The Indonesian Journal of Computer Science
www.ijcs.net
Volume 14, Issue 1, February 2025
https://doi.org/10.33022/ijcs.v14i1.4619

Evaluation of Scrum Implementation Maturity Level based on Scrum Body of Knowledge

Andi Akram Yusuf[1], Eko K. Budiardjo[2], Kodrat Mahatma[3]
andi.akram@cs.ui.ac.id[1], eko@cs.ui.ac.id[2], kodrat.mahatma21@cs.ui.ac.id[3]
[1,2,3]Faculty of Computer Science, Universitas Indonesia

Article Information

Received : 3 Jan 2025
Revised : 27 Jan 2025
Accepted : 7 Feb 2025

Keywords

Maturity Level, Scrum Implementation, Scrum Body of Knowledge, SBOK

Abstract

The banking industry's core business involves collecting public funds and distribute them into loans. Intense competition within the banking sector demands speed in business development processes, including the development of mobile banking applications. At PT Bank XYZ, the Mobile Banking Team develops mobile banking applications by implementing the Agile methodology and adopting the Scrum framework. The choice of the Scrum framework aims to facilitate the development of small-value features that can be quickly released, thus creating significant value for users. The faster this value reaches the end users, the quicker user feedback can be gathered for further improvements. However, the implementation of Scrum has not been fully effective, as evidenced by the presence of carried-over features and feature enhancements. This research employs a mixed-method approach with an exploratory sequential design, starting with qualitative research followed by quantitative research. Qualitative data were collected through observations and interviews, while quantitative data were gathered through questionnaires. The questionnaires were developed using the Scrum Body of Knowledge (SBOK). The data were then processed using the KPA Rating method. The results indicate that the implementation is still at Level 1 (Explored). Recommendations are formulated based on SBOK and focus on achieving Level 2 (Defined) and Level 3 (Improved). Level 4 was not included in the recommendations, as mobile banking development at PT Bank XYZ has not yet reached the enterprise stage.

A. Introduction

The primary business of banks is to collect public funds and distribute them as loans. Intense competition makes speed a crucial factor in the development process. The development of the mobile banking application at Bank XYZ is carried out by the Mobile Banking Team (MB Team) using the Agile methodology, enabling the application to be quickly launched to customers. Typically, development processes at Bank XYZ followed the waterfall methodology, which is known for its lengthy timeline as all requirements must be completed before application development begins. However, with the evolution of development methods, when Bank XYZ began developing its own mobile banking application, the Agile methodology was adopted through the issuance of the Tribe Decree. This decree binds multiple teams to provide dedicated resources for the project, including the MB Team, which plays the role of developers. At the beginning of the development, Agile was still unfamiliar, and the process relied solely on following trends and holding simple discussions with acquaintances who were also implementing this methodology.

Agile itself is a project management framework for software development. The chosen framework is Scrum, one of the most popular frameworks within the Agile methodology [1]. Fundamentally, Agile encompasses several frameworks, such as Rapid Application Programming, Extreme Programming, and Scrum. Its primary advantage is the speed at which applications can be released to users. Applications do not need to have all features fully developed before being released; instead, they should deliver value that users need. These values are what developers aim to achieve during application development, and once these values are realized, the application can be released to users. However, these values are typically small and released incrementally, ultimately forming significant value for users. The faster these values reach users, the quicker feedback can be obtained from users for further development.

The decision to use Scrum was initially based on trends, leading to an iterative process with a trial-and-error approach. Over time, as knowledge grew and new members joined, bringing fresh insights, the implementation of Scrum improved. However, as the team expanded, the challenges of implementing Scrum also increased. There were aspects of Scrum implementation deemed suboptimal by the Scrum Master but difficult to change. As a result, the development process became inconsistent, application quality suffered, evidenced by frequent feature enhancements, and development speed became unstable, marked by carryovers.

Fishbone analysis was used to break down the issues, revealing that the problems were in the process domain. These included the absence of a product backlog before sprint planning, a lack of a flexible definition of done, and overly general product slicing. These issues hindered the smooth implementation of Scrum processes. Thus, the research questions are: What is the maturity level of Scrum framework implementation at Bank XYZ, and what recommendations can be made to improve its maturity level?

B. Literature Review

1. Agile Software Development

Agile Software Development (ASD) is a methodology for developing software. Its foundation lies in creating software within a short time frame and with a high level of adaptability. This method emphasizes teamwork and collaboration. ASD is built upon four main principles, often referred to as the Agile Manifesto [2]: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.

