COLLABORATIVE FEEDBACK APPROACH FOR IMPROVING CUSTOMER ENGAGEMENT AND PRODUCT INCREMENT IN AGILE SOFTWARE DEVELOPMENT

Rosnani Shuib and Sa'adah Hassan

Abstract — Customer engagement and feedback are crucial in agile software development to ensure the product released meets their needs. However, lack of involvement and delayed feedback are the main challenges, especially if the customers are in distributed locations. These could cause the product increment quality, poor iteration planning decisions, and late product delivery. A collaborative feedback approach is proposed in this paper to support gathering feedbacks of product increment and decision-making in iteration development planning. The proposed approach also embraces a quality assurance (QA) mechanism to ensure quality adheres to the product increment. An evaluation was conducted where a tool, called CrowFeID, was developed as a proof-of-concept of the proposed approach and to support the iteration review process. As a conclusion from the responses received, the proposed approach and tool can reduce delayed feedback during the iteration review and provides valuable input in making decisions for the product increment and next iteration planning.

Keywords — Agile development, collaborative feedback, iteration review, quality assurance, Scrum.

I. INTRODUCTION

AGILE software development focuses on collaboration and continuous improvement for delivering a good quality product in a shorter time. Wherein, customer involvement is crucial to determine the achievement, acceptance, and direction of the software product under development. Their feedbacks are a valuable input to track the project status, identify changes on the requirements, and rearrange priority for the next release. Wherein at the initial planning, the software requirements are only at high-level details and then will be refined based on customer' feedbacks in each iteration. Thus, by not engaging the customers in the process, the project is likely to fail.

Sa'adah Hassan is with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM Serdang, MALAYSIA (corresponding author, phone: 603-97691761; fax: 603-97696576; e-mail: saadah@upm.edu.my).
Rosnani Shuib, was with Universiti Putra Malaysia. She is now with the Implementation Coordination Unit, Prime Minister Department, Putrajaya, MALAYSIA (e-mail: rosnani@icu.gov.my).

The success of an agile project depends on the customers' requirements delivered in the product [1]. Quality assurance (QA) in software development is a planned strategy in assuring that the product under development meets the specified requirements. QA can be checklists, standards, project audits, methodology, or development procedures. Review sessions in agile software development methodology considered as a QA mechanism [2]. The review process will be carried out at each iteration to know the readiness of the product increment for release. Customers will review the product increment and provide their feedback regarding its acceptance level (i.e., whether to accept, reject or request a change) and identify any changes or corrective action needed (if any) to improve the product increment. That is how to assure the quality of the product increment. Generally, the review process supports verification and validation of the product increment. However, delayed feedback resulted in late validation in which can cause excessive changes [3] thus, affect overall development planning. The main reasons for delayed feedback are the distance barrier of distributed customers and the lack of a specialized communication tool to support the review [4]. Moreover, agile projects can fail due to improper iteration and daily planning meetings [1]. It is important to manage the requirements and deliver a product increment that meets the acceptance criteria at the point of its release. Several measures to address the issues have also been highlighted [17].

Thus, it has motivated this research is to study how to avoid delayed customers' feedback by engaging customers during the review sessions in agile software development. Also, aim to assist decision-making in iteration planning to ensure quality adheres to the product increment. This work focuses on feedback coordination and communication among customers in Scrum methodology. Scrum is an agile methodology that highlights the importance of quality assurance through feedback mechanisms, crucially for a complex project [5]. However, our proposed approach is also suitable for other types of agile methodologies. The proposed collaborative feedback approach adapts the current feedback process of

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

iteration review with additional steps to adapt the QA mechanism. In addition, an online tool has been developed as a proof of concept of the proposed approach.

The remainder of this paper is organized as follows; Section 2 presents the related work, followed by the presentation of the proposed approach in Section 3. The evaluation and results are discussed in Section 4. While, Section 5 raises the conclusion, which also includes the limitations and recommendations for future research.

II. RELATED WORK

The agile methodology process is performed iteratively and incrementally where the customers and the development team work closely together. It aims to continuously deliver workable product increment at the end of each iteration.

