

Towards Adopting Software Quality Assurance in Agile Development Methodology

Rosnani Shuib¹, Sa'adah Hassan^{*2}

^{1,2}Faculty of Computer Science and Information Technology, Universiti Putra Malaysia
gs52387@student.upm.edu.my¹,saadah@upm.edu.my^{*2}

Article History: Received: 10 November 2020; Revised: 12 January 2021; Accepted: 27 January 2021;
Published online: 05 April 2021

Abstract: In agile software development methodology, software product is developed through a few iterations and delivered incrementally. In spite of the known benefits of the agile methodology, yet issue still exists in developing high-quality software product using this methodology. Hence, quality assurance mechanism should be in place in agile development to ensure quality adherence to the software product. However, taking up quality assurance in agile software development is not an easy task as it has its own manifesto and principles that need to be met. Thus, this paper highlights the issues and challenges towards adopting quality assurance in agile development methodology. Particularly, in an iteration where we need to assure that the product increment meets the requirements as well as to decide the requirements for next increment and iteration planning. Subsequently, few measures are identified concerning the issues. The discussion in this paper is based on Scrum case, yet still relevant to other types of agile methodologies.

Keywords: agile development, quality assurance, review, Scrum

1. Introduction

Agile software development methodology focuses on frequent delivery through light and fast development cycle while adapting customers' requirements changes. Nevertheless, quality assurance mechanism must be established in agile development methodology in order to ensure high-quality software product is produced. Quality assurance involves tasks to evaluate software products for conformance against the established software requirements (ISO/IEC/IEEE, 2011). In agile development methodology, the software product is developed through a few iterations and delivered incrementally. Significantly, the quality assurance mechanism is needed to assure that the developed product increment meets the requirements, as well as to inform the requirements for the next increment and iteration planning that might also evolve. This will also ensure that the quality is encountered earlier before the final product delivered.

It is important to adopt strategy in order to address quality issues in agile software development at the early stage. However, quality assurance in agile software development is not an easy task since the working software must be releasable and yet still open for frequent requirements changes. Therefore, this paper discusses issues as well as measures that need to be considered in adopting software quality assurance in agile, particularly Scrum methodology. Our work focuses in adopting quality assurance during the review process of product increment.

The remainder of this paper is organized as follows: Section 2 discusses the related work, section 3 discusses issues and challenges in adopting quality assurance mechanism in Scrum, and section 4 presents few suggestions to be considered to resolve the issues. Finally, section 5 concludes this paper and our future work.

2. Related Work

Figure 1 shows some examples of known quality assurance approaches being adopted in software development activities. Traditionally, quality assurance activities in software development life cycle usually being focused at the later phases; implementation and testing phases. However, adopting quality assurance activities at the later phases lead to increase of cost and rework. Realizing that, many efforts placed to adopt quality assurance activities at the earlier stages (i.e., requirements engineering). For example, writing requirements specification by referring to the IEEE standard for requirements specification (IEEE, 1998), and using users' feedback to validate the measurements of quality and user satisfaction (ISO/IEC/IEEE, 2011). One of the common and effective approaches has been adopted at requirements engineering stage is inspection. In which, it prevent issues from being propagated from requirements to other artefacts that will cause follow-up defects and avoidable rework (Briand, Freimut, and Vollei, 2000).

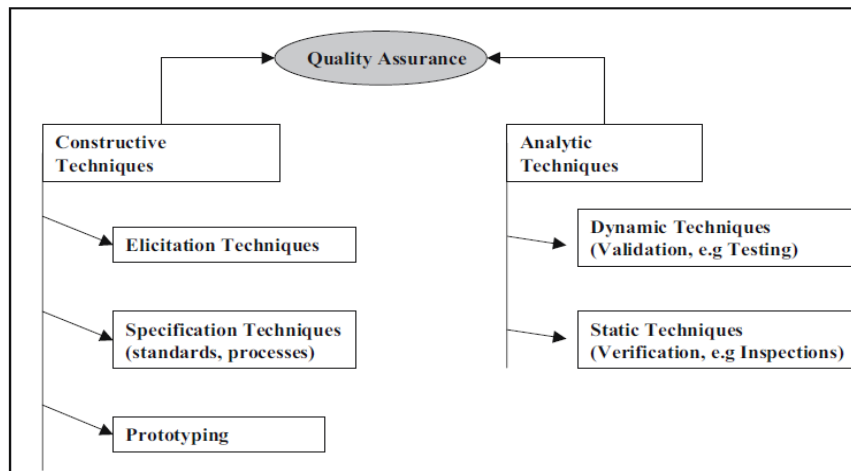


Figure 1. Excerpt of quality assurance approaches (Denger and Olsson, 2005)

