November-December 2019
ISSN: 0193-4120 Page No. 3067- 3073

SMMS: Document Management in Agile Model for Software Maintenance

[1]Pathiah Abdul Samat, [1]Mohamed Zulkifli Mohamed Nawi
[1]*Faculti Sains Komputerdan Teknologi Maklumat, Universiti Putra Malaysia, 43400 Serdang Selangor.*
pathiah@upm.edu.my

Article Info
Volume 81
Page Number: 3067- 3073
Publication Issue:
November-December 2019

Article History
Article Received: *5 March 2019*
Revised: *18 May 2019*
Accepted: *24 September 2019*
Publication: *14 December 2019*

Abstract:
Study shows that most challenging task in software development life cycle is software maintenance, because all of changes made must documented for future used. Thus, an efficient and newest documentation is vital in software maintenance to ensure its activities will not jeopardized. However, this is not as we thought. The popular models such as agile development has ruined the importance of maintenance tasks in software development life cycle. The agile models focused only on verbal communication within development team to provide faster development where the information of the product and its features exists within the heads of the developers but any changes will not documented in a design document. Therefore, the major problem of Agile models is the absence of latest documentation and this issue supposed not exist in software maintenance which is depend on previous documentation. Thus, our aim is to add document management in agile model for software maintenance processes. The result shows that our concept is successful and useful for software maintenance.

Keywords: *Software Maintenance; Agile Model; Documentation, Software Development Life Cycle*

1. INTRODUCTION

Software maintenance consists crucial activities, which the result of changes is to ensuring its maintainability and evolution. On the other side, agile software development is a sort of software in which it method is to blend the collaboration among self-organizing, cross-functional groups.

There are many issues in software maintenance after the development accomplished. Software maintenance is costly activity that consumes a major portion of the cost of the total project [1]. The time used and effort needed to review the defects at this phase consumes about 40%-70% of the cost of the entire life cycle. In other words, software maintenance is the changes made to the computer program after they have been deploy to user or customer. Thus, the software maintaining is keep evolving as long as there are change request from shareholder. To ensure the software keep evolving and maintainable, the software must always in the latest documentation. Nowadays, there are many methods to guarantee that all of software through the processes of verification, testing, debugging and validation. However, the active change request during software development not guarantee the documentations are up to date including the popular method such as Agile.

The strength of agile methods are raise adaptive preparation, evolutionary development, fast distribution, ongoing improvement, and inspires rapid and flexible response to change. The flexibility [2] provided by Agile method caused it becomes a popular approach in software development. Amongst the flexibility includes helping in software design, assisting in getting fast feedback from customer, face-to-face communication and better quality focus. Agile method is very effective method in software development. However, agile is an inefficient method in software maintenance because it not guarantee updated documentation from previous

phase. Therefore, information finding and seeking in software maintenance become a main challenge for software maintainers to solve the change request.

In agile method, maintenance sprints [3] are subject to interruption by urgent customer request. The maintenance team must work closely in many different systems and always dependence on customer involvement. Thus, this will cause the documentation is incomplete when both of maintainers and customer lack understand the issues and unable to fix the changes. In this study, we focus to manage documentation in agile method for software maintenance. Based on [4], typically problem occurs in documentation either poor quality of the documentation or outdated documentation or both of them.

Related works: There are many researches to add documentation for software maintenance in Agile development practice in order the after delivery process satisfy the customer and stakeholders. Gastegger [5] proposed the technical documentation, which includes all pre-release documentation and user documentation, which includes documentation for installation, operation and management of software systems. Nitin [6] proposed a documentation process in SCRUM based products. Ambler[7] proposed a documentation of the design decisions, and overall opinion of the design. However, the proposed method still unresolved the outdated documentation due to the lack of documentation management during Scrum process and improper documentation of communication between customers and developer team.

This paper is organize as follows: The following section discuss the materials and methods of research. Then the results and discussion are present in section 3. Finally, the conclusions and directions for future research are discuss in section 4.