Scrum is one of the prominent agile methodologies that have a shorter time cycle for the fast delivery of the product to the customers [6]. Fig. 1 is a graphic that represents Scrum as described by Ken Schwaber and Jeff Sutherland [7]. The *product backlog* consists of a list of requirements (i.e., *user stories*). Product backlog will be developed into product increments through several iterations. The iteration is called *sprint* in Scrum. The product backlog is continuously reviewed and prioritized in each iteration during iteration planning[8].The *sprint backlog* is the mechanism used to add 'ad-hoc' features into the development pipeline, which then to be re-prioritized by the *Product Owner* during the Scrum meeting (i.e., sprint planning) [1]. In sprint planning, the Product Owner needs to decide on which requirements should be apriority to develop first. While the scrum team is the development team responsible for completing the product increments development tasks. Customers provide their feedbacks on product increment through *sprint review* sessions. In large or distributed organizations, the Product Owner must be able to gather feedback collectively from all the distributed customers to ensure the successful implementation of product increments.

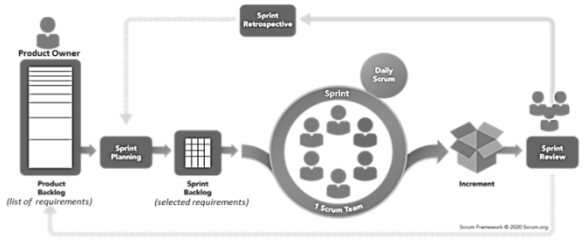


Fig. 1 Scrum in action (adopted from [7])

Typically, the Product Owner is a customer representative involved in the iteration and managed all the customers' feedbacks. The Product Owner should have been informed regarding the project status and the customers' acceptance before making any decisions on the product increment and next planning. During the iteration planning, the Product Owner must make the most crucial decision on the iteration scope, acceptance criteria, and schedule for requirements or user stories for the next iteration. Those are considered strategic decisions to determine what the organization can deliver to customer [9]. However, sometimes the decisions made are without sufficient information and based on uncertainty. These lead to poor decisions and results in code backlogs, incomplete work, bugs, and inaccurate estimates [9].

Moreover, the nature of agile development that has short iterations has abandoned the importance of quality metrics, experience, or decision consequences [9]. Agile team should therefore use appropriate decision strategies to achieve quality and ensure project success. However, the choice to use a particular strategy depends on many factors, including the availability and accuracy of information [9]. The importance of customer involvement in agile development is also crucial as it determines the options that can be selected by the development team for subsequent iteration development [9]. Meanwhile, Shameem, M., *et al*. [4] have mentioned that one of the obstacles faced by agile practices is limited project documentation. Where, Kawamoto and De Almeida [10] also highlight the importance of the documentation and evidence needed for the formal process improvement model as in their proposed agile software development model.

For distributed agile system development, it can be as effective as co-location when mixed of synchronous and asynchronous methods are used appropriately in every phase [11]. It is also mentioned that richer communication (e.g., face-to-face) might suitable and suffice during the earlier phase of agile development and can be reduced to a lower synchronous method (e.g., online web conference) or less rich communication medium (e.g., emails) depending on the necessity to what extend it can support enough communication in each phase. It has been said that quality requires a combination of richer and less rich communication mediums to ensure that the correct messages are sent and received, thus resulting in higher quality and fewer defects in software products [11].

There are several efforts on collaborative feedback approaches and tools. For example, Krusche and Alperowitz [12] implemented a few steps in the deployment process by using integrated tools to release

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

product increments by time-based or event-based to get a shorter feedback cycle. The proposed tool has three components, version control (VC), continuous integration (CI), continuous delivery (CD), and the customer accessing via web from the CD server. The integrated tools represent the delivery workflow steps cover from the version control server to the delivery server and then to the customer for feedback during development. The product increment can be accessible through the delivery server, called Web Application under Test. The feedback can get in shorter cycles for the benefits of development in which enables the team to track the current build state of the software, configuration errors, and handle releases. In addition, Krusche *et al*., [13] proposed an agile process model called Rugby with workflows for continuous delivery and feedback management that aims to improve the coordination across multiple teams as well as the communication between developers and customers. Rugby improves the use of executable prototypes as communication models can reduce the status reports and discussion and helps in the requirements elicitation process [14]. While Oriol *et al*., [15] proposed FAME, a framework wherein feedback management is the main component that manages and configures the feedback dialogue and finally sends the collected feedback to the data storage. FAME provides online feedback mechanisms that enable end-user to remotely communicate problems, experiences, opinions while monitoring valuable information about runtime events. It aims to support continuous requirements elicitation by combining user feedback and usage monitoring [15]. Table 1 provides a summary of the existing efforts on collaborative feedback.

