THE SCRUM GUIDANCE FOR IT SERVICE SUPPORT DOMAIN AND SQA TO GENERATE SUCCESSFUL AND QUALITY SOFTWARE

Rahma Fitria

Sistem Informasi Universitas Malikussaleh Lhokseumawe
Jl. Cot Tgk Nie-Reulet, Aceh Utara, 141 Indonesia
email: rahmafitria@unimal.ac.id

***Abstract*—** The article is presenting how to implement the framework of scrum in IT service support domain and software quality assurance (SQA). Basicly, scrum framework is being used in other software development methods such as agile, but the reseach found that there is merely a few could support the software development. This framework could be implemented in small, medium and large projects. By promoting the usage of scrum framework could add more valu to organization and reach its goal. The collection of scrum framework implementation from previous researcher could present the real example of scrum framework in IT service domain and SQA. As this framework is still new in practice to IT domain, so that the guidance of this scrum framework will help developer to have more experience in applying in any software development projects. As the result, the company will enhance their business value.

I. INTRODUCTION

Software Development Process becomes complicated once producing complex software that need frequently changes on user requirement. Applying *Scrum* Methodology on this kind of product is very helpful that embody simplicity, flexibility, accuracy, and responsiveness toward the process. *Scrum* also applies very good team structure that consists of 3-9 experts that do their work correctly and accordingly. It can be concluded that small *team member* can work better than larger teams. In this paper will also apply *scrum framework* in *IT service support domain* to control fast changing on user requirement and develop a good team structure as it has applied in software development. Deploying successful and quality software is very crucial in IT project management that can meet user requirement. This *scrum* method is used to adapt changes from user requirement. The purpose of this study is to

guide people how to use scrum method and describe all attributes, roles, principles, framework that has been approached.

II. SCRUM GUIDE

Scrum Definition

Scrum is one of *agile methodology* that is designed for complex products to have clarity and transparency in planning and implementation phase with the higher possible value. According to Schwaber and Sutherland (2011) *Scrum* can be implemented for small, mid-sized and large software development. Scrum is lightweight, easy to comprehend, but intensely hard to master (Schwaber & Sutherland, 2011). Once scrum is rightly implemented, it will increase speed of software development process.

Scrum Framework

The scrum framework is composed of *scrum teams* and their associate *roles, events, artifacts,* and *principles* (Schwaber & Sutherland, 2011). *Scrum framework* has their own components and particular goal to determine a successful scrum work.

Scrum Theory

Scrum is invented on practical process control approach. It states that knowledge grows from involvement of practice and decides a solution based on the knowledge and experience gained.*Scrum* conducts an iterative, incremental approach to maximize resemblance and restraint troublesomeness. Based on Schwaber and Sutherland, (2011) there are three vital elements support every development of practical process control which is:

- *Transparency*; the process of important elements must be obvious to those engaged to the result.
  Transparency needs those elements to be described by a common principle so teams share a common comprehension of what is being observed.
- *Inspection*; *Scrum* developer must regularly go through *scrum artifacts* and move forward to discover offensive bugs.
- *Adaption*; once result of the product are unacceptable, the progress or the component being developing must be adapted. This kind of modify must be quickly conducted to decrease additional change.
- *The scrum team*; Product owner, team members, and a scrum master are called *scrum team*. There are two divisions in team which are *self-*

organizing and cross-functional team. Self-organizing team will organize the team to complete the work well. *Cross-functional team* is complete the work independently. So this kind of this team model can maximize flexibility, creativity and productivity.

- *Product owner* has responsibility for optimizing quality of product and the obligation of the team member.
- *Team member* (developers, testers, and others) are people who are professional in work that produce product.
- *The scrum master*
  The scrum master is a person who has responsibility to ensure scrum is understand and enacted.

Scrum Events

According to Schwaber and Sutherland (2011) scrum create event regularly, so that it can ensure appropriate amount of time is spent in the planning process. This event is created to allow critical *transparency* and *inspection.* Lack of success to add any of these events effect in decrease *transparency* and is a missed chance to examine and adjust.

Scrum Sprints

The most significant part in scrum is a sprint (Schwaber & Sutherland, 2011). During development process, sprints have regular time that is a new sprint begins right away after the previous sprints have been concluded. Based on Schwaber and Sutherland (2011) *sprints* contain the *Sprint Planning Meeting*, *Daily Scrums*, *the development work*, *the Sprint Review*, and *the Sprint Retrospective*. *Sprints* are employed to attain something. Each *sprint* described of what is to be developed, a blueprint and pliable arrangement that will lead developing it, the progress, and the end product.

Scrum Artifacts

As Schwaber and Sutherland (2011) s*crum's artifacts* depict task in several parts. It is valuable in equipping *transparency* and favorable circumstances for *inspection* and *adaption*. *Scrum artifacts* are created to optimize facts required. So that the scrum teams assure that the product is conveyed a "Done" increment favorably.