It is easier to adopt quality assurance in traditional software development methodology as it focuses on comprehensive documentations throughout the development life cycle. However, the nature of agile development methodologies is differing from traditional software development methodology. In which, it focuses more on coding rather than documentations. Besides, it acknowledges that customer requirements are always changing and the changes must be adapted quickly, while enables short time frames and iteration releases (Bruegge, Krusche, and Wagner, 2012). Therefore, many software developers adopt agile methodologies because it support fast delivery and adapt to changes. There are a few styles of agile methodologies introduced, such as Scrum, XP Programming, Crystal Method, Pair Programming, and Feature Driven Development (FDD). Despite of its popularity, a number of recent studies address issues on the quality (Cho, 2010; Medeiros, *et al.*, 2018; Alsaqaf, Daneva, and Wieringa, 2019) and customers' involvement in agile methodology (ISO/IEC/IEEE, 2011; Curcio, *et al.*, 2018; Hussain *et al.*, 2015; Medeiros, *et al.*, 2018). Besides, the quality of the software product would also be affected due to the short increments and limit period for stabilization (Krusche, *et al.*, 2014), and the requirements volatility (Kawamoto and De Almeida, 2017). Particularly when the changes occur at 30% in every single iteration of agile development (Amjad, *et al.*, 2017).

There are some efforts to adhere quality assurance in agile. For example, Amjad and colleagues (Amjad, *et al.*, 2017) proposed a method, called A-SPSRI, that able to measure the quality and completeness of scope definition of the user stories. While, there are also tools and techniques to ensure effective collaboration and the successful development of the product (Hess, Diebold, and Seyff, 2019). For example, task assignment and task tracking are done by using online management tools, online versioning tools, online file sharing and online meetings (Curcio, *et al.*, 2018). Another effort is by Kawamoto and De Almeida (2017), in which they proposed an agile software development model that adheres to capability maturity model integration (CMMI) quality practices named as DOC-DR as additional practices to current Scrum process. Besides, they also highlight the importance of the documentation and evidence needed for the formal process improvement model. Regardless all the efforts, quality related issues in agile software development methodology are still exists, particularly when dealing with customers' involvement. For example, findings from review by Inayat *et al.* (2015) highlighted obstacles that make the requirements engineering process more challenging in a distributed agile project, including customers' involvement difficulties, communication difficulties, and, lack of documentation. These are also included in recent study by Shameem, M., *et al.* (2020) that highlights the key obstacles faced by agile practices in a global software development. Among the identified obstacles mentioned are poor requirements and their specifications, inappropriate selection of communication technology, lack of customer involvement, limited project documentations, and lack of communications. Those obstacles make the implementation of quality assurance activities become harder. In agile, the customer involvement will make the development activities more effective (e.g., writing user requirements, discussing product features, prioritizing the client requirements) (Hastie and Wojewoda, 2015; Hussain *et al.*, 2016; Kasauli *et al.*, 2017). For example work by Wil, *et al.*, (2019) that involving customers in backlog prioritization.

The common quality assurance strategies in agile are prototyping and review. In which, the strong customers' involvement in this process will be leveraged to assess conformance of the product or increment. This is where the software product can be reviewed by the customers at the early stage for acceptance or improvement (De Sousa, *et al.*, 2016). It is important that the customers indicate that all of the accepted deliveries constitute acceptable completion of the project. Whereby, the manager, according to IEEE Software Quality Assurance Standard (IEEE, 2014), should establish a process to access, collect, and monitor available sources of qualitative

feedback about the information for evaluation and acceptability of the product (i.e., conformance to the established requirements). Therefore, we focus on review process as a main channel to adopt quality assurance in agile methodology, in which, high involvement of customers at the early stage, such as in review, will improve the software quality significantly (Sirsharandarif, 2012).