Table 1. Summary of Related Work On Collaborative Feedback in Agile.

Paper	Description	Proposed Work
Introduction of continuous delivery in multi-customer project courses [12].	Continuous delivery concept. Gather customer feedback for every releasable product increment by time-based or event-based handled by release manager/coordinator.	A few platforms integrated (VC, CI, CD) and customer access via web from CD server.
Rugby: An Agile process model based on continuous delivery [13].	Rugby process uses separate tool but integrated between development and collaboration Gather feedback by event-based which release increment as development prototype for elicitation.	Rugby process model.
FAME: Supporting continuous requirements elicitation by combining user feedback and monitoring [15].	FAME is a feedback management component in the framework used to gather user feedback and handles runtime monitoring data as valuable information in continuous elicitation	FAME, an online (web and mobile) feedback mechanism

High involvement of users in activities and techniques such as review, inspection, and testing will significantly improve the software quality [2]. Besides, their involvement will make the development activities more effective, for example involving customers in requirements prioritization [18]. However, it is difficult if their lack tool to assist them to carry out the activities and to encourage their involvement in the review process. Some using spreadsheets, yet hard to control the backlog, and thus it sees the relevance of adopting tools [5]. Getting feedback from the distributed customers by facilitating the team with appropriate communication tools is recommended to reduce risk because of the distance barrier [16]. Moreover, the tools will help to improve the requirements traceability and facilitate the impact analysis of requirements changes, especially in complex projects or projects with large teams [3]. In distributed agile development, it is essential how the Product Owner manages and gathers feedback from the distributed customers in an effective and timely manner and avoids any feedback delays.

Though many existing standards and models implement agile in a co-located environment but still failed to scale agile practices in large companies [4]. Some of the barriers found from the study are the lack of customers' involvement, lack of communications, inappropriate selection of communication technologies, lack of organizational commitments, lack of roles and responsibilities, and lack of scaling tools and standards [4]. In agile development, the customer needs to participate actively in requirements gathering, requirements prioritization, and feedback processes among the managers and team [4].

It is crucial to ensure customers' engagement during the review and how the communication medium can provide an effective feedback process and supply enough information to decide on sprint backlog as well as assist in decision-making for iteration planning.

III. THE PROPOSED APPROACH

A collaborative feedback approach in agile software development, called CrowFeID, is proposed. It aims to acquire feedback on the product increment from customers during the iteration review. Typically, the customers involved in agile development are in distributed locations, and getting feedback from them will be time-consuming. Thus, CrowFeID utilizes an online

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

platform to support the distributed customers. For this study, the proposed approach focuses on Scrum agile methodology. However, it is also suitable to support the review process of other agile development methodologies, like extreme programming (XP), feature-driven development (FDD), and dynamic system development methods (DSDM).

CrowFeID is not only aiming to support the feedback process in the iteration review but also to ensure the quality of the product increment before it is released. This can be achieved thru customers' involvement in the verification and validation process to check upon the acceptance criteria and changes required. Besides, timeliness (through schedule or time-based feedback) can also be achieved to support the iteration planning. Furthermore, the CrowFeID feedback attributes help customers provide directed and concrete feedback. Even though it is good to get various opinions from customers but it is necessary to avoid customers writing opinions that are diverse and deviate from the original purpose.

For the collaborative work approach, a suitable communication medium is used to enable timely review sessions, and the collection of the feedback attributes can be captured, stored, and analyzed to produce valuable information in a QA report. The CrowFeID uses a semi-rich communication medium, an asynchronous method in a collaborative way. The communication medium can support the task assignment and document sharing, capture structured feedback attributes for timely compilation, and online reporting.

Fig.2 shows the illustration of the CrowFeID approach. The customer's feedback as qualitative input use to aid in the decision-making process, especially by the Product Owner for backlog prioritization and the acceptance of the product increment.