2. Scrum

Scrum is a lightweight framework that helps individuals, teams, and organizations generate value through adaptive solutions to complex problems [3]. This framework is commonly applied in software development, but Scrum values can also be used outside of software development. Scrum is based on two key principles: empiricism and lean thinking. Empiricism is the mindset that knowledge comes from experience and making decisions based on what is observed. Lean thinking focuses on reducing waste and concentrating on what is essential or improving effectiveness. In general, there are three main pillars of Scrum: transparency, inspection, and adaptation [3].

3. Agile Maturity Model (AMM)

The Agile Maturity Model (AMM) is a model used to measure the maturity level of Scrum implementation. AMM serves as an alternative to existing process assessment models, such as CMMI [4]. It is utilized to gain deeper insights into the implementation of Agile methods and assess how mature the implementation is compared to commonly practiced standards. This maturity assessment can be conducted repeatedly at specific intervals to track implementation progress. The maturity levels are Initial, Explored, Defined, Improved, and Sustained.

4. Scrum Maturity Model (SMM)

Another model for measuring maturity levels is the Scrum Maturity Model (SMM). SMM is specifically designed for the Agile methodology, particularly Scrum. Its primary goal is to assist and guide software development organizations [5]. This model consists of five levels, like those in AMM, which are: Initial, Managed, Defined, Quantitatively Managed, and Optimizing [5].

5. Scrum Body of Knowledge (SBOK)

The Scrum Body of Knowledge (SBOK) was developed as a guide for organizations and project management teams seeking to implement Scrum, as well as for those already using Scrum but needing improvements to their existing processes [6]. The SBOK content is generally divided into three sections: Principles, Aspects, and Processes. Processes refer to the activities performed throughout a Scrum project. These are divided into five main categories comprising a total of 19 processes. Each process produces outputs essential for subsequent processes. These processes are detailed in the Table 1Table 1.

6. Capability Maturity Model Integration (CMMI)

The Capability Maturity Model Integration (CMMI) is a framework designed to enhance the performance of continuously evolving projects [7]. CMMI provides 32 components, known as Practice Areas (PA), that serve as a reference for assessments and as a foundation for recommendations to improve an organization's performance in executing a project [7]. These 32 components are categorized into four groups: Doing, Managing, Enabling, and Improving. Assessments in CMMI are conducted using two levels: Continuous Representation/Capability Level and Staged Representation/Maturity Level.

Table 1. Table Scrum processes

Phase	Fundamental Scrum Processes
Initiate	Create Project Vision Identify Scrum Master and Business Stakeholder(s) Form Scrum Team Develop Epic(s) Create Prioritized Product Backlog Conduct Release Planning
Plan and Estimate	Create User Stories Estimate User Stories Commit User Stories Identify Tasks Estimate Tasks Update Sprint Backlog
Implement	Create Deliverables Conduct Daily Standup Refine Prioritized Product Backlog
Review and Retrospect	Demonstrate and Validate Sprint Retrospect Sprint
Release	Ship Deliverables Retrospect Release

The main difference between Capability Level and Maturity Level lies in Levels 4 and 5, which are specific to Maturity Level. Capability Level assesses an organization's ability regarding individual PAs, while Maturity Level evaluates the organization's overall implementation of the CMMI model [7].

7. Related Works

Prior to conducting this research, several studies had already evaluated Scrum. Studies [8] and [9] assessed the maturity level of Scrum implementation using the Scrum Maturity Model (SMM), influenced by the Agile Maturity Model (AMM) and analyzed with KPA ratings. Studies [10] and [11] also used SMM, though [11] was influenced by the Scrum Body of Knowledge (SBOK). Another study, [12], used the Capability Maturity Model Integration (CMMI) as its framework. Studies [8] and [11] used the Scrum Guide and SBOK to formulate recommendations, while [9] relied solely on SMM, and [10] adopted Scrum at Scale.

Based on previous research, most studies utilized SMM, with some using CMMI, but none employed SBOK as the framework. This study uses SBOK as its framework because SBOK provides detailed explanations for each Scrum process and is specifically tailored to Scrum, whereas SMM is less detailed, and CMMI does not focus on Scrum. Additionally, SBOK offers guidance for scaling to large projects and enterprises, making it suitable for the research context, as this study focuses on the MB team applying Agile on a large project scale.

