a self-sustaining environment of continuous optimization and shared Agile values.

V. Conclusion

The study evaluates the Scaled Agile implementation in SINSW development, identifies key challenges and inefficiencies, and proposes recommendations to enhance Agile adoption. The findings reveal that team coordination, structured Agile events, and role clarity remain significant areas for improvement. The appraisal results indicate that critical Agile elements such as Scaled Coordination Meetings and Team Retrospectives have not yet been implemented. At the same time, Team Coordination Meetings, Agile Master, and Scaled Agile Master roles are only partially achieved. Although the Success Criteria demonstrated some progress, the overall Agile capability remains at Capability Level 0 (CL0), signifying that Agile processes are still in their early stages and require substantial improvements.

To address the challenges in achieving Capability Level 1, this study proposes a three-phased improvement strategy. This roadmap begins with Phase 1: Standardization, which focuses on elaborating Agile workflows and utilizing project management tools to unify team coordination. It progresses to Phase 2: Synchronization, which institutionalizes structured Agile ceremonies to resolve inter-team dependencies. Finally, Phase 3: Optimization enhances leadership through specialized training for Agile Masters and Scaled Agile Masters. Together, these recommendations bridge identified gaps, fostering a sustainable Agile transformation and improved operational efficiency across LNSW.

Strengthening Agile implementation in SINSW development will lead to greater efficiency, improved team collaboration, and better alignment with organizational goals. Implementing these recommendations can help ensure that Agile practices are adopted, sustained, and continuously enhanced, enabling LNSW to achieve higher levels of agility and responsiveness in the evolving landscape of government digital services.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to all individuals and institutions who contributed to the successful completion of this research. We especially thank Universitas Indonesia for their generous support under contract number NKB-1820/UN2.RST/HKP.05.00/2020. We also wish to acknowledge the valuable collaboration of the Indonesia National Single Window Agency (LNSW) for providing essential data and insights for this study.

USE OF AI TECHNOLOGY

This manuscript used Grammarly and the free version of ChatGPT solely to enhance English fluency. No other AI tools were involved in the research or writing process.

REFERENCES

[1] World Customs Organization, "BUILDING A SINGLE WINDOW ENVIRONMENT," 2017. Accessed: Sep. 19, 2023. [Online]. Available: https://www.wcoomd.org/-/media/wco/public/global/pdf/topics/facilitation/instruments-and-tools/tools/single-window/compendium/swcompendiumvol1all-parts.pdf.

[2] Ministry of Finance of the Republic of Indonesia, "214/PMK.012/2022 on the Management of the Indonesia National Single Window and the Implementation of the Indonesia National Single Window System," Jakarta, Dec. 2022. Accessed: Sep. 19, 2023. [Online]. Available: https://peraturan.bpk.go.id/Details/238739/pmk-no-214pmk0122022.

[3] National Single Window Agency, "Monthly Report December 2023 Procurement of Consulting Services for the Development of INSW System (SINSW) Generation-II for the 2023 Fiscal Year National Single Window Agency (LNSW)," 2023.

[4] D. Rose, Enterprise Agility For Dummies. Wiley, 2018.

[5] H. Vedal, V. Stray, M. Berntzen, and N. B. Moe, "Managing Dependencies in Large-Scale Agile," 2021, pp. 52–61. doi: 10.1007/978-3-030-88583-0_6.

[6] digital.ai, "17th State of Agile Report," 2024.

[7] P. L. Ayunda and E. K. Budiardjo, "Evaluation of Scrum Practice Maturity in Software Development of Mobile Communication Application," in 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE), IEEE, Sep. 2020, pp. 317–322. doi: 10.1109/IC2IE50715.2020.9274625.

[8] A. Ferdinansyah and B. Purwandari, "Challenges in Combining Agile Development and CMMI: A Systematic Literature Review," in 2021 10th International Conference on Software and Computer Applications, New York, NY, USA: ACM, Feb. 2021, pp. 63–69. doi: 10.1145/3457784.3457803.

[9] T. Raharjo and B. Purwandari, "Agile Project Management Challenges and Mapping Solutions," in Proceedings of the 3rd International Conference on Software Engineering and Information Management, New York, NY, USA: ACM, Jan. 2020, pp. 123–129. doi: 10.1145/3378936.3378949.

[10] A. Ferdinansyah and B. Purwandari, "Challenges in Combining Agile Development and CMMI: A Systematic Literature Review," in 2021 10th International Conference on Software and Computer Applications, New York, NY, USA: ACM, Feb. 2021, pp. 63–69. doi: 10.1145/3457784.3457803.