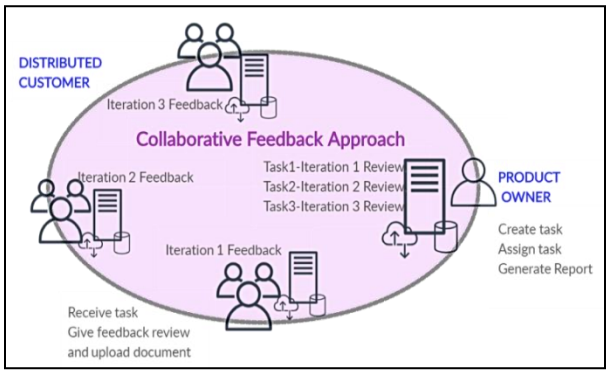


Fig. 2 Collaborative feedback between Product Owner and distributed customers

Three roles involved in CrowFeID are the Product Owner, Customer, and Lead Customer. Their responsibilities are as follows:

(i) Product Owner
- Create Iteration Review Task
- Add Iteration Review Task Details
- Add Reviewer or Assign Task to Reviewer
- Search Iteration Review Task Assigned to All Users
- Search Informed Decision
- Generate QA report

(ii) Customer
- Feedback for Iteration Review Task
- Search Iteration Review Task Assigned to User
- Search Iteration Review Task Assigned to All Users (Lead customer)
- Search Informed Decision (Lead customer)

CrowFeID allows the Product Owner to create tasks regarding the product increment and assign the task to the respective customers to get their feedback. The customer performs requirements validation by checking the product increment against acceptance criteria (product backlog or sprint backlog) and deciding on the acceptance status.

The customer is responsible to inform the Product Owner about the validation status, comment, and acceptance level of each requirement. For example, the requirement has been validated, not validated, or partially validated; the feature accepted, changed due to changing requirement, or rejected. CrowFeID feedback attributes are as highlighted in Fig.3.

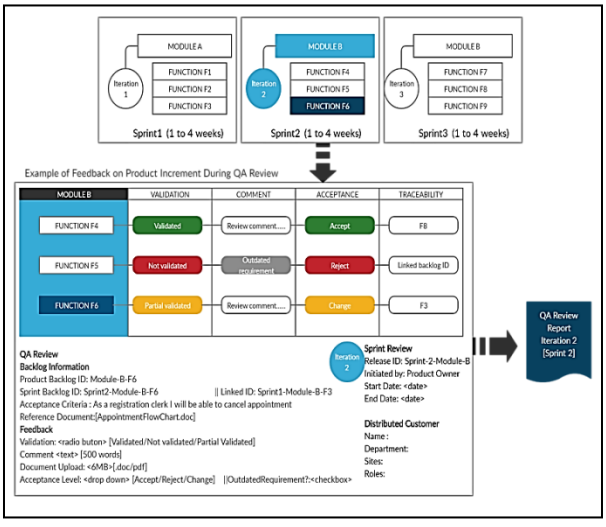


Fig. 3 CrowFeID feedback attributes on product increment during iteration review

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

Besides, to support the QA, CrowFeID provides rules and guidelines (refer to Table 2) for reporting feedback and decision making. The report generated from the online tool is a compilation of feedbacks per backlog item and related issues that will use as references in the next sprint meetings and prioritizing requirements.

Table 2. Rule And Guideline For Decision-Making Using Crowfeid In Iteration Review