2. MATERIALS AND METHODS

Agile software development method: Nowadays, most of the software companies are moving towards applying agile approach in their software development. The investment in agile approach is worth because the outcome of software development very impressively. The key important of agile method is the flexibility processes that is arrange based on twelve principles. Based on the agile principles, the flexibility would help software developer and project manager to design system, iterative development that help customer to gain an idea how the system look like, face-to-face communication, and better quality focus.

In general, agile methods share common characteristic[8], which comprise an iterative development process, concentrated work purposes around delivery points and many others. Maintenance sprints are focus to disruption by crucial user requests and a small number of common delivery points or integrated releases is one more challenge in agile maintenance. The maintenance team have to work closely in many different systems and always need the dependence customer involvement. Maintenance engineers have less face-to-face communication and often work closely with customer. The necessary documentation is incomplete from the development team causes a delay to understand the issue. The unhelpful use case with low communication value from customer also causing this issue not fixed and maintenance unable to complete fix on time.

Recently, software development in agile methods typically reduced the need of documentation. Agile method prefer to used unofficial communication among teams members and user, agile methods attempt to understand the necessity of the user using the unofficial method [9]. This method able to minimize the dependency for documentation in the software development.

Previously, few researcher recommended the need to define the documents of software maintainers. There are no one competent to identify the type of documentation which can aid software engineer understand the system [10]. Therefore, a number of researchers propose their idea the documentation required for software maintenance. Whereas Lethbridge [11] propose the documentation of design decisions, and a general perspective of the design such as necessity, business rules and building style. In other paper [12], the author found that the specification documents to be the most negotiated whereas quality and low-level documents less negotiated.

Proposed method: We design and developed documentation of management that would

November-December 2019
ISSN: 0193-4120 Page No. 2732 - 2742

incorporated with SCRUM methodology called Software Maintenance Management System (SMMS), which based on method proposed by Nitin [6]. We called our model as Software Maintenance Management System (SMMS). Fig. 1 shown the SMMS. The SMMS consist five elements: contract document, requirement specification document, design document, SCRUM process and lesson learnt document.

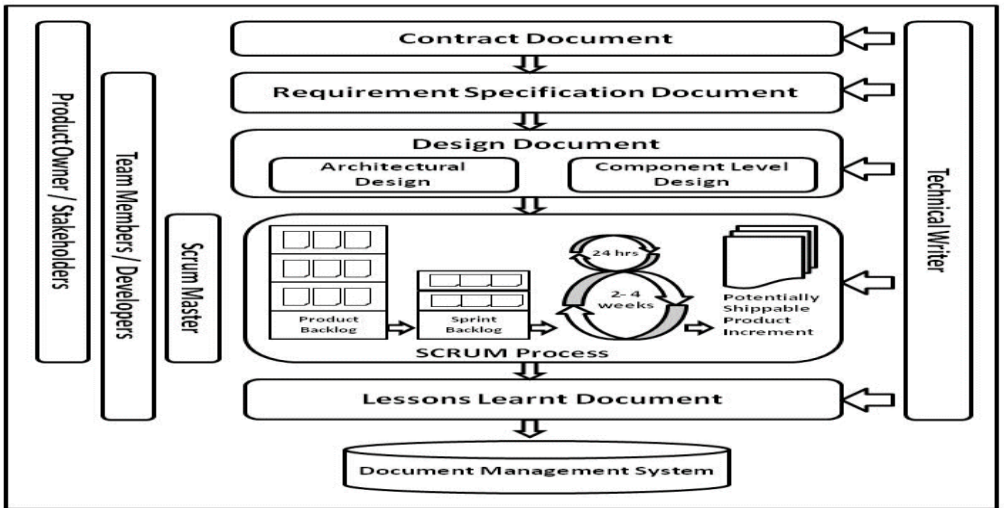

Fig. 1: Software Maintenance Management System, Nitin [6]

i. **Contract document.** The documentation stars with the initial phase. The objective of this element is to constructs a formal agreement between two or more parties. The documentation of contract document usually created by technical writer in order to maintaining the relationship between team members and stakeholder.