3. Issues And Challenges: Scrum Case

Agile methodology focusing more on working software product (i.e., coding) and less concentrate on documentation. Therefore, the customers' involvements are extremely important to determine the achievement, acceptance, and direction of product development. Whereby, their feedback becomes valuable input for planning and making necessary adjustment in order to deliver a product that meets the business needs. Scrum was chosen as an exemplar to discuss the issues and challenges of agile methodology for this study. Scrum is one of prominent agile methodology that has a shorter time cycle for the fast delivery of the product to the customers (Moniruzzaman and Hossain, 2013). In addition, Scrum highlights the importance of quality assurance through feedback mechanisms for managing a complex project (De Sousa, *et. al*, 2016). Figure 2 shows Scrum methodology, wherein the process works in an iterative and incremental manner and the customers and development team are working together closely. Customers give their feedbacks mostly through review meetings.

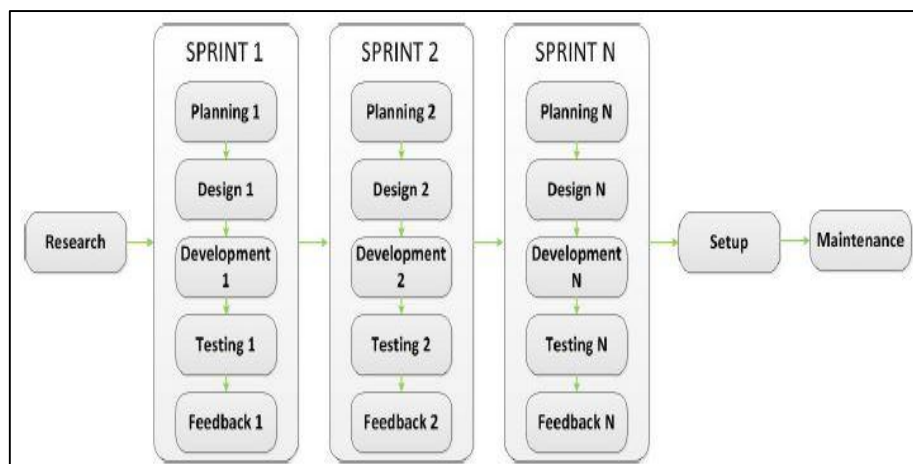


Figure 2.Scrum Software Development Methodology (Moniruzzaman and Hossain, 2013)

In Scrum methodology, user requirements is called as product backlog that will be developed into few iterations called *sprint*. The product backlog is continuously reviewed and prioritized in each iteration during iteration planning (Tomanek and Juricek, 2012). Usually, the product backlog will constantly changing (i.e., feature is removed, refined, inserted), thus the backlog has to re-prioritize and decide on which should be developed first. The iteration review in Scrum looks into the perspective of functionality delivered in the product increment whether the features (or functionalities) meet the acceptance criteria. Validation and verification of the increment is based on the feedback gathered in the iteration review. In which, customers can participate to ensure the quality of the product increment satisfies the business goals as well as bring up any requirements changes or new request. A review process will be carried out at the end of iteration when a product increment is ready for released. Customers review the product increment, give comment and determine the acceptance level of a product increment whether to accept, reject or request for a change. Commonly, Product Owner as a customer's representative that involved in the iteration and who managed all the feedback gathered from customers. The process of gathering feedback should be in an effective and timely manner within iterations and avoid any feedback delays. In addition, changing of requirements at every single iteration is another challenge for Product Owner and the team. There are 4 round level of decision making need to be made by Product Owner at every single iteration, which are - iteration planning, iteration execution (refinement, development and testing), iteration review and iteration retrospective. Whereas, poor decisions in this process can results in code backlogs, incomplete work, bugs, and inaccurate estimation (Dury-Grogan, Conboy, and Acton, 2017).