Validation and Verification			Iteration Review Status Per Backlog ID	Decision-Making Justification
Validation / Acceptance / Outdated				
Validated	Accept	NO	DONE / COMPLETE	The Increment (refer to release ID) has been VALIDATED and this means that the requirement (refer to sprint product backlog ID) remains AS IS. Based on feedback review ACCEPT on the increment, it means that increment IS COMPLY with the specified acceptance criteria (refer to acceptance criteria of the sprint product backlog ID). In report means the complete or done status will be given.
Validated	Reject	NO	NOT DONE / REWORK	The Increment (refer to release ID) has been VALIDATED BUT feedback review found that the increment is REJECT because NOT COMPLY with the specified acceptance criteria (refer to acceptance criteria of the sprint product backlog ID). This also means that the requirement (refer to sprint product backlog ID) remains AS IS (it also means that the increment needs to revalidate -- rework for future sprint review).
Not validated	Reject	YES	REMOVE	The Increment (refer to release ID) has NOT VALIDATED BECAUSE feedback review found that the increment is REJECT because considered obsolete Requirement (at the time the review is done sprint product backlog ID is no more needed.) This means that the requirement (refer to sprint product backlog ID) considered OUTDATED and no need to validate (considered to be removed later in the report).
Partially validated	Change	YES	NOT DONE / CHANGE	The Increment (refer to release ID) has PARTIALLY VALIDATED BECAUSE feedback review found that the increment needs to CHANGE because considered has changing Requirement (at the time the review is done sprint product backlog ID has changed.) This means that the requirement (refer to sprint product backlog ID) considered OUTDATED and need to re-validate.

IV. EVALUATION

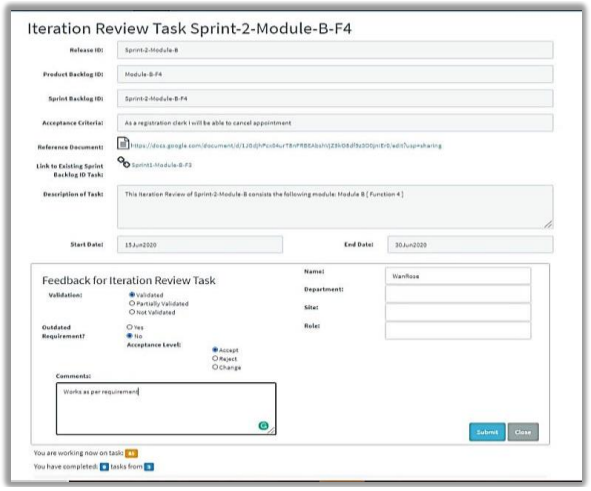
This section presents the materials and procedure for the evaluation that has been conducted. The results are also discussed in this section.

A. CrowFeID Tool

It is an online platform tool to support the proposed collaborative feedback approach to engage distributed customers and reduce delayed feedback. Some of the main features are:

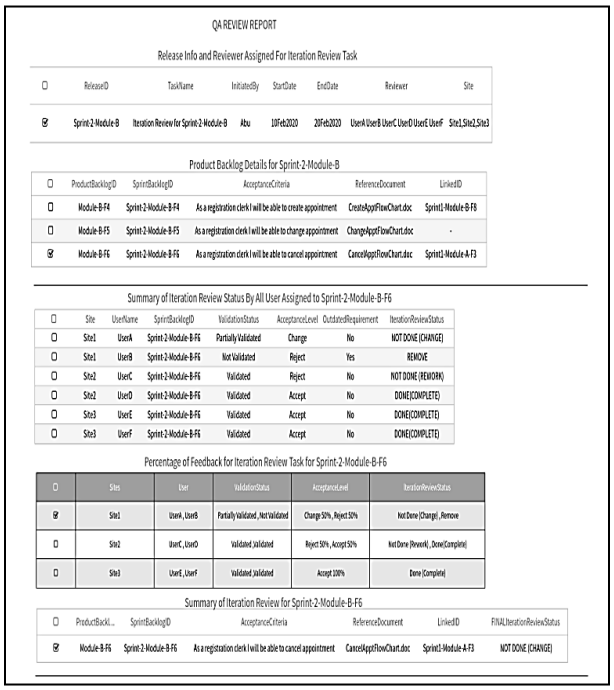
- an online platform that can be accessed anywhere and anytime.
- create and assign tasks for the customers
- capture feedback and record the evaluation result for each backlog item.
- capture multiple feedbacks for a single task from different customers.
- provide traceability or link between the requirements to know the dependency related to a change request.

Fig. 4(a) and (b) provides examples of the CrowFeID interfaces.



(a)

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731



(b)

Fig. 4. (a) Feedback for Iteration Review Task; (b) QA Report.

