ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 12 (2022) No. 6, pp. 561-582
DOI:10.33168/JSMS.2022.0633

Measuring Maturity Level of Scrum Practices in Software Development Using Scrum Maturity Model

Irma Panjaitan, Nilo Legowo

Information Systems Management Department, Binus Graduate Program, Master of Information Systems Management, Bina Nusantara University, Jakarta Indonesia 11480

irma.panjaitan@binus.ac.id, nlegowo@binus.edu

Abstract. The implementation of scrum framework in software development is expected to be responsive to changes and deliver quality products to market quickly. However, there are many obstacles in implementing scrum and scrum adoption process is often not carried out optimally. Such obstacles have resulted in features that are not in line with stakeholders' expectations and software quality that has not sufficiently met the expectations of users. Based on the problems, this study evaluated the implementation of scrum by measuring the maturity level of software development using scrum maturity model (SMM). SMM is more focused on evaluating and providing direction for software development process through the utilization of scrum. Practices in SMM were modified following scrum guide 2020 and measured using key process area (KPA) rating used in the agile maturity model (AMM). The data of this study were retrieved using focus group discussions with scrum team members who answered all modified practices in SMM. The assessment showed that the scrum practice is at level 1 (initial). The gaps between SMM practices and the existing practices implemented in product development were observed. Based on the observations, this study provides recommendations for scrum events, scrum artefacts and scrum roles that can be used to improve scrum implementation process in the future.

Keywords: software, scrum guide 2020, scrum maturity model, agile maturity model, KPA rating

1. Introduction

Software is a set of instructions or computer programs, data structures, and documents that can perform a certain function when certain instructions are executed (Pressman, 2011). Software product development consists of several processes of creating changes to products (Gambo et al., 2018). It consists of several activities to deliver a product to the market and consistently meet customer expectations and business requirements. A good product is a product that successfully provides solutions to customers. The focus of the outcomes would be that stakeholders see the value of the product development process and that the product can be made available in the market faster.

Software development has recently embraced a powerful paradigm called agile development. Agile software development has a significant influence on product development. It consists of smaller cycles, which is usually referred to as iterations, as well as cross-functional and self-organizing teams of software developers throughout the software development life cycle (Schwaber & Sutherland, 2020). There are various frameworks that follow the values offered by agile development. One of the most widely used frameworks today is scrum. Scrum is one of the frameworks that can solve complex problems (Betta & Skomra, 2019). One main focus of scrum is breaking larger projects into smaller, more manageable pieces of work.

To cite an example, a media monitoring and analysis company with more than 100 years of experience in monitoring printed media, online media, social media, radio, and television offers cutting-edge technology to provide monitoring services to thousands of clients. It must react quickly to the evolving needs and requirements in a rapidly changing market and environment. To provide data to support the media monitoring and analysis product, the IT development division works on several types of IT projects and provide some features that are suitable for customer needs.

The company has applied the scrum framework across all its project development. However, the process of implementing scrum in software development experienced numerous obstacles. Based on the evaluation results within the last two years, several projects were completely delayed, the sprint backlog was not achieved, the product backlog was not appropriate, and the sprint stopped. Based on this study's observations, this problem occurred because the adoption of scrum, such as scrum events, has not been carried out optimally, the scrum guide of roles and responsibilities was not properly followed, and other scrum practices were not well implemented. Therefore, it is necessary to evaluate software development process and measure maturity level of current scrum practices.

Mature software organizations have the ability throughout the organization to manage software development and maintenance process (Paulk et al., 2000).

Maturity levels allow organizations to evaluate and classify their capabilities and encourage organizations to improve capabilities through gradual methods. One method used to measure maturity levels and provide guidance to ensure a stable, capable, and mature process is the capability maturity model integration (CMMI). It is a framework used to assess the maturity level of an organization that can be used as a benchmark for process improvements (Kulpa and Johnson, 2008). According to (Yin (2011), CMMI cannot adequately assist organizations in implementing agile processes. The agile maturity model (AMM) (Chetankumar & Ramachandran, 2009) is designed to be inspired by CMMI, in which each goal at each level refers to the existing practices in agile development. Among the various agile maturity models is the scrum maturity model (SMM) that is more focussed on evaluating and providing direction for software development process by using scrum (Yin, 2011).

In evaluating the maturity level and providing suggestions for improvement of the software development process, this study modified the practices of scrum maturity model proposed by Yin (2011). The modification was initiated since the scrum maturity model compiled by Yin still refers to the scrum guide 2010, and therefore the overall practices need to be modified and adjusted to the latest scrum guide, namely scrum guide 2020.

This study has developed an evaluation to measure the scrum practice in PT XYZ, a media monitoring and analysis company using a scrum maturity model. The results of the evaluation describe the gaps in the existing rules and procedures that should be carried out. In addition, this study also provides recommendations for improvements in the future by referring to the 2020 scrum guide. The recommendations can be used to evaluate present problems and improve the quality of software delivered in subsequent development projects.