ii. **Requirement Specification Document**. This phase is purposely to assemble, examine and identify the requirements. The contractual agreement between customer and developer are establish based on Software Requirement Specification (SRS). Therefore, this phase will formed as common accepting and visible outlining of customer expectation.

iii. **Design document**. Software design consists the models and applies the results to a high quality system or product. The written document of a software product, offers a whole guidance the structure of software project. Therefore, this phase assists to view a picture of system architecture including its data structure, interface and components involved.

iv. **SCRUM process**. Scrum provide a framework for teams to generating and familiarizing processes. Scrum also play a role as project manager by sets up meeting and monitoring to ensure the work done based on the requirement. The product backlog usually used as a documentation and easily created from the requirement specification. This will help product owner to remember the overall system.

v. **Lessons learnt document**. Lessons learnt document is actually the documents that recorded all the challenges and constraint during development. It comprises the particulars about the development and the team. It used to construct the knowledge base of an organization and to create the history in project management and client affairs.

vi. **Technical Writer**. The most important roles of technical writer is to manage all the related documents of the project like an updating the documents and add new version of each related documents to the project. According to Mattsson[13], the quality of work product and understanding of maintenance tasks are rely on the quality of system documentation. Another researcher claims that technical writer is a professional writer to prioritize the tasks[14]. They are also capable to produce a quality documentation for inventors and clients by possessing certain skills. Thus, it is very important having a well-organized documentation as a reason to comprehend the requirement and get opinion from the stakeholders.

November-December 2019
ISSN: 0193-4120 Page No. 2732 - 2742

SMMS Design and Implementation: Basically, there are five user classes involved in this system as follows which each of the user classes have their own characteristics; system administrator, user/stakeholder, technical write, scrum or project manager and software support maintenance or agile development. Fig. 2 shows the overall processes of SMMS.

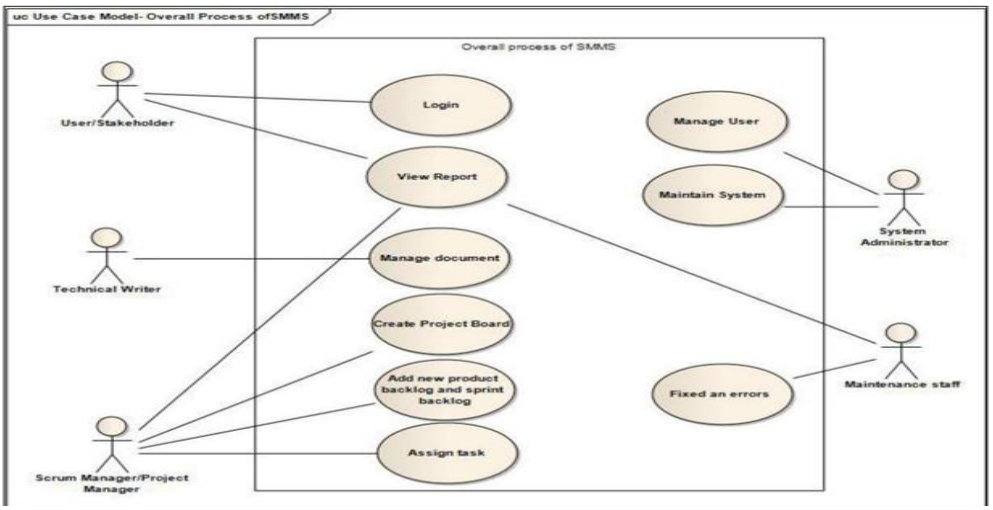

Fig. 2: The overall processes of SMMS

Based on Fig. 2, the system administrator is responsible to add user, delete user and reset password. The User/Stakeholder has main role to give the feedback/complaint if the system is not work as requested in a specification. Using the product backlog, user will add new requirements that can increase the functionality of the system. The technical writer is responsible to manage all the documents related to the project. By using the document control panel, technical writer is responsible to lock and unlock the document. Other than that, technical writer has to update the documents if there are some changes requested by users. The scrum manager or project manager job is typically involved in creating the project board, adding the new product backlog and sprint backlog, manage any documents related to the project, assign task for the support maintenance, updating the version for each document and view all the related report. Lastly, software support maintenance or agile development team software support maintenance is responsible to fix all the errors that requested by users/stakeholders.