[11] M. Palomino, A. Dávila, K. Melendez, and M. Pessoa, "Agile Practices Adoption in CMMI Organizations: A Systematic Literature Review," 2017, pp. 57–67. doi: 10.1007/978-3-319-48523-2_6.

[12] CMMI Institute, CMMI Development V2.0. 2018.

[13] M. Degerli, "Practical Suggestions to Successfully Adopt the CMMI V2.0 Development for Better Process, Performance, and Products," in 2020 5th International Conference on Computer Science and Engineering (UBMK), IEEE, Sep. 2020, pp. 126–129. doi: 10.1109/UBMK50275.2020.9219438.

[14] The CMMI Product Team, CMMI for Development, Version 1.3. Software Engineering Institute, 2010. doi: 10.1184/R1/6572342.v1.

[15] CMMI Institute, CMMI Model V2.2. 2021.

[16] A. Ferdinansyah, "SCALING AGILE MODEL BASED ON CMMI 2 DAN THE ESSENCE OF SOFTWARE ENGINEERING," Universitas Indonesia, 2022.

[17] A. Marotti de Mello and T. Wood Jr, "What is applied research anyway?," Revista de Gestão, vol. 26, no. 4, pp. 338–339, Oct. 2019, doi: 10.1108/REGE-10-2019-128.

[18] T. E. Hedrick, L. Bickman, and D. J. Rog, Applied Research Design: A Practical Guide. SAGE Publications, Inc, 1993.

[19] J. T. Marchewka, Information Technology Project Management: Providing Measurable Organizational Value, 5th Edition. Wiley, 2015.

[20] C. Patel and M. Ramachandran, "Agile maturity model (AMM): a software process improvement framework for agile software development practices," International Journal of Software Engineering, IJSE, vol. 2, pp. 3–28, 2009.

[21] A. Joskowski, A. Przybyłek, and B. Marcinkowski, "Scaling scrum with a customized nexus framework: A report from a joint industry-academia research project," Softw. Pract. Exp., vol. 53, no. 7, pp. 1525–1542, Jul. 2023, doi: 10.1002/spe.3201.

[22] K. Schwaber and J. Sutherland, "The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game," 2020.

[23] K. Schwaber, "The NexusTM Guide: The Definitive Guide to Scaling Scrum with Nexus," Jan. 2021.

[24] J. Sutherland, "The Scrum At Scale® Guide: The Definitive Guide to the Scrum@Scale Framework," Feb. 2022.

[25] K. S. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process. Upper Saddle River, NJ, 2012.

[26] J. Dalton, Great Big Agile. Berkeley, CA: Apress, 2019. doi: 10.1007/978-1-4842-4206-3.

[27] H. Fissalma, A. Ferdinansyah, and B. Purwandari, "Investigating challenges in Agile software development: a cross-country comparative analysis," International Journal of Electrical and Computer Engineering (IJECE), vol. 15, no. 1, p. 855, Feb. 2025, doi: 10.11591/ijece.v15i1.pp855-869.

[28] Project Management Institute and Agile Alliance, Agile Practice Guide. 2017.

[29] M. H. Zahedi, A. Rabiei Kashanaki, and E. Farahani, "Risk management framework in Agile software development methodology," International Journal of Electrical and Computer Engineering (IJECE), vol. 13, no. 4, p. 4379, Aug. 2023, doi: 10.11591/ijece.v13i4.pp4379-4387.

[30] A. Moran, Agile Risk Management. Cham: Springer International Publishing, 2014. doi: 10.1007/978-3-319-05008-9.

[31] R. Wanner, "The ROAM Risk Model in Agile Projects and Its Deficiencies." Accessed: Feb. 04, 2025. [Online]. Available: https://rolandwanner.com/roam-risk-model-agile-projects-deficiencies/

[32] J. Angara, S. Prasad, and G. Sridevi, "Towards Benchmarking User Stories Estimation with COSMIC Function Points-A Case Example of Participant Observatio," International Journal of Electrical and Computer Engineering (IJECE), vol. 8, no. 5, p. 3076, Oct. 2018, doi: 10.11591/ijece.v8i5.pp3076-3083.

[33] M. Fernandez-Diego, E. R. Mendez, F. Gonzalez-Ladron-De-Guevara, S. Abrahao, and E. Insfran, "An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review," IEEE Access, vol. 8, pp. 166768–166800, 2020, doi: 10.1109/ACCESS.2020.3021664.