2. Literature Review

2.1. Agile

Agile is a software development methodology that requires collaborations between teams. It focuses on team interactions rather than documentations or tools during software development. Agile is faster in responding and tolerating change requirements.

Values that are promoted in software development using agile, or agile values (Beck, 2001), include (1) collaboration of development team or each party involved in the project, (2) focus on well-functioning software, (3) more effective collaboration between developers with customers in achieving project success compared to contract negotiation, (4) higher emphasis on responding to changes than following a plan on environmental uncertainty.

The agile manifesto (Fowler & Highsmith, 2001) is a guide that can be used as a basis in software development by implementing agile as follows: (1) client

satisfaction is a top priority, (2) changes in requirements are tolerated even when they occur at the end of software development, (3) results are provided in a span of between two weeks and two months, (4) business teams and software developers collaborate on a daily basis during software development, (5) software is developed by highly-motivated team members who can be trusted in completing the development, (6) the most effective and efficient method of conveying information is through face-to-face conversations, (7) the primary measure of project progress is well-functioning software, (8) support from sponsors, developers, and users can maintain consistency in software development speed, (9) continuous attention to technical and design aspects can increase the speed of software development, (10) simplicity is essential, (11) good architecture, requirements, and design originate from team independence, (12) periodical evaluations are carried out to measure the improvement and effectiveness of processes.

2.2. Scrum

Scrum is a framework that can solve complex problems that frequently change without compromising the quality and value of the product (Schwaber & Sutherland, 2020). Products are developed iteratively and incrementally to minimize risks and increase predictability. Scrum values such as commitment, courage, focus, openness, and respect support the development environment. Based on the Scrum Body of Knowledge (VMEdu,2016), the scrum stages are divided into five stages, namely initiation, planning and estimation; implementation; review; retrospective; and release. According to Rubin (2019) the scrum essentials consist of roles, activities, and artifacts.

A software development team in scrum typically consists of 5 to 9 people, including product owner, scrum master, and developers. Each member has their responsibilities and self-organizing is expected. Product owner represents users and is responsible for maximizing the product values. Product owner has to manage the product backlog. They optimize the work value of the development team performance, ensuring that the product backlog is visible, transparent, and clear to the team developer as well as ensuring that the development team understands all items in the product backlog. A developer is a team member who has the right skills as part of the team to do the work. The developer team is a self-organizing cross-functional unit. It decides how to accomplish the results stipulated by the product owner, has all the skills needed in the team, and is empowered to do what is necessary to manage the work and meet this principle (Noll et al., 2017). The scrum master helps to facilitate scrum and ensure scrum rules are followed, understood, and lived in the organization. The scrum master helps the scrum team perform at the highest level.

The main event in scrum is the sprint. Sprint is at the heart of scrum. Its timeboxed iteration and its duration range from one to four weeks. Sprint consists of

different events such as sprint planning, daily scrums, sprint reviews, and sprint retrospectives. Sprint planning aims to define the deliverables in the sprint and how to achieve the sprint goal. A great starting point for sprint planning is the product backlog. The team will meet every day in a daily scrum. The purpose of the daily scrum is for the development team to synchronize its activities, evaluate progress towards the sprint goal and create a plan for the next task. The sprint review event aims to review what have not been accomplished during the sprint. The last scrum event is the sprint retrospective. It will discuss future improvements.

Scrum artifacts allow scrum team members to have the same basic understanding of the product to be built. Each artefact contains a commitment whose progress can be measured. There are product backlogs, sprint backlogs, and increment as artifacts in the scrum. A product backlog is an ordered list of everything needed in the product. Product backlog item (PBI) contains features, functionality, requirements, and product improvement. In practice, PBI can be a user story, use case, feedback, or bug. It is a living artefact that will continue to grow along with business developments and input from users or stakeholders. Meanwhile, a sprint backlog is a collection of selected product backlogs that will work in the sprint. The developers updates the sprint backlog and completes it to achieves the sprint goals. Increment is a PBI that has been completed in a Sprint. It becomes a foothold towards the product goal and commitment refers to the definition of done.

2.3. Software Process Improvement

Software development process needs to be assessed to ensure that the process meets the criteria for good software development in accordance with proven practices. (Pressman & Maxim, 2014). Software process improvement (SPI) denotes activities undertaken to improve and direct the software process for the better. Organizations use SPI to accelerate software development process while maintaining good quality (Sommerville, 2011). Reducing software development time can be expected to reduce cost. The approaches that can be taken by using SPI include:

- Maturity process, which focuses on improving software development process by implementing software development best practices in an organization. Maturity process is carried out to evaluate whether software development best practices are adopted by the organization. Improvement of software development process aims to enhance the predictability of the process and the quality of the software being developed.
- Agile methodology, which focuses on iteration by developing software quickly and ensuring responsiveness to changes.