The model of SMMS is transformed to a system of maintenance to assist technical writer cooperate with development in providing latest documentation. Fig. 3 shown the main page of SMMS.

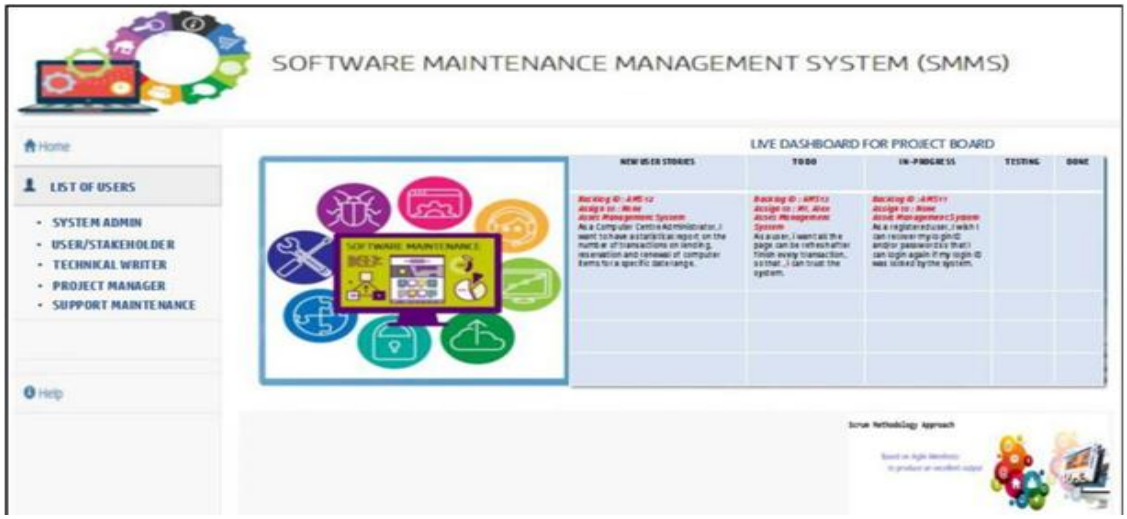

Fig. 3: The main page of SMMS

November-December 2019
ISSN: 0193-4120 Page No. 2732 - 2742

Based on Fig. 3, the dashboard will display all the latest requested product backlog and user stories. By using this facility, the user can monitor and get the latest information about the requested changes of the related application. First time user needs to sign up before able to access into the system. For existing user, they need to login to the system and only authenticated user able to access this system.

The important feature of our SMMS model is technical writer. The Document Control Button of technical writer will control all access to the document. If the documents is being updating status, this document cannot reached until the technical writer finishes updating the document. This is to avoid the overlapping of information. The new SMMS system will locked the document from reached by any users. Project Manager is typically involved in creating the project board, adding the new product backlog and sprint backlogs, upload the related document to the agile support maintenance team, assign task for the support maintenance, updating the version for each document and view all the related report.

Scrum Manager/Project Manager which is consist in creating the project board,, adding the new product backlog and sprint backlog, download the related document of the project, upload the related document to the agile support maintenance team and assign task for the agile support maintenance. Fig. 4 shows the example of project board. This menu will facilitate Scrum Manager/Project Manager to monitor the status of the project. Firstly, Scrum Manager/Project Manager have to search the PROJECT ID then this system will display all the related user stories base on the selected project.

Fig. 4: Example of the Project Board

Once the tasks has confirmed in a scrum daily meeting, Scrum Manager/Project Manager will assigned the tasks to the agile software maintenance team, which considered as appropriate. Based on the given tasks, the agile support maintenance try to solve all the requested tasks within the prescribed period. The purpose of the document is only as reference. Agile software support maintenance will write the status of the task like to do, in-progress, testing or done. If they have any issues that be notified, the support maintenance will write a comments in the comment space.