C. Methodology

In summary, this research began with problem identification through interviews and document observations. The root causes of the issues were then analyzed using fishbone analysis. Before addressing the problems, the researcher reviewed previous studies. These studies led to the selection of SBOK as the chosen methodology. Next, a questionnaire was developed using SBOK as the model, as no prior research had utilized SBOK in this way. Once finalized, the questionnaire was distributed to respondents, consisting of six Squad Leaders (SL) and six Product Owners (PO). The collected data was processed using KPA Rating, which determined the maturity level and the phases requiring improvement. These phases were then analyzed further, and recommendations were formulated based on SBOK's implementation guidelines. This workflow is illustrated in Figure 1.

1. Creating Questionnaire with SBOK

The model development began with categorizing phases, which were then outlined into maturity levels adopted from the AMM method: Explored (1), Defined (2), Improved (3), and Sustained (4). The Initiation phase corresponds to level 1. Plan and Estimate, Implement, Review and Retrospect, and Release correspond to level 2. Scaling for Large Projects is added for level 3, and Scaling for Enterprise for level 4. These phases include mandatory processes required for implementing SCRUM. Additionally, each process has outputs identified as essential for executing SCRUM. Regularly producing these outputs indicates that the SCRUM implementation is functioning effectively. These mandatory outputs were then formulated into questions.

After the questionnaire was created, the researcher first distributed it to two members of the development team to test whether it was easily understood. If any big issues were found, the questions were adjusted and retested, but if only minor issues were found then the questions were adjusted accordingly before being distributed to the Squad Leaders (SL) and Product Owners (PO).

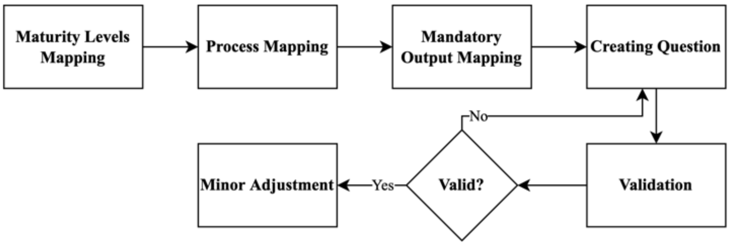

Figure 1. Creating Questionnaire Flow

2. Data Analysis

The data collected from the questionnaire was then processed using the following formula:

$$\frac{\sum(Y_n) + \frac{1}{2}(P_n)}{\sum(T_n) - \sum(NA_n)} \times 100\% \tag{1}$$

Where Yn represents the total number of "Yes" responses, Pn represents the total number of "Partially" responses, Tn is the total number of responses, and NAn is the total number of "Not Applicable" responses. The results of the KPA Rating calculation will be categorized into four categories, namely:

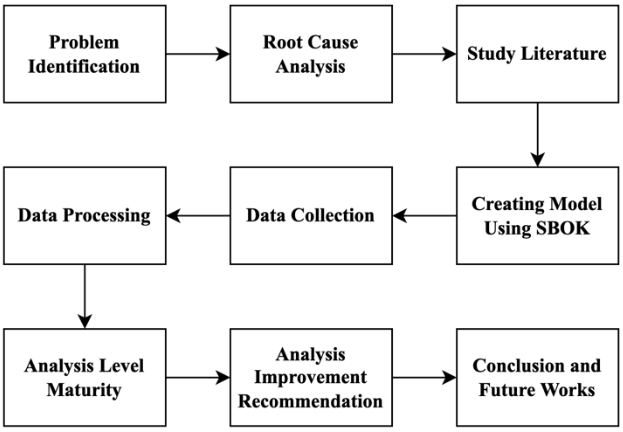

Figure 2. Research Flow

- Fully Achieved: This rating indicates that all processes within the phases outlined in the SBOK Guide have been implemented effectively and do not require recommendations for improvement in that phase. The rating value for this category is above 85%.
- Largely Achieved: This rating indicates that the processes within the phases outlined in the SBOK Guide have been implemented well but are not yet optimal, leaving room for improvement in the implementation of those phases. The rating value for this category is above 50%.
- Partially Achieved: This rating indicates that the processes within the phases outlined in the SBOK Guide have not been implemented effectively and require improvements in the implementation of those phases. The rating value for this category is above 15%.
- Not Achieved: This rating indicates that the processes within the phases outlined in the SBOK Guide have not been implemented at all. The rating value for this category is 15% or below.