2.4. Scrum Maturity Model

Maturity is a method that can measure the maturity level and capability of

development management process. A maturity model gradually directs and improves the software development process in an organization. According to (Shen, 2021) maturity in project management is important to create success and get the strategic planning in agribusiness. He combined perspective of project management and agribusiness project management to create a maturity model indicator of agribusiness project management. There are environmental discernment, project planning, project implementation, project control, process closeout and integrated project management capabilities in maturity evaluation as basic indicator evaluation of project management maturity.

The scrum maturity model refers to two maturity models, namely the agile maturity model (AMM) and the capability maturity model integration (CMMI). CMMI is a collection of best practices that help organizations improve their processes. There are four of detail capability levels of CMMI , namely: CL0(incomplete), the organization has not been able to meet the specific goals of the overall process area. CL1(performed), the organization has met the specific goals of the overall process area, but with individual expertise. CL2(managed), the organization has met the specific goals of the overall process area through sub-units that have a planned process, but the organization does not yet have a standard. CL3(defined), the organization has met the specific goals of the process area as a whole and has standardized.

AMM is designed to improve methodologies and principle in software development using agile. There are 5 levels in AMM, namely initial, explored, defined, improved, and sustained. An organization at the initial level has not clearly defined the software development process by using agile. The main problems commonly found in organizations include development beyond schedule, communication problems, quality problems and cost of software development. The explored stage has indicated more structured and complete software development practices but is still mired with problems in internal communication, standard programming, integration practice, and development overtime and customer satisfaction. An organization that has focused on practices related to customer relationship management, pair programming, communication, testing and software quality has reached the defined level. An organization is at the improved level when it has collected detailed measurement of development process and practices as well as software quality measurements. Meanwhile, at the sustained level, an organization has routinely improved process through questionnaires on processes and tested ideas related to innovation and technologies.

A key process area (KPA) rating is used to calculate and assess the implementation of practices contained in the AMM. All goals in AMM will be transposed into questions. Any question can be answered by yes, partially, no and not applicable (N/A). The questionnaire data calculation uses an equation adopted from the AMM as follows (Chetankumar & Ramachandran, 2009):

$$\frac{\sum(Yn) + ½ \sum(Pn)}{\sum(Tn) - \sum(NAn)} * 100\%$$

Yn = number of "Yes" answers;

Pn = number of “Partially” answers;

Tn = total number of questions;

NAn = number of N/A answers.

Based on the calculation of the KPI rating, several categories of assessment will be obtained as follows:

1. **Fully Achieved**, which ranges from 86 % to 100 %. There are no significant weaknesses across organizational units.
2. **Largely Achieved**, which ranges from 51% to 85%. There is evidence of a sound and systematic approach to and achievement of the critical practices defined in the assessed KPA.
3. **Partially Achieved**, which ranges from 16% to 50%. There is evidence of a sound and systematic approach to and achievement of key practices defined in the assessed KPA. Certain aspects of achievement may be unpredictable.
4. **Not Achieved**, which ranges from 0% to 15%. There is little evidence of achieving key practices defined in the assessed KPA.

The scrum maturity model (SMM) is a framework resulting from the development of AMM that can be used by organizations in software development using scrum. SMM aims to help guide the process of developing or improving software development scrum gradually and independently. SMM has five maturity level with goals and objectives at each level as follows (Yin & da Silva, n.d.).

1. Level 1 - Initial

   Processes are generally slow, and the team has various issues delivering value. Success depends on the competence of team members. Level 1 is assigned if an organization does not meet the goals defined in level 2.

2. Level 2 - Defined

   The characteristics of a level 2 organization are following scrum practices in a tailored version. There are two main goals defined in level 2: basic scrum management and software requirement engineering. Basic scrum management evaluates the role of the scrum team, the presence of scrum artifacts, whether there is a scrum event, and whether sprints are executed correctly. Software

requirement engineering evaluates a clear definition of product owner's roles, product backlog management, and successful sprint planning meeting.

3. Level 3- Managed

At this stage, the unit has mastered the proper scrum framework based on the correct agile principles and values. There are two main goals defined in level 3: customer relationship management and iteration management. Customer management evaluates availability of product owners, successful sprint review meeting and whether the definition of done is explained clearly. Iteration management evaluates sprint backlog management, planned iteration, daily scrum success, and measured velocity.

4. Level 4- Quantitatively Managed

By this stage, scrum is institutionalized. The main goals of level 4 consist of standardized project management and process performance management. The main objective is to evaluate the uniformity of project management, analysis and measurement management.

5. Level 5- Optimizing

Level 5 is the highest in SMM with continuous self-improvement. The main goal for level 5 is performance management to measure and analyse all the self-improvement processes.

3. Research Methodology

This study utilized qualitative methodology through case studies and action research. The case studies included all project developments to support media monitoring and product analysis that implemented scrum. The study identified problems and collected data by conducting focus group discussions (FGD) with all scrum teams, direct observations, and collecting relevant data from internal project documents. The study compiled a list of questions used as a guide in conducting the FGD. The questions were collected from objectives and practices from level 2 (managed) to level 5 (optimizing) in the scrum maturity model. The research stages are presented in Figure 1.

1. Problem identification

A study of the project's internal documents and observations was conducted and it produced a formulation of research problems, questions, scope, benefits and objectives.