There are issues mentioned in the previous studies in relation to customers involvement in agile process (e.g., Shrivastava and Rathod, 2017; Medeiros *et al.*, 2018; Curcio *et al.*, 2018; Patanakul and Rufo-McCarron, 2018; Hess, Diebold, and Seyff, 2019). Those are also barriersto adopt quality assurance mechanism in Scrum, particularly for review process. The issues and challenges can be summarised as:

- Late validation – the availability of customers for validation and verification of product increment and requirements for next iteration are crucial. Thus, the unavailability of customers in the review process will cause late validation since multiple iterations pending for customers' feedback.
- Ineffective way of getting feedback – feedback from customers should be gathered within stipulated time; yet, gathering and managing the feedbacks are tedious. Thus, if the communication medium used to collect and the process of gathering the individual feedback are ineffective, it will cause the increased overhead and response time for overall validation process.

While, the consequences of the mentioned issues are,

- Excessive changes and improvement needed on product increment due to requirements or scope changes
- Too many adjustments and development work required for next iteration. This including, to prioritize requirements and to identify requirements changes and its dependency
- Late delivery of product increment and poor customer's satisfaction.

Thus, a proper strategy needed in the review process to confirm the quality of the product increment with the specified acceptance criteria, and to identify any changes or corrective action needed as well as inform decisions for future planning in a timely manner.

4. Discussions

As iteration review is the significant quality assurance mechanisms in agile development process. Therefore, coordination and communication among the team and customers are crucial in the iteration review session. Inadequate and ineffective of coordination and communication between the Product Owner (i.e., customer's representative) and customers can slowdown the process of overall development progress. This section lists the main measures need to be considered in adopting quality assurance in the review process, particularly for Scrum.

1) Customers Availability – customers' feedbacks are important. Thus, need to ensure that customers are available and able to give feedback within the specified time. The Product Owner must be able to gather feedbacks collectively from all the customers to determine successful implementation of a product increment. Besides, not forgotten for the customers that are in geographically far sites. This is common scenario in most organizations where customers can give a greater flexibility for remote validations and also reduce stakeholder's displacement costs (Medeiros, *et al*, 2018). Thus, the availability of customers in different geographical locations should also need to be considered in order to ensure their participation in the review session. There are many obstacles might arise, but a proper approach is needed to ensure the customers availability.

2) Communication Medium – the challenge related to distance barrier has caused inadequate coordination between the team especially Product Owner and distributed customers for frequent meeting. Current medium of communication used in collaboration between Product Owner and distributed customers such as emails and telephone are not an effective way for getting feedback. While, medium such as video conferencing or skype cannot help in capturing details about the feedback attributes and hard to find suitable time to be in the same time for the online session. It can caused difficulties in compilation of all the feedback attributes and also difficult to segregate the unstructured feedback information for further iteration action planning. It is mentioned that the current approach for remote validations did not affect the project execution if the customers had an adequate collaboration (Medeiros, *et al*, 2018). While, a too relaxed collaboration with the customer entails a bottleneck in the project, and said that this challenge can be better dealt with face-to-face communication (Medeiros, *et al*, 2018). In view of those issues, an appropriate framework of communication medium for collaboration between Product Owner and distributed customers is needed to capture and compile the feedback attributes during the review session.

3) Supporting tools - lack of tools to support customers to carry out quality assurance activities, such as validating the requirements in each iteration review, will make the process difficult. Ineffective documentation also bring difficulties, such as control of backlog using spreadsheets, thus it sees the relevance of adopting appropriate tools (De Sousa, *et. al*, 2016).

Above all, Figure 3 illustrates the quality assurance issues and its measures in Scrum's review session and the affects in the iteration planning.

In summary, quality strategies are needed to define how, when and where to adopt quality assurance approaches in order to assure achieving desired quality of the software product. By indicating the measures, expectantly it will help in designing and embracing quality assurance strategies in agile methodology.