Product Backlog

Based on Schwaber and Sutherland (2011) *Product Backlog* is a sole source of product requirement that is used when the product need to conducts some changes. The product owner has responsibility on product backlog which is the content, availability, and ordering. The Product Backlog records all attribute, functions, requirements, improvements, and adjusts changes to the product when it is need in future release. *Product Backlog* lists has features like definition of product, scheme, and approximate calculation.

Sprint Backlog

According to Schwaber and Sutherland (2011) *Sprint Backlog* is the series of *product backlog* lists that is use for deploying the product increment. The sprint backlog lists the functionality of the next increment of product and records how that functionality will be conducted.The Sprint Backlog describes the work team members which are developers, testers, and others that will accomplish to revolve Product Backlog lists into a "Done" Increment. Only the Team Members (developer, testers, others) can do changes to*Sprint Backlog* during a *Sprint. The Sprint Backlog* is a tremendously viewable, real-time activity of developers, testers and others that plan to perform during the sprint.

Definition of "Done"
Once the *product Backlog* lists or an increment is defined as "Done" by team members, it means for works to be complete, assure transparency, and evaluate when work is finish on the product increment (Schwaber & Sutherland, 2011).

III. PRACTICAL TEAM STRUCTURE

According to Mundra et al. (2013) there are three significant concepts often used in *scrum* that is *product backlog*, the *sprint* and the *scrum team.* The *product backlog* can be defined as the series of requirements to be converted into *sprints.* The *sprint* is an iterative process during product development. This SCRUM methodology concept can be considered that the analysis, design, and development activities are changeable and unpredictable. In the real world most of scrum challengers have difficulty during development process. Nevertheless, once the team conducts the process, then the productivity is growth as the team generates various sprints until the product is assumed complete for deployment. In conclusion, the scrum team conducts the backlog, execute the backlogs, and generate several sprints that are frequently

given comments by *product owner* and users. Normally the *scrum team* is a small team which consists of 3-9 professional. Inside that small team, it can be divided to other two or three team. It depends how the behavior of the organization does.

SCRUM of SCRUMS

According to Mundra et al. (2013) *Scrum of scrums (SOS)* is a several *scrum teams* handling on the similar product development. When small scrum team work with large product development, the team can break the product functionality to smaller streams and create other different scrum teams. Scrum of scrum also can be used if scrum teams are in several locations and develop the product in the same time.

IV. APPLYING SCRUM FRAMEWORK IN IT SERVICE SUPPORT DOMAIN

According to Shalaby and El-Kassas (2011) *Sprint Planning* should distinct in the *IT service support domain*, as the domain nature is frequently changeable. The new proposed way for sprint planning in IT service support domain is named Dynamic Sprint Planning (Shalaby & El-Kassas, 2011).

Vertical vs. Horizontal Scrum Team Formation

Based on Shalaby and El-Kassas (2011) *Vertical Scrum Team* is formed to have several support levels which are from *level1 support*, *level2 support*, and *level3 support*. This level is adjusted with the *scrum team theory* which is *cross-functional* that integrate various levels and scope of information. The *scrum team* is formed various level of support to provide high pace of knowledge sharing with other levels. The *cross-functionality* of the team member is prominently encouraged to have various level of knowledge specialization.

Dynamic Sprint Planning

In the area of IT service support, it arise intervention and changes during sprint activities in early stage. Once the sprint goes on and the changes are accepted, then the changes will be put to the next sprint. This action can affect the customer satisfaction and may bring to a loss in public stock. In other side, if the sprints are abort, it may get some issues with all the sprints (Shalaby & El-Kassas, 2011).

The *dynamic sprint planning* proposed the *sprint* not with the up speed volume of the team, but the estimation of ancient data from the domain, based on scrum *team members*' knowledge, the level of effort of the several routine

urgent dynamic issues (Shalaby & El-Kassas, 2011). The issue may arise once the team members overestimate for one of the sprints. So that one item of product backlog can be expanded into similar tasks without occupying the whole of sprint. In case the team members complete the sprint backlog tasks, but find overestimation of the level of effort of the urgent dynamic issues, so that the next blueprint sprint tasks can be catered to the current sprint orderly.