2. Literature study

From the formulation of problems, questions, scope, benefits, objectives and keywords, a literature study was carried out to obtain a definition and theoretical basis.

3. Formulation of FGD questions

Questions were arranged in accordance with the practices contained in the SMM. These practices would be changed and adapted to comply with the scrum rules in the 2020 scrum guide document. The modified practices were arranged in the form of questions. In addition, the list of questions that had been compiled would be answered by structured interviews through focus group discussion (FGD) activities with all members of the scrum team.

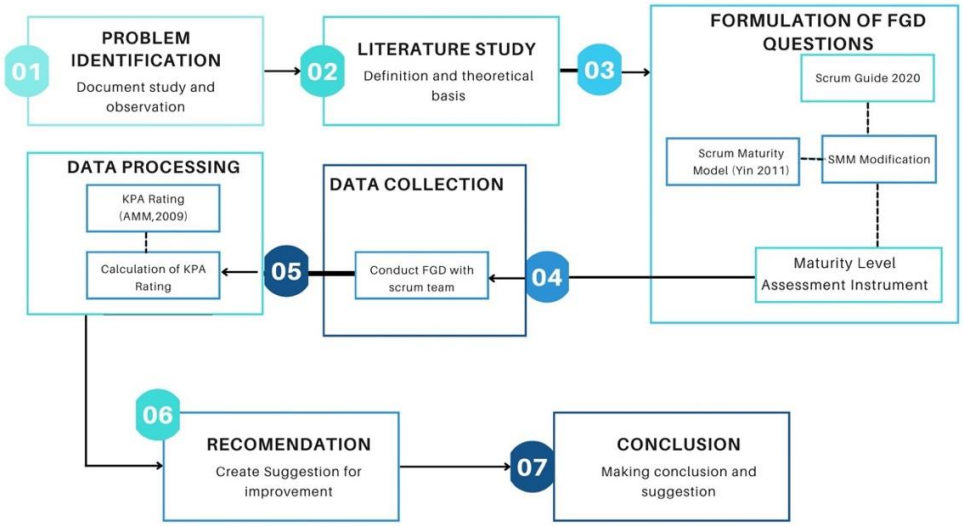

Fig.1: Research Method

4. Data collection

To collect study data, focus group discussions were organized with all the scrum team members. Project internal documents were used as supporting data to answer research questions. During FGDs, scrum members discussed and answered each question with "yes" if the practice in question is to be implemented thoroughly, "partially" if the practice in question is implemented partially, "no" if there is no implementation of the practice in question, and "not applicable" if the practice in question is not possible to be implemented. This stage produced questions that had been answered.

5. Data processing

From the results of data collection, data processing was subsequently carried out to measure the maturity level of the scrum implementation using the KPA rating equation in AMM. Consequently, the maturity level of the software development process at PT XYZ was obtained.

6. Recommendation

Based on the results of the maturity level obtained in the previous stage the researcher will map the maturity level with proposed improvements based on the 2020 scrum guide. This stage will produce recommendations for improving the

software development implementation process at PT XYZ which can be implemented in the next project or sprint. The proposed improvement is based on the answers to the questions that are rated "No", "Partial" and "Not Applicable" which are at level 2 and level 3 of the Scrum Maturity Model.

Each answer is then mapped with practices that should be carried out according to the scrum best practices contained in the 2020 scrum guide in order to achieve a higher rating value at level 2 and level 3 of the Scrum Maturity Model.

7. Conclusion

The final step is to draw conclusions and research suggestions based on the maturity level of the software development process. The findings and recommendations are based on the questions and scope of the research.

4. Results and Discussion

4.1. Assessment Results

The assessment and evaluation of the maturity level of the software development started with processing the data obtained from the focus group discussions with scrum team members. Data processing began by calculating the KPA rating of each objective in the scrum maturity model. The results of all objectives were recapitulated so that they would produce a maturity level of software feature development.

4.1.1. Assessment Results for SMM Level 2 – Defined

The assessment of SMM maturity level 2 consists of two goals, namely basic scrum management and software requirement engineering. Table 1 shows the assessment results on the goals of basic scrum management.

Table 1: Scrum Assessment Results in Basic Scrum Management Goals

Objectives	Rating
There are scrum roles	100%
There are scrum artifacts	66.66%
There are scrum meetings	60%
Sprint is implemented correctly	75%
Average rating	75.41%
Interpretation	Largely Achieved

Table 1 highlights that the average rating obtained for all objectives in basic

scrum management is 75.41 % (largely achieved). The result indicates that most of the basic scrum management goals have been achieved. In the existing product development, there were scrum roles which included a product owner, a scrum master, and a group of people as developers. For scrum artifacts, there were product backlogs and sprint backlogs which were updated by the product owner in the Trello application. Meanwhile, there were practices that were not completed or partially completed because the team had not implemented the release burndown chart and sprint burndown chart. The release and sprint burndowns were not updated following the developments reported by the team. The organization conducted a daily scrum, and all developers attended the daily scrum. However, other events such as the sprint planning and sprint review were still partially carried out, and the sprint retrospective and release planning meeting had never been organized.