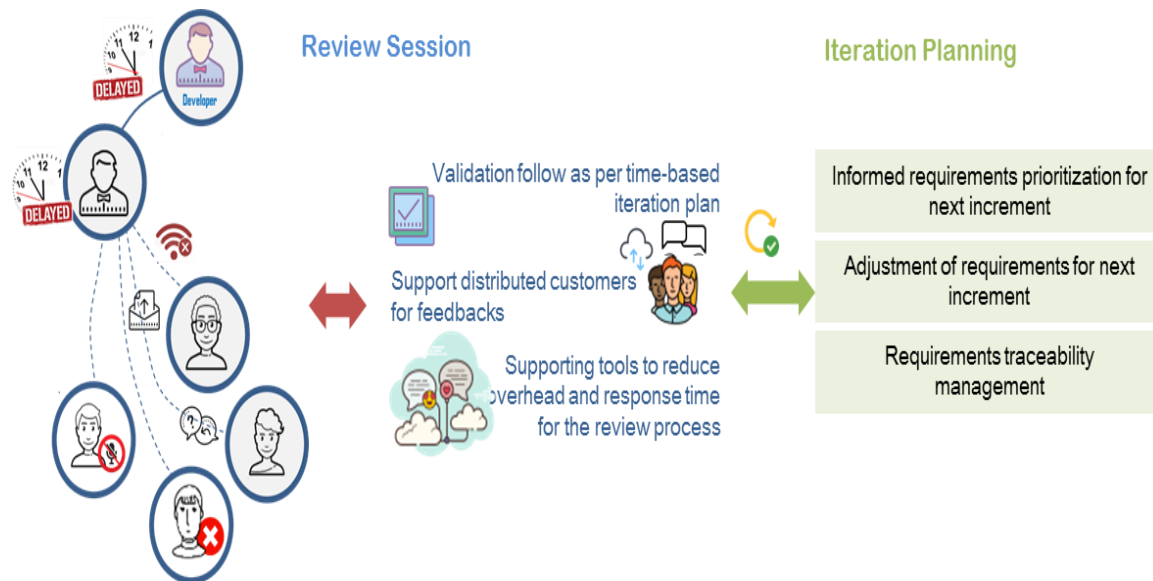


Figure 3. Quality assurance measures in Scrum's review session

5. Conclusion

In agile software development methodology, feedbacks from customers are crucial in order to ensure the quality of the product delivered meet the business goal. However, there are some obstacles in adopting quality assurance mechanism based on the nature of agile methodology itself. Thus, the main obstacles for adopting quality assurance are highlighted in this paper. Subsequently, few measures are identified concerning the issues discussed. For this study, Scrum methodology was taken as an exemplar. Yet, the findings are still relevant to all agile software development methodologies. Looking at the state of the art, there are room for improvement needed to adopt quality assurance activities, particularly in the review process. The research findings are very useful towards the development of our proposed approach and currently, we are working on a framework and tool that take into account the findings.

6. Acknowledgment

We would like to express appreciation for the support of the Universiti Putra Malaysia.

References

1. Alsaqaf, W. Daneva, M. & Wieringa, R., 2018. Quality requirements challenges in the context of large-scale distributed agile: An empirical study, *Inf. Softw. Technol.*, vol. 110, no. July 2018, pp.39–55.
2. Amjad, S. *et al.*, 2017. Calculating Completeness of Agile Scope in Scaled Agile Development, *IEEE Access*, vol. 6, pp.5822–5847.
3. Briand, L.C, Freimut, B., Vollei, F. 2000. Assessing the cost-effectiveness of inspections by combining project data and expert opinion. In: *Proc. of the 11th Inter.Symp.on Soft. Reliability Eng.*, pp.124-135.
4. Bruegge, B. Krusche, S. & Wagner, M., 2012. Teaching Tornado Categories and Subject Descriptors, *Edu. Symp12*, pp.5–12.
5. Cho, Juyun Joey, 2010. An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum, All Graduate Theses and Dissertations. 599.
6. Christian Denger & Thomas Olsson, 2005. Quality Assurance in Requirements Engineering, In: Aurum A., Wohlin C. (eds) *Engineering and Managing Software Requirements*, pp. 163-184.
7. Curcio, K., Navarro, T., Malucelli, A. & Reinehr, S. 2018. Requirements engineering: A systematic mapping study in agile software development, *J. Syst. Softw.*, vol. 139, pp.32–50.
8. De Sousa, T. L. Venson, E. Figueiredo, R. M. D. C. Kosloski, R. A. & Ribeiro, L. C. M., 2016. Using scrum in outsourced government projects: An action research, *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 2016-March, pp. 5447–5456.
9. Drury-Grogan, M. L., Conboy, K., & Acton, T., 2017. Examining decision characteristics & challenges for agile software development. *Journal of Systems and Software*, 131, pp.248–265.
10. Hastie, S. & Wojewoda, S. 2015. Standish group 2015 chaos report - q&a with Jennifer Lynch, [Online]. Available: <https://www.infoq.com/articles/standish-chaos-2015>.