D. Result and analysis

Based on the questionnaire results in Table II, it can be seen that the levels with values above 85% are only present at Level 1, indicating that the implementation of Scrum in the MB team is currently at Level 1.

Table 2. Result Recapitulation

Level	Phase	Process	KPA	KPA
1	INITIATE	Create Project Vision	90.63%	85.42%
		Identify Scrum Master and Business Stakeholder(s)	87.50%	
		Form Scrum Team	87.50%	
		Develop Epic(s)	71.88%	
		Create Prioritized Product Backlog	81.25%	
		Conduct Release Planning	93.75%	
2		Create User Stories	75.00%	68.63%

Level	Phase	Process	KPA	KPA
	PLAN AND ESTIMATE	Estimate User Stories	56.25%	
		Commit User Stories	79.17%	
		Identify Tasks	73.21%	
		Estimate Tasks	53.13%	
		Update Sprint Backlog	75.00%	
	IMPLEMENT	Create Deliverables	60.42%	61.81%
		Conduct Daily Standup	56.25%	
		Refine Prioritized Product Backlog	68.75%	
	REVIEW AND RETROSPECT	Demonstrate and Validate Sprint	68.75%	71.88%
		Retrospect Sprint	75.00%	
	RELEASE	Ship Deliverables	83.33%	67.78%
		Retrospect Release	52.23%	
3	SCALING SCRUM FOR LARGE PROJECTS		79.21%	79.21%
4	SCALING SCRUM FOR THE ENTERPRISE	Create/Update Program or Portfolio Teams	66.67%	70.46%
		Create/Update Program or Portfolio Components	70.41%	
		Review and Update Scrum Guidance Body	50.00%	
		Create/Refine Prioritized Program or Portfolio Backlog	75.00%	
		Create/Update Program or Portfolio Releases	85.71%	
		Retrospect Program or Portfolio Releases	75.00%	

Recommendations are focused on Level 2 since the maturity level is still at Level 1. The recommendations are developed based on processes that have not yet been Fully Achieved. The process is detailed in **Error! Reference source not found.**.

Table 3. Recommendations

Process	Recommendation
Create User Stories	Write user stories in a clear format understandable by both the team and stakeholders and ensure user stories meet the Definition of Ready (DoR).
	Define and agree upon Acceptance Criteria for each user story.
	Prioritize user stories based on business value.
Estimate User Stories	Involve the entire team in estimating user stories that meet the DoR.
	Use previous similar user stories as reference for estimation.
	Apply established estimation methods like Wideband Delphi, Planning Poker, Fist of Five, or Affinity Estimation.
Commit User Stories	Review user stories to confirm all team members understand them.
	Validate that team capacity and workload are balanced, ensuring tasks are neither excessive nor insufficient.
	Commit to user stories once finalized, avoiding mid-sprint changes.
Identify Tasks	Divide user stories into manageable tasks aligned with completing the user story. Further split user stories if necessary for better manageability.
	Involve the team in task creation for shared understanding.

Process	Recommendation
Estimate Tasks	Involve the team for estimate tasks.
	Use previous similar task as reference for estimation.
	Estimate tasks using agreed methods, referring to similar tasks for consistency.
Update Sprint Backlog	Regularly update the Sprint Backlog with the latest details.
	Use Sprint Burndown/Burnup Charts to monitor progress and evaluate the sprint.
Create Deliverables	If any impediments arise, promptly communicate and resolve them with the assistance of the Scrum Master, Product Owner (PO), or Squad Leader.
	Always update the status of tasks being worked on.
	Monitor Sprint progress using the Sprint Burndown/Burnup Chart, and if necessary, prioritize tasks with the highest business value.
	Adhere to the Definition of Done (DoD) or Done Criteria to ensure consistent quality in deliverables.
	Ensure that there are no changes to the requirements for user stories currently in progress.
Conduct Daily Standup	Limit meeting durations to a maximum of 15 minutes.
	Focus the discussion on three key points: what has been done, what will be done, and any impediments encountered.
	Rotate the facilitator role daily to enhance a sense of ownership and team camaraderie.
Refine Prioritized Product Backlog	Conduct refinement/meetings before sprint planning to ensure the Scrum Team thoroughly understands the user stories to be worked on and can provide feedback.
	The Product Owner ensures that user stories meet the Definition of Ready (DoR).
Demonstrate and Validate Sprint	The Scrum Team ensures deliverables meet the Definition of Done (DoD) before demonstrating them.
	Showcase how the deliverables provide value or solutions to business needs.
	Discuss and record feedback received so it can be promptly addressed.
Retrospect Sprint	Focus on identifying what worked well, what needs improvement, and the plans for improvement in the next sprint.
	Use varied retrospective techniques to prevent monotony.
	Create a comfortable environment to encourage full participation from everyone.
Ship Deliverables	Conduct final validation to ensure the product is ready for release.
	Ensure all items meet the Definition of Done (DoD) and acceptance criteria.
	Prepare a deployment checklist that includes all technical steps and final validations.
	Provide clear documentation and user guides for the end users.
Retrospect Release	Conduct a comprehensive release process evaluation, covering both technical aspects (deployment stability, deliverable performance) and non-technical aspects (communication).
	Involve all relevant teams in this evaluation.
	Foster a constructive atmosphere, focusing on areas that can be improved.