Table 2 shows the assessment results and recapitulation of software requirements engineering goals in level 2 of the scrum maturity model.

Table 2: Scrum Assessment Results in Software Requirement Engineering Goals

Objectives	**Rating**
The definition of product owner is clear	91.66 %
Product backlog management	62.5 %
Successful sprint planning meeting	66.66 %
Average rating	73.61 %
Interpretation	Largely Achieved

Table 2 describes that the average rating obtained for all objectives in software requirement engineering is 73.61 % (largely achieved), suggesting that most of the basic goals of software requirement engineering have been accomplished. In the existing product development, product owners did not always communicate product backlog items clearly. For some cases, product owners did not have power to set priorities and had to follow the mandates of top management. For product backlog management objectives, all scrum team members carried out product backlog estimation, but the top item could not be estimated well in some sprints. Sprint planning generated a product backlog, but not all team member participated in the sprint planning of certain sprints.

4.1.2. Assessment Results for SMM Level 3 – Managed

The assessment of SMM maturity level 3 consists of 2 goals, namely customer

relationship management and iteration management goals.

Table 3 presents the assessment results and recapitulation of customer relationship management goals.

Table 3 describes that the average rating obtained for all objectives in customer relationship management is 55.55 % (largely achieved). In the existing product development, there was no definition of done for all projects. The team only had acceptance criteria for certain tasks. The product owner was not always there for the team due to work commitment on another project. In addition, the product owner also assumed the role as the developer, hence the team could not always contact the product owner. In the sprint review event, feedback was always received from stakeholders and product owners.

Meanwhile, some of the increments displayed in the sprint review event did not function properly, and some did not meet the needs of users. There were some untested backlogs before being presented in the sprint review.

Table 3: Scrum Assessment Results in Customer Relationship Management Goals

Objectives	**Rating**
There is a definition of done	50%
Product owner is available	50%
Successful sprint review meeting	66.66%
Average rating	55.55 %
Interpretation	Largely Achieved

Table 4 shows the assessment results and interpretation of iteration management goals in SMM Level 3.

Table 4: Scrum Assessment Results in Iteration Management Goals

Objectives	**Rating**
Sprint backlog management	64.28%
Planned iteration	62.5%
Measured velocity	66.66%
Successful daily scrum	50%
Average rating	60.86%
Interpretation	Largely Achieved

Table 4 describes that the average rating obtained for all objectives in iteration management is 63.57 % (largely achieved). In the existing product development, team members could easily access the sprint backlogs on Trello application, but not all team members updated the sprint backlog on a daily basis. For the objective planned iteration, the sprints did not always end on time, not all failed iterations could stop early, and not all the work carried out by the team followed the commitments agreed in the sprint planning. The daily scrum is held every morning at 09.00 a.m. and is always attended by all developers. However, the daily meetings sometimes exceeded 15 minutes. The burndown chart was not always updated based on developer progress reports.

4.1.3. Assessment Results for SMM Level 4 – Quantitatively Managed

The assessment of SMM maturity level 4 consists of two goals, namely standardized project management and process performance management. Table 5 shows the assessment results in standardized project management goals.

Table 5 describes that the average rating obtained for all objectives in standardized project management is 50 % (partially achieved). In the existing product development, some projects were not managed in accordance to the goals, objectives, and practices contained in SMM levels 2 and 3.Table 6 shows the assessment results in process performance management goals.

Table 5: Scrum Assessment Result in Standardized Project Management Goals

Objectives	Rating
Quantitative project management	50%
Average rating	50%
Interpretation	Partially Achieved

Table 6 describes the average rating obtained for all objectives in performance management as 50 % (partially achieved). In the existing product development, all measurement results at SMM levels 2 to 4 were not monitored and managed based on sustainability in some projects.

Table 6: Scrum Assessment Results in Process Performance Management Goals

Objectives	Rating
Measurement and Analysis	50%
Average rating	50%
Interpretation	Partially Achieved

4.1.4. Assessment Results for SMM Level 5 – Optimizing

Level 5 scrum maturity model must achieve performance management goals. Table 7 shows the implementation assessment results and interpretation for performance management goals.

Table 7: Scrum Assessment Results in Performance Management Goals

Objectives	Rating
Success sprint retrospective	0%
Positive indicator	50%
Average rating	25% %
Interpretation	Partially Achieved

Table 7 describes that the average rating obtained for all objectives in performance management is 25 % (partially achieved). In the existing product development, all practices related to the sprint retrospective were not implemented. The sprint retrospective was never implemented by product owners, it always took more than 3 hours, the lessons learned were never recorded, and not all team members had high energy levels during the sprint. Some stakeholders did not have high satisfaction and some of developer often had to work overtime and not conducted experiments to improve the scrum process. The overall maturity level assessment for each of the previously discussed goals is shown in Table 8.