Other issue is if the sprint is underestimating, the scrum team member can ask other team to assist them or ask the knowledgeable person from out of scrum team. If the issues aren't solved, then the sprint will not send the attached deliverables. So the sprint retrospective should overwhelm this kind of issue in the next. As Shalaby and El-Kassas (2011) the dynamic *sprint planning* is depicted as below:

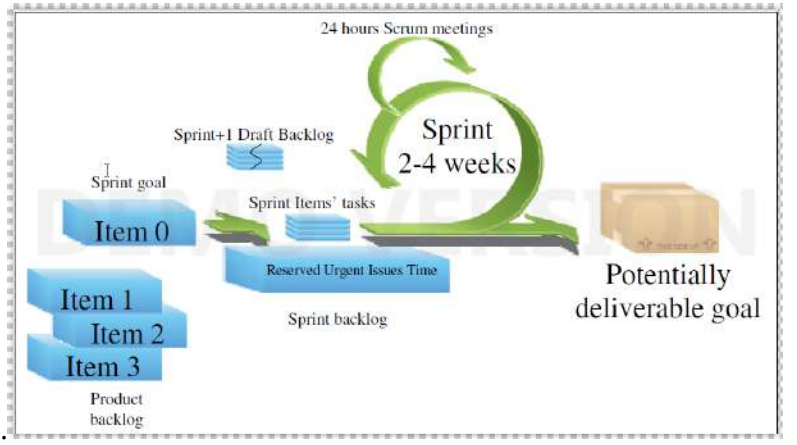

Figure 1.1: Dynamic Sprint Planning

Based on the figure derives abovethe *sprint* is estimated according data history from the domain, team members experiences and several daily *urgent dynamic issues*. For *urgent dynamic issues* we allocate time in *sprint planning* before starting the *normal sprint planning*. After that we proceed to load in the *sprint* with the whole of the *scrum team*'s speed. Once the team completes the *sprint backlog tasks* and has over estimation of tasks, then we can supply from the expected drafted *sprint tasks* into the existent tasks orderly. Once these phases in the *dynamic sprint planning* to be completed, it can be deployed in the actual environment.

V. IMPLEMENTING SOFTWARE QUALITY ASSURANCE (SQA) IN SCRUM

According to Khalane and Tanner (2013) the "lack of Concrete Guidance" in *scrum* push the project team to conceive *innovations* in order to overwhelmed the challenges and enhances the performance to reach user expectations. The next part describes the *innovations* by the project team and twist around this with the literature on *method tailoring* to alter *Scrum* for *software quality assurance* objectives.

1. Innovations

The team members should conceive the innovation of project that involve in designing the *Process Structure, Adopting SQA practices* from other methodologies, and perform *Guiding Principles of Collective Ownership, Constant Feedback,* and *Continuous Improvement*. This innovation brings empowerment, feeling of ownership, and adaptability to various environments (Khalane &Tanner, 2013).

1.a) Process Structure

The *process structure* defines the process, stages, approach, and roles of team members to change completely all the requirements on the product backlog that list all the project requirement. *Process structure* represents a team's "Definition of Done" on the activities, the roles, and the tools (Khalane & Tanner, 2013).

1.b) Adopted Practices

The process structure must be designed well, so that other methodology lie XP can be integrated. This scrum methodology can help developer to do coding rapidly with a good feature code if it is all written with a good structure (Khalane &Tanner, 2013).

1.c) Guiding principles

According to Khalane and Tanner (2013) the principles that disclose of process structure and adoption of practices are; *Collective Ownership, Constant Feedback,* and *Continuous Improvement.* Collective ownership is significant to have quality requirements and programming standard can enhance technical quality. In addition, constant feedback can be used to meet user expectation. Eventually, continuous improvement was conducted by Scrum Master that led to integration practices that control constant feedback and collaborative ownership of activities.

2. Method Tailoring

According to Khalane and Tanner (2013) organization require to tailor scrum process innovatively that provide SQA practices to be involved in scrum process structure. A lack detail of success to scrum package on technical SQA practices may occur to several challenges. Adopting practice into scrum is in path with one flow from research in software development that emphasis on tailoring method for a suitable development phase.

VI. CONCLUSION

The paper proposed presents the scrum guidelines and its characteristics. It also describes the team structure of scrum that can be applied for software development. Consider this as guidelines to use scrum methodology either on software development or on IT service support domain. Eventually the paper also briefly provides SQA practices to meet user expectations that can enhance business value by producing quality software.

VII. REFERENCES

Khalane, T., & Tanner, M. (2013). Software quality assurance in scrum: The need for concrete guidance on SQA

strategies in meeting user expectations. *International Conference on Adaptive Science and Technology (ICAST)*,1-6. doi:10.1109/ICASTech.2013.6707499

Mundra, A., Misra, S., & Dhawale, C.A. (2013). Practical scrum-scrum team: Way to produce successful and quality software. *13th International Conference on Computational Science and Its Applications (ICCSA)*,119-123. doi:10.1109/ICCSA.2013.25

Schwaber, K., & Sutherland, J. (2011). *The definitive guide to scrum: The rules of the game*. Retrieved September 24, 2014 from http://www.scrum.org/

Shalaby, M., & El-Kassas, S. (2011). Applying scrum framework in the IT service support domain. *IEEE Asia-Pacific*

Services Computing Conference,9-15. doi:10.1109/APSCC.2011.76