11. Hess, A. Diebold, P. & Seyff, N., 2017. Understanding information needs of agile teams to improve requirements communication, *J. Ind. Inf. Integr.*, vol. 14, pp.3–15.
12. Hussain, A., Nuruzzaman, M., Tahir, H.M. (2015). Industrial web application customization mechanism to improve software quality and productivity. *International Journal of Mathematical Models and Methods in Applied Sciences*, 9, pp. 396-403.
13. Hussain, A., Mkpojiogu, E.O.C., Kamal, F.M. (2016). Antecedents to user adoption of interactive mobile maps. *Journal of Telecommunication, Electronic and Computer Engineering*, 8 (10), pp. 41-45.
14. IEEE, 2014. *IEEE Standard for Software Quality Assurance Processes*, vol. 2014.
15. IEEE. 1998. IEEE Recommended practice for software requirements specification. *IEEE Std 830-1998*.
16. Inayat, S.S. Salim, S. Marczak, M. Daneva, S. Shamshirband, 2015. A systematic literature review on agile requirements engineering practices and challenges, *Comput.Hum. Behav.* 51 (2015), pp. 915–929.
17. ISO/IEC/IEEE, 2011. International Std.ISO / IEC / IEEE Systems and software engineering — agile environment,” *ISO/IEC/IEEE 26515 First Ed. 2011-12-01; Corrected ver. 2012-03-15*, vol. 2018, pp. 94.
18. Kasauli, R. Liebel, G. Knauss, E. Gopakumar, S. Kanagwa, B., 2017. Requirements engineering challenges in large-scale agile system development, in: *Req. Eng. Conf.(RE), IEEE 25th Int.*, IEEE, pp.352–361.
19. Kawamoto, S. & De Almeida, J. R., 2017. Scrum-DR: An extension of the scrum framework adherent to the capability maturity model using design rationale techniques, *2017 Chil. Conf. Electr. Electron. Eng. Inf. Commun. Technol. CHILECON 2017 - Proc.*, vol. 2017-Janua, pp.1–7.
20. Krusche, S., Alperowitz, L., Bruegge, B. & Wagner, M. O., 2014. Rugby: An agile process model based on continuous delivery, *1st Int. Work. Rapid Contin. Softw. Eng. RCoSE 2014 - Proc.*, pp.42–50.
21. Medeiros, J., Vasconcelos, A., Silva, C. & Goulão, M., 2017. Quality of software requirements specification in agile projects: A cross-case analysis of six companies, *J. Syst. Softw.*, vol. 142, pp.171–194.
22. Moniruzzaman, A. B. M. & Hossain, D. S. A., 2013. Comparative study on agile software development methodologies, *Global Journal of Computer Science and Technology*, vol. XIII (VII), pp.5–18.
23. Tomanek , M. & Juricek, J., 2012. Addressing main problems in project management (PRINCE2) by adopting agile principles (Scrum), in *Proc. IADIS*, vol. 5., pp.385-389.
24. Patanakul, P., & Rufo-McCarron, R., 2018. Transitioning to agile software development: Lessons learned from a government-contracted program. *J. of High Tech. Management Research*, 29(2), pp.181–192.
25. Shameem, M., Ranjan, R., Nadeem, M., & Ali, A., 2020. Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process. *Applied Soft Computing Journal*, 90, 106122.
26. Shrivastava, S. V., & Rathod, U., 2017. A risk management framework for distributed agile projects. *Information and Software Technology*, 85, pp.1–15.
27. Wil, C. S. C., Ban, A., Hassan, S., Pa, N. C., & Din, J. (2019). VoVo: A hybrid requirements prioritization technique in scrum-agile environment. *International Journal of Adv. Sc. and Tech.*, 28(2), pp.363 - 369.