Table 8: Recapitulation of Rating Results and Overall Interpretation of Goals.

Level	Goals	Rating	Interpretation
1	-	-	-
2	Basic scrum management	75.41%	Largely Achieved
	Software requirement engineering	73.61%	Largely Achieved
3	Customer relationship management	55.55%	Largely Achieved
	Iteration management	60.41%	Largely Achieved
4	Standardize project management	50%	Partially Achieved

Level	Goals	Rating	Interpretation
	Process performance management	50%	Partially Achieved
5	Performance management	25%	Partially Achieved

Table 8 shows that the average rating obtained for all objectives on the goals at levels 2 and 3 in the scrum maturity model is largely achieved. Meanwhile, at level 4 and level 5, the interpretation is still partially achieved. At level 2, the basic goal scrum management obtained a rating of 75.41%, while software requirement engineering achieved 73.61 %. Goal customer relationship management and operation management at level 3 obtained a rating of 63.88 % and 53.57%, respectively.

Based on the measurement results in Table 8, it can be concluded that the practice of implementing software development projects using the scrum framework is still at maturity level 1 (initial) and has not reached the maturity levels 2 and 3. This is evidenced by the fact that the KPA rating at level 2 and level 3 for each goal is still below 86% or in other words, it has not yet reached the interpretation of fully achieved and that some improvements are still needed in the scrum implementation process at PT XYZ prior to proceeding to a higher level of maturity. The results of the assessment shows that there is several unrealized practices of scrum. The recommendations for improvement focus on levels with “largely achieved” interpretations (level 2 and level 3). A gap was found that distinguished the assessment results and the maximum value of the SMM level 2 and level 3. The recommendation for improvement refers to the latest version of the scrum guide.

4.2. Improvement Recommendations

From the "partial" and "no" answers, the study compared the current practice with the practice that should be carried out in accordance with the scrum rules (referring to the 2020 scrum guide document). Table 9 describes the maturity level of the KPA rating percentage in each objective currently at level 2 and 3 (as is) and the desired maturity level by PT XYZ (to be) with KPA rating of “fully achieved”.

Table 9 Existing vs Expected Maturity at Level 2

No	Code	Objective	As is (%)	To be (%)
1	L2.1	There are scrum roles	100	100
2	L2.2	There are scrum artifacts	66.66	86
3	L2.3	There are scrum meetings	60	86
4	L2.4	Sprint is implemented correctly	75	86
5	L2.5	The definition of product owner is clear	91.66	100
6	L2.6	Product backlog management	62.5	86
7	L2.7	Successful sprint planning meeting	66.66	86

For the objective of "there is a role in scrum", PT XYZ has achieved the KPA rating of “fully achieved” (KPA rating = 100%). The objective “a clear definition of product owner” has also achieved the KPA rating of “fully achieved” (KPA rating = 91.66%). Therefore, for these two objectives, PT XYZ targets an increase in the KPA rating by 100% (the maximum value of “fully achieved” rating), as well as pursuing “fully achieved” interpretation with a KPA rating of 86% (the minimum value of “fully achieved” interpretation) for other objectives. The comparison of the current KPA rating value with the target KPA rating to be achieved is illustrated in the radar diagram as shown in Figure 2.

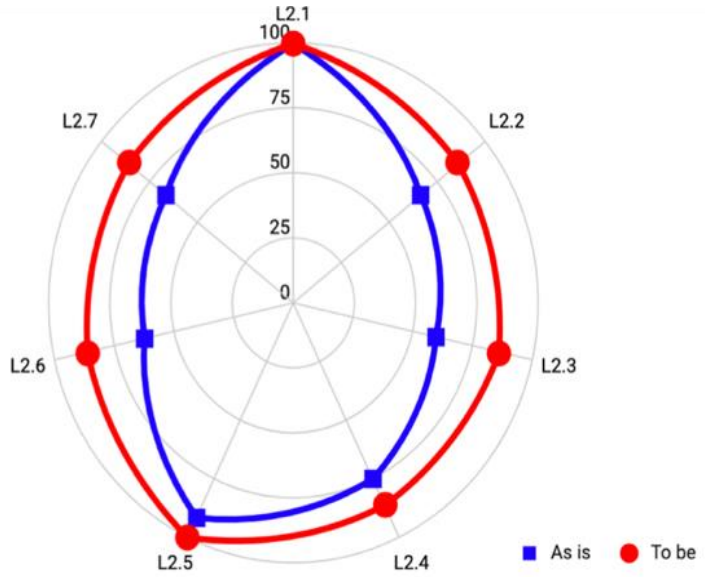

Fig. 2: Representation SMM at Level 2

Table 10 below describes the maturity level of the rating percentage in each objective at level 3 (as is) and the desired maturity level by PT XYZ (to be) with the KPA rating of "fully achieved".