B. Case Study

The case study used in this evaluation is based on a government pilot project to develop a centralized clinical health system in public health clinics. This system has few modules: patient management, appointment, clinical documentation, billing, report, lab, x-ray, and a few more specific health specialist modules. The system used to record Electronic Medical Records (EMR) of a patient and support the lifetime health record concept to provide continuous healthcare for patients visiting the public health clinics. The stakeholders involved in the project including the top management and domain specialist, as well as the clinic staffs such as consultants, doctors, nurses, x-rays and laboratory technician, and a few others that use the system for their daily work-related with patient record in the clinic. There was a different degree of users' involvement within this project depending on the roles and responsibilities appointed to them.

C. Procedure

There are ten (10) participants (i.e., representing the customers) that have been invited for the evaluation. The evaluation was carried out through the following steps:

(i). Participants were provided with an example of a scenario for them to understand the process, roles, and responsibilities in Agile-Scrum methodology

(ii). Participants were given details regarding the modules, iteration, and release information to get a clearer structure of the project product increment. Participants were also given CrowFeID's user manual and explanation regarding the CrowFeID approach

(iii). Participants were given the URL, user ID, and password to access the system (i.e., CrowFeID). They start using the system based on the roles and responsibilities were assigned to them.

(iv). A questionnaire was given to the participants at the end of the process. The questionnaire was developed using Google Form, and participants accessed the questionnaire via URL link thru WhatsApp communication. The questionnaire consists of 34 questions and is divided into two categories: to evaluate the effectiveness of the proposed approach and to evaluate the usefulness of the tool.

D. Results

Based on the questionnaire responses, most participants were satisfied and agreed that the proposed approach meets its objectives.

The majority of participants where there are more than 50% of participants are agreed that the proposed approach has improved the current feedback process in iteration review and helped the iteration planning decision-making process. All also agreed that implementing CrowFeID has helped the Product Owner to identify what has deviated from the iteration plan, issues and concerns for the next meeting. All participants agreed that the Product Owner can make better decisions in backlog management and planning tasks (e.g., to remain, update, or remove the requirement that has been developed in another product increment, to prioritize backlog to be developed in the next release).

Besides, more than 50% show their confidence level that the tool has eased the iteration review process where the results shown a major satisfaction based on the positive feedback received. In which, the majority agreed that CrowFeID eases the review process by providing a suitable platform to improve customers' involvement in giving feedback regardless of their location and avoid the compilation overhead.

Fig. 5 shows an example of the responses regarding CrowFeID as a communication medium that has helped in

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

reducing delayed feedback and improving the quality of the product increment.

E. Limitations

Factors to be concerned with are the selection of participants in the evaluation that comes from different background seven though IT-related, besides considering if the feedback process was not carried out properly in a real setting. Those could affect the opinions given that may affect the dependent variables without the researcher's knowledge.

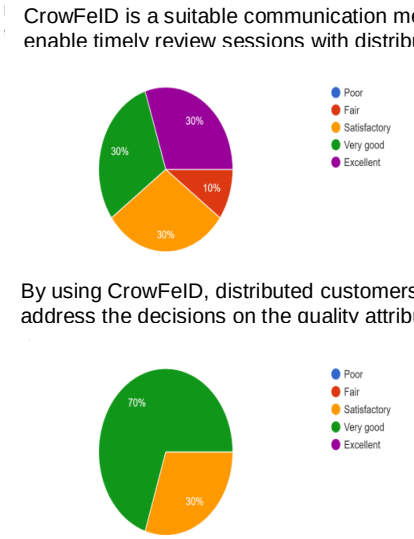


Fig. 5. Example of the responses from the evaluation

V. CONCLUSION

Customers' feedbacks in agile software development are useful input for the Product Owner to identify any improvement or changes required, and plan for the next iteration. However, delays in getting feedback are common thus become more challenging if the customers are at the distributed location. Moreover, getting feedback from specific customers (e.g., developer or customers' representative)during iteration review in agile software development is not adequate. The involvement of many customers in the feedback process during the review is very important to determine the quality of product increment. Thus, in this paper, a strategic process enhancement for the current feedback process and a tool is proposed to enable iteration review of customers (regardless of locations and time) through a collaborative feedback approach towards quality of product increment. Besides, the collaborative feedback also aims to aid the Product Owner in the decision-making process during iteration planning. The proposed work called CrowFeID adopts a QA mechanism in agile software development, focused on the feedback process during the review session. Furthermore, by engaging customers for an informed decision, quality assurance will be more seamless within the iterative development. Notification of requirement changes and backlog management can be effectively propagated to all parties in the development team.