3. RESULT AND DISCUSSION

The result shows that SMMS facilitate user to manage and update the documentation in a proper method. Fig. 5shows the usefulness of SMMS. Based on Fig. 5, 80% of respondents strongly agree that the SMMS able to provide latest documentation. Whereas 70% of respondents believe that the SMMS is useful in assist them to complete all of the tasks. It shows that all of the respondents agree that documentation provided in this system is up to date. SMMS facilitate user to manage and update the documentation in a proper method. This function is useful to user and showed the propose SMMS are effective to improve the software maintenance process for agile development team.

November-December 2019
ISSN: 0193-4120 Page No. 2732 - 2742

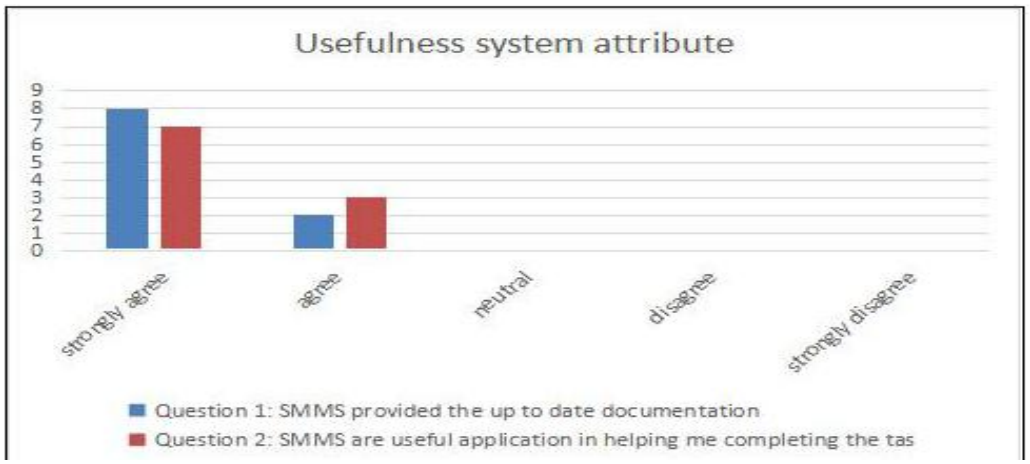

Fig. 5: The usefulness of SMMS

In terms of user-friendly attribute, almost 100% of participants agree that the proposed SMMS are user friendly and easy to use application. Most of the respondents agree that function of dashboard in SMMS help users to manage and monitor the progress. Only one respondent chose the neutral scale. Completeness attribute is regarding the sufficient facilities and functions in SMMS that assist agile development team/maintenance team in conducting the maintenance task. There are three questions involved to evaluate on completeness attribute. Fig. 6 shows that most of participants agree that SMMS provides an enough functions to facilitate the maintenance task. SMMS provide requirement regarding to understanding the tasks by using the latest documentation. As a conclusion from the survey, SMMS performs the completeness attributes by providing the function and capabilities that expected by the users.

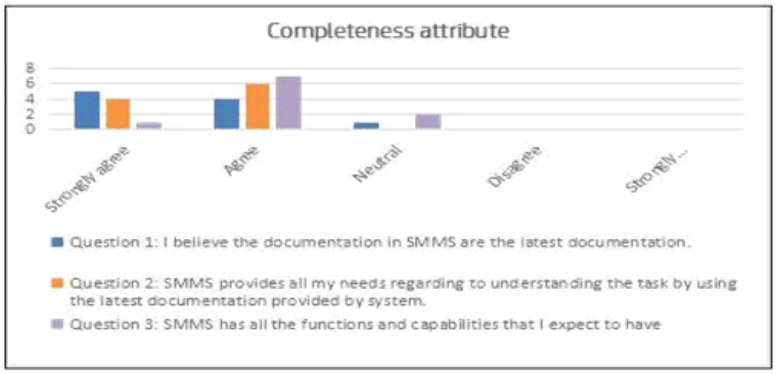

Fig 6: The completeness of SMMS

From the result, it proven that by implementation the up to date documentation, it will help the agile development team/maintenance team to increase the significance of the work. An efficient documentation is a major factor to understand the requirement and. The documentation also plays essential role in communicating between developers, managers and customers. From the perspective of documentation, the proposed SMMS is better than the existing agile software maintenance application.