Table 10 Existing vs Expected Maturity at Level 3

No	Code	Objective	As is (%)	To Be(%)
1	L3.1	There is a definition of done	50	86
2	L3.2	Product owner is available	50	86
3	L3.3	Successful sprint review meeting	66.66	86
4	L3.4	Sprint backlog management	62.5	86
5	L3.5	Planned iteration	62.5	86
6	L3.6	Measured velocity	66.66	86
7	L3.7	Successful daily scrum	50	86

Based on the table above, it can be concluded that the interpretation of the achievement for level 3 has not reached "fully achieved". This is evidenced by the fact that all objectives contained in SMM level 3 are still below 86%. The interpretation target to be achieved is the "fully achieved" interpretation with a KPA rating of 86% (the minimum interpretation value of "fully achieved"). The comparison of the current KPA rating value with the target KPA rating value to be achieved is illustrated in the radar diagram as shown in Figure 3.

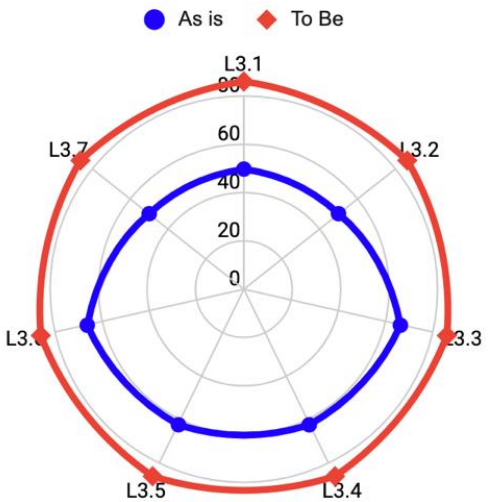

Fig. 3: Representation SMM at Level 3

The study made proposed improvements based on practices still not and partially implemented in levels 2 and 3. The preparation of proposed improvements refers to the 2020 scrum guide. The proposed recommendations are categorized into basic scrum components, namely scrum roles, events, and artifacts, as follows.

4.2.1. Scrum Roles

A typical scrum team consists of 3 roles, namely product owner, scrum master and developer. Ideally, a scrum team should consist of one product owner, one scrum master, and several developers. Therefore, it is not recommended that the role of product owner is concurrently assumed with the role of developers or scrum master. Several recommendations for the improvement of each scrum team are presented in Table 11.

Table 11 Improvement Recommendation for Scrum Roles

Scrum Roles	Improvement Recommendation
Product owner	Product owner is responsible for determining product direction and ensuring that product backlog is still in line with product goals. Product owner must always available for the team and easily contactable.
Scrum Master	Scrum master should conduct routine analysis and evaluate current scrum implementation.
Developers	Developer must be committed to what has been agreed in the sprint planning and must know other developers and what others are doing in achieving product goals.

4.2.2. Scrum Events

All scrum team members must commit to working during the sprint that has been agreed upon in the sprint planning and not being interrupted by other tasks. The team must be committed to completing the work in the sprint. During the sprint, the scope may change but the changes made should not affect the sprint goal. The sprint can be cancelled without waiting for the sprint duration to end if the sprint goal is not suitable. Recommendations for improvement in scrum events are presented in Table 12.

Table 12 Improvement Recommendation for Scrum Event

Scrum Event	Improvement Recommendation
Sprint planning	• Sprint planning must be carried out prior to the start of the sprint and carried out collaboratively by all members of the scrum team. It is possible to invite external experts in the technical field or potential software users. • Sprint planning must produce a sprint backlog and sprint goal.
Sprint review	• Sprint review must be conducted after the sprint ends

	and the event must produce a review of the latest increments and product backlog. • Mandatory inputs from the event include product backlog, increments and sprint goals. The scrum team must therefore ensure that all aspects are functioning and tested before implementing sprint planning. • The meeting must be attended by all scrum members and attended by stakeholders.
Sprint retrospective	• A sprint retrospective is mandatory after the sprint ends and ensures that all scrum team members (developers, product owners and scrum masters) are present. • It must be ensured that the event results in the improvement in commitment for the following sprints (including but not limited to adding new points in the definition of done)
Daily scrum	• It must be conducted with focus to avoid irrelevant discussions. • Daily scrum meetings are limited to a maximum duration of 15 minutes. • Consistent venue and schedule will better facilitate the team in the daily scrum implementation process.

4.2.3. Scrum Artefacts

Recommendations for improvement in scrum artifacts are presented in Table 13.

Table 13 Improvement Recommendation for Scrum Artefact

Scrum Artifact	Improvement Recommendation
Product backlog	Product owner must have a vision of the product, including features requested by other teams. Product owner should conduct periodic evaluations and updates on the product vision so that the products developed meet business needs and provide values. Each product backlog has the velocity to measure the number of tasks a team can deliver during a sprint.
Sprint backlog	All items in the sprint backlog must have estimates. Estimates can be measured based on complexity, resources, and capabilities. The improvement recommendations for sprint backlog include: • ensuring that the sprint backlog consists of only one sprint goal (The sprint backlog must be managed and updated daily. In the event of a change, the scrum team must collaborate to discuss the scope of the

Scrum Artifact	Improvement Recommendation
	sprint backlog without affecting the sprint goal). • ensuring that the scrum team knows that it can use the release burndown chart and the sprint burndown chart to monitor progress during the sprint, and that the chart must be updated daily according to the progress of developers.
Increment	Work is considered to be an increment if it meets the definition of "done" since it is necessary to create transparency by defining "done" and all members of the scrum team must follow the standards contained in the definition of "done". A list of the definitions of "done" is necessary. It is collection of criteria for a project to be considered done and ready to be accepted by users.