Suggestions for future research include:

1) The subject of this study is the development team for mobile banking applications at PT Bank XYZ. Future research is recommended to focus on different subjects to enhance this method further. The more studies conducted using SBOK as the primary framework, the more robust and well-tested the method will become.

2) The recommendations and methods used in this study solely rely on SBOK without any adjustments or incorporation of other methods. Future research should consider adding other methods as supplementary references to refine and strengthen this approach.

E. References

[1] M. Hron and N. Obwegeser, "Why and how is Scrum being adapted in practice: A systematic review," *Journal of Systems and Software*, vol. 183, Oct. 2021, doi: 10.1016/j.jss.2021.111110.

[2] K. Beck *et al.*, "Manifesto for Agile Software Development," *Manifesto for Agile Software Development*. 2001. [Online]. Available: http://www.agilemanifesto.org/

[3] K. Schwaber and J. Sutherland, "The Scrum Guide," 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:114128971

[4] M. Leppänen, "A Comparative Analysis of Agile Maturity Models," *Information Systems Development: Reflections, Challenges and New Directions*, pp. 329–343, Apr. 2013, doi: 10.1007/978-1-4614-4951-5_27.

[5] A. Yin, S. Figueiredo, and M. Mira da Silva, "Scrum Maturity Model," Jan. 2011.

[6] SCRUMstudy, *A Guide to the Scrum Body of Knowledge (SBOK™ Guide)*, 4th ed. Phoenix, AZ: SCRUMstudy, 2022. [Online]. Available: https://www.scrumstudy.com/sbokguide/download-free-buy-sbok

[7] CMMI Product Team, "CMMI for Development, Version 1.3," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Technical Report CMU/SEI-2010-TR-033, 2010. [Online]. Available: https://insights.sei.cmu.edu/library/cmmi-for-development-version-13/

[8] H. Zelfia, T. Simanungkalit, and T. Raharjo, "Comparison of Scrum Maturity Between Internal and External Software Development: A Case Study at One of the State-Owned Banks in Indonesia," Jul. 2022, pp. 312–317. doi: 10.1109/ICISIT54091.2022.9872843.

[9] K. Abimaulana, E. Budiardjo, K. Mahatma, and A. Hidayati, "Evaluation of Scrum-Based Software Development Process Maturity using the SMM and AMM: A Case of Education Technology Startup," Oct. 2021, pp. 1–5. doi: 10.1109/ICACSIS53237.2021.9631308.

[10] Ahmad Jalaluddin, Eko K. Budiardjo, and Kodrat Mahatma, "Recommendation for Scrum-Based Software Development Process with Scrum at Scale: A Case Study of Software House XYZ," *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 8, no. 3, pp. 401–412, Jun. 2024, doi: 10.29207/resti.v8i3.5646.

[11] N. Arifin, B. Purwandari, and F. Setiadi, "Evaluation and Recommendation for Scrum Implementation Improvement with Hybrid Scrum Maturity Model: A Case Study of A New Telco Product," Nov. 2020, pp. 178–183. doi: 10.1109/ICIMCIS51567.2020.9354311.

[12] N. Pramitasari, "Peningkatan Kualitas Penerapan Scrum dalam Pengembangan Aplikasi Berdasarkan CMMI V2.0: Studi Kasus Mobile Apps ABC di Bank XYZ,"

Tugas Akhir Sarjana, Universitas Indonesia, Depok, Indonesia, 2023. [Online]. Available: https://lib.ui.ac.id/detail?id=9999920530587&lokasi=lokal