4. CONCLUSION

This study has showed agile model has improved by introduced technical writer. We have proved our concept by developing SMMS prototype which focusing on managing the documentation in conducting the maintenance task for the agile development team. By introduced the new roles of technical writer in agile software development the software maintenance process can be more effective. This paper also presented the relationship of documentation and technical writer where can enhance the productivity and maintainability of software. The additional roles of technical writer would increase effectiveness in conducting the software maintenance process.

ACKNOWLEDGMENT

This research sponsored by Ministry of

Education Malaysia (MOE) and Universiti Putra Malaysia (UPM) via the Fundamental Research Grant Scheme (FRGS, Vot No: 5524956).

REFERENCES

[1] Cohn-Muroy, D., & Pow-Sang, J. A. *Trends and Applications in Software Engineering*. Advances in Intelligent Systems and Computing.2016;405:15-24.

[2] Zahid, A.M and Samia, F. *The Benefits and Key Challenges of Agile Project Manage ment under Recent Research Opportunities*, International Research Journal of Management Sciences. 2017;5(1):20-28.

[3] Muhammad O A, Kuvaha P, Oivo M, Markkula J. *Transition of Software Maintenance Teams from Scrum to Kanban*, 49th Hawaii International Conference on System Sciences(HICSS), Hawaii. 2016;5427-5436

[4] Sergio Cozzetti B. de Souza, Nicolas Anquetil, and Káthia M. de Oliveira. *A study of the documentation essential to software maintenance*. In Proceedings of the 23rd Annual International Conference on Design of Communication: Documenting &Amp;Designing for Pervasive Information, SIGDOC '05. 2005:68–75.

[5] Gastegger, Mario &Zünd, Simon. Maintenance of technical and user documentation.2015. Available:https://www.researchgate.net/publication/303883087_Maintenance_of_technical_and_user_documentation [July 2, 2016]

[6] Nitin Uikey ,UgrasenSuman,A.K. Ramani. A Documented Approach in Agile Software Developme, International Journal of Software Engineering (IJSE), 2011; 2 (2) : 13-22.

[7] Ambler, S. W.*Agile/Lean Documentation: Strategies for Agile Software Development*. 2010. Available:http://www.agilemodeling.com/essays/ agileDocumentation.htm [Dec. 1, 2010]

[8] Prochazka, J. *Agile Support and Maintenance of IT Services*. In: Information Systems Development, Prague, Czech Republic. Springer, 2011; 597–609.

[9] D.Knippers. *Agile Software Sevelopment and Maintainability*, 15th Twente Student Conf.,2011,theNetherlands,©University of Twente.

[10] Kapitsaki, G. M., & Christou, M. (2014). *Where is Scrum in the current Agile world?* Proceedings of the 9th International Conference on Evaluation of Novel Approaches to Software Engineering, 2014: 101–108.

[11] T.C. Lethbridge et al. *How Software Engineers Use Documentation: The State of the Practice*. IEEE Software, IEEE Computer Society, 2003; 20(6):35-39.

[12] Raj, G., Yadav, K., & Jaiswal, A. *Emphasis on testing assimilation using cloud computing for improvised agile SCRUM framework*. [1]st International Conference on Futuristic Trends in Computational Analysis and Knowledge Management, ABLAZE 2015: 219– 225.

[13] Mattsson, M.K., Azizyan, G., &Magarian, M.K. Classes of Distributed Agile Development Problems. Agile Conference (AGILE). 2010;51-58.

[14] Baptista. J. *Agile Documentation with uScrum*. SIGDOC '08, Proceedings of the 26th annual ACM international conference on Design of communication, ACM, New York, NY, USA, 2008:275-276.