5. Conclusion

This research was conducted to evaluate the scrum implementation process in the development of software projects at PT XYZ and provide recommendations for improvement as a proposal to improve the implementation of scrum practices in accordance with scrum rules. Based on the study, the following conclusions are drawn:

1. The maturity level of the scrum implementation process at PT XYZ is still at level 1 (initial). This is evidenced by the fact that the overall goals achieved at level 2 have not yet received "fully achieved" interpretations. The basic scrum management goal is still at the "largely achieved" level with a KPA rating of 75.41%, while the software requirement engineering goal is also in the "largely achieved" category with an interpretation of 73.61%.
2. Recommendations for improvement are provided based on the answers "partially", "no" and "not at all" in each question defined at level 2 and level 3 and the recommendations for improvement are grouped based on scrum team, scrum event and scrum artifact. For organizations in which case studies are conducted, research results can be used to evaluate software development. Organizations can implement improvement proposals offered by this study to overcome software development delays and increase the level of software development maturity. For further research, the company can review the software development process on other scrum teams on other products. Offering suggestions for improvement based on the latest scrum guide can also be combined with advice or recommendations from scrum experts.

The results of this study can be used by PT XYZ as an evaluation material for

the development of software projects that are currently implemented. Companies can also implement the recommendations for improvements to address present issues in software development and increase the maturity level of scrum implementation in subsequent projects or sprints. Suggestions for improvement at level 2 and level 3 compiled by this study can be used as a guide for other organizations that adopt scrum in the process of implementing software projects in accordance with organizational needs.

Further research can review software development process in other teams at PT. XYZ and it is necessary to conduct a study to determine the competency maturity level of the scrum team, namely product owner, developer and scrum master.

References

Beck, K., et al., (2001). Manifesto for Agile Software Development.

Betta, J., Chlebus, T., Kuchta, D., & Skomra, A. (2019). Applying Scrum in the New Product Development Process. Advances in Manufacturing II, - Volume 1, LNME, pp. 190–200, 2019.

Chetankumar, P., & Ramachandran, R. (2009). Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices. International Journal of Software Engineering (IJSE), 3-28.

C.V. Weber, M.C. Paulk, C.J. Wise, and J.V. Withey. (1991) Key Practices of the Capability Maturity Model, Software Engineering Institute, CMU/SEI-91-TR-25, ADA240604.

Fowler, M., & Highsmith, J. (2001). The Agile Manifesto. Software Development Magazine.

Gambo Kasuwar Kuka, M. (2018).Product Development and Management Strategies. Product Lifecycle Management - Terminology and Applications.

Kulpa, M., & Johnson, K. (2008). Interpreting the CMMI: A Process Improvement Approach, Second Edition. Auerbach Publications.

Noll, J., Razzak, M. A., Bass, J. M., & Beecham, S. (2017). A Study of the Scrum Master's Role. Lecture Notes in Computer Science, 307–323. doi:10.1007/978-3-319-69926-4_22

Patel, C., & Ramachandran, M. (2009). Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices. International Journal of Software Engineering.

Pressman, R. S. (2011). Software engineering: a practitioner's approach 5th ed. New York: McGraw-Hill.

Pressman, R. S., & Maxim, B. (2014). Software Engineering: A Practitioner's Approach, 8th International edition. 8. McGraw-Hill Science/Engineering/Math.

Shen, J.(2021). An Exploratary Study on Project Management Maturity Assessment for Agricultural Enterprises. Journal of System and Management Sciences, Vol.11, No. 3, 235-245

Sommerville, I. (2011). Software Engineering (Edition 9). Massachusetts: Addison Wesley.

Rubin, K. (2019). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Canada: Addison Wesley.

Schwaber, K., & Sutherland, J. (2020). The Scrum Guide (The Definitive Guide to Scrum: The Rules of the Game). Creative Commons.

Schwaber, K., & Sutherland, J. (2020, November). The Scrum Guide. Retrieved from Scrum Guide: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf

Schwaber, K., & Sutherland, J. (2017, November). The Scrum Guide. Retrieved from Scrum Guide: https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf

Schwaber, K. Scrum development process. In Business Object Design and Implementation; Springer: Berlin/Heidelberg, Germany, 1997; pp. 117–134. ISBN 354076096

VMedu. (2016). A Guide to the Scrum Body of Knowledge (SBOK Guide). Phoenix: SCRUMstudy.

Yin, A. (2011). Scrum Maturity Model. Instituto Superior Técnico, Engenharia Informatica e de Computadores.: Lisboa: Instituto Superior Técnico.