As for future work, CrowFeID will be enhanced to provide service in a continuous feedback environment. In which, we believed that it is important to have an infrastructure that able to support a continuous feedback environment in agile software development.

ACKNOWLEDGMENT

This work supported by Fundamental Research Grant Scheme of Ministry of Higher Education (Project No.: 08-01-20-2313FR), and Universiti Putra Malaysia.

REFERENCES

[1] Amjad, S., Ahmad, N., Saba, T., Anjum, A., Manzoor, U., Balubaid, M. A., & Malik, S. U. R., 2017. Calculating Completeness of Agile Scope in Scaled Agile Development. *IEEE Access*, *6*, 5822–5847. https://doi.org/10.1109/ACCESS.2017.2765351

[2] Sirshar, M., & Arif, F., 2012. Evaluation of Quality Assurance Factors in Agile Methodologies. *International Journal of Advanced Computer Science*, *2*(2), 73–78.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Ken Schwaber & Jeff Sutherland. 2012. *Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust.* Wiley.

[8] 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.

[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] 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.*

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731

Electr. Electron. Eng. Inf. Commun. Technol. CHILECON 2017 - Proc., vol. 2017-Janua, pp.1–7.

[11] Green, R., Mazzuchi, T., & Sarkani, S., 2010. Communication and Quality in Distributed Agile Development : An Empirical Case Study, 322–328.

[12] Krusche, S., & Alperowitz, L., 2014. Introduction of continuous delivery in multi-customer project courses. *36th International Conference on Software Engineering, ICSE Companion 2014 - Proceedings*, 335–343. https://doi.org/10.1145/2591062.2591163

[13] Krusche, S., Alperowitz, L., Bruegge, B., & Wagner, M. O., 2014. Rugby: An agile process model based on continuous delivery. *1st International Workshop on Rapid Continuous Software Engineering, RCoSE 2014 - Proceedings*, 42–50. https://doi.org/10.1145/2593812.2593818

[14] S. Klepper, S. Krusche, S. Peters, B. Bruegge and L. Alperowitz, 2015. Introducing Continuous Delivery of Mobile Apps in a Corporate Environment: A Case Study. *2015 IEEE/ACM 2nd International Workshop on Rapid Continuous Software Engineering*, Florence, Italy, 2015, pp. 5-11, doi: 10.1109/RCoSE.2015.9.

[15] Oriol, M., Stade, M., Fotrousi, F., Nadal, S., Varga, J., Seyff, N., Schmidt, O., 2018. FAME: Supporting continuous requirements elicitation by combining user feedback and monitoring. *Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018*, *0*, 217–227. https://doi.org/10.1109/RE.2018.00030

[16] Shrivastava, S. V., and Rathod, U., 2017. A risk management framework for distributed agile projects. *Information and Software Technology*, *85*, pp.1–15.

[17] Shuib, R. and Hassan, S., 2021. Towards adopting software quality assurance in agile development methodology. *Turkish Journal of Computer and Mathematics Education*, vol. 12 (3), pp. 2152-2157.

[18] Wil, C.S.C., Ban, A., Hassan, S, Pa, N.C., and 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.

Biographical Notes:

Rosnani Shuib received the B. Information Technology degree from the Universiti Utara Malaysia in 2002, and has completed her education for degree in Master in Software Engineering from the Universiti Putra Malaysia in 2020.
She is currently an Information Technology Officer at the Prime Minister's Department of Malaysia. She has involved as project team in outsource and in-house government ICT projects. Her interests include software engineering, quality assurance and ICT project management.

Sa'adah Hassan received the B. Comp. Sc. degree from the Universiti Teknologi Malaysia, Master of Software Engineering degree from the University of Malaya, and a Ph.D. degree from University of Ulster, Northern Ireland, United Kingdom.
She is currently an Associate Professor at the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. Her research interests include software engineering, intelligent systems, and management information systems.

ISSN (Print): 2204-0595
ISSN (Online): 2203-1731