A Comparison of Scrum and Kanban for Identifying their Selection Factors

Mashal Alqudah, Rozilawati Razali
Research Center for Software Technology and Management, Faculty of Information Science and Technology
Universiti Kebangsaan Malaysia, (UKM)
Bangi, Selangor, Malaysia
E-mail: mashal_alqudah@yahoo.com[1], rozilawati@ukm.edu.my[2]

***Abstract*—Scrum and Kanban methods handle and manage the progress of software development. These methods are used in different situations and workflows. Hence, they are effective for different Agile team members and projects in diverse situations. However, the use of inappropriate method or practice leads to software development processes that are inflexible and wasteful, impacting the organization and causing inefficient development. Adopters of Scrum and Kanban believe that wrong practices of both methods are risky and thus, Agile team members should be assisted in their decision making. The aim of this study is to determine the main factors to consider during the selection of Kanban and Scrum method. The identification of the factors was conducted through in-depth review of the relevant work, after which, content analysis was employed in the analysis of data. As discovered, method prescription, roles and responsibilities, adoption time, team size, batch size, requirements prioritization, feature size, lead time, technical practices, cost and quality were the main factors that guide Agile team members in choosing Scrum or Kanban as well as their practices.**

Keywords—Scrum; Kanban; Selection factors

I. INTRODUCTION

Agile method is the main concept for a group of software development methods that correspond to four values and twelve principles of the Agile Manifesto [2]. Scrum and Kanban methods are developed based on these values and principles [2]. The values emphasize on: "individuals and interactions, working software, customer collaboration, and responding to change" [2]. Alliance refines the values captured in their manifesto into 12 principles [2], the following are the examples of the principles [2]:

- Stakeholders (customers and users) and developers must work together daily throughout the project.
- Projects should be built around motivated individuals.
- Face-to-face conversation is the most effective and efficient method of conveying information to and within a delivery and development team.
- Working software, which quantify the business value, is the best way to measure progress.
- Sustainable delivery and development should be ensured by maintaining a constant pace.

A range of practices have been developed to use with Scrum and Kanban methods to cover and achieve the values and the principles (See TABLE I and TABLE II). The 2013 IT Project Success Rates Survey Results affirmed the superiority of Agile methods with respect to effectiveness and success as opposed to the traditional methods [3-5]. Scrum is among the first-generation Agile methods [6, 8]. It is among the popularly selected Agile methods when firms switch to Agile approach [7, 9-11, 43, 46, 47]. Meanwhile, Kanban is the second-generation Agile methods [4, 8]. This method is popularly used by organizations globally, particularly those operating in Europe [12, 13]. Both methods have been used by thousands of firms around the world [35]. However, Agile team members agree that one method does not fit all [12, 39], neither Kanban nor Scrum is suitable for all projects [12]. Hence, a comparison is needed since both of them focus on project management.

Scrum is a minimal array of practices used extensively within the industry [9] as defined by Scrum Alliance. The three roles of a Scrum team are: the Product Owner (PO), a Scrum Master (SM), and a Development Team [1, 36]. In Scrum, each team handles their tasks by way of four artefacts: a Product Backlog, a Sprint Backlog, a Product Increment, and Definition of Done [1]. The Scrum team is required to perform five activities to achieve its goals. These activities are: Backlog Refinement, Sprint Planning, Daily Scrum Meeting, Sprint Reviews, and Sprint Retrospective. Scrum is thus regarded as a prescriptive method [1].

In comparison to Scrum, the Kanban method is actually less detailed in terms of its required practices and principles; Kanban is less rigid in comparison to Scrum [4, 6]. Still, Kanban is regarded as a powerful method since this method has the capacity to handle the challenges that the previous methods could not [8] and proves to be more effective for teams in carrying business value incrementally [8]. In the implementation of Kanban, generally, the main practices that enable process evolution are; visualizing the workflow by the use of Kanban board, limiting work in progress by reducing the number of features to be implemented, manage and measure flow, make policies explicit, implementing feedback and finally looping and improving collaboratively and continuously [16]. Similar to Scrum, Kanban is highly dependent on self-organized teams. The method also requires high level of involvement of leadership in the creation of other practices, which ease the Agile usage in the organization. Nonetheless, selecting the right method will determine the success of a software development [17]. Therefore, this study primarily aims to discover the factors that help the members of Agile team in their choice of Scrum and Kanban or in using these

This research under Fundamental Research Grant Scheme project (FRGS/2/2013/ICT01/UKM/02/2).

978-1-5386-0475-5/17/$31.00 ©2017 IEEE

methods together in the implementation of numerous projects, taking into account the methods' nature and constraints so that successful outcome can be attained. Thus, both methods are reviewed first in order to obtain knowledge on their practices, principles, differences and similarities, while taking into consideration the determination of selection factors.

Different factors have to be considered when selecting the suitable Agile method [18], or scaling Agile methods [19]. To ensure successful software projects, team members of any organization must be concerned with project management [20]. In this regard, both Kanban and Scrum methods are appropriate in managing the projects and developing them successfully [23]. This proceeding review of the current studies demonstrates that the decision to select either the Scrum or the Kanban method or develop a hybrid method of both requires a comprehension of the factors impacting the selection. It is also important to balance the strengths of each method according to the value of method prescription, roles and responsibilities, adoption time, team size, batch size, requirements prioritization, feature size, lead time, technical practices, cost and quality as this would help the team in choosing whether to use a single fitting method or adopting a hybrid one [23].

II. Related work

This section discusses a brief background of Scrum and Kanban methods when they started to emerge as well as each of their main roles, practices and definitions.

A. Scrum Method

Scrum refers to a framework that enables the Agile team members to create and maintain complex products and it is deemed to be a method characterized by simplicity and lightweight. Nevertheless, the method is still challenging to fully learn and implement [1]. Additionally, Scrum enables the team members to use different processes and methods [1]. Scrum is bound by rules that join the events, roles and artifacts and govern relationships and interactions among them. It lays stress on self-organization of teams that are not driven by external members. Moreover, Agile members using Scrum have to comprise a team that is small enough to remain flexible but large enough to achieve considerable work in a sprint. More specifically, 5 to 11 team members in a team, with the inclusion of the Product Owner and the Scrum Master, is considered to be the most suitable size for the development team [1].

Contrasting to Kanban, Scrum lays down various roles [13, 21, 22]. For instance, the product owner is primarily responsible for the management of product backlog, the development team is responsible for delivering a releasable increment of product after each sprint, while the Scrum master is responsible for making sure that Scrum method is comprehended by all and is effectively used [1, 22]. The Scrum master is a servant-leader of the members of the development team. Table I presents the Scrum method practices including a brief definition of each one.

TABLE I. Scrum practices definitions.

Practices, Artifacts and Activities of Scrum	Definition
Product Backlog [1]	Product backlog is described as the list of the entire features and requirements that need to be implemented [1, 31, 34]. The creation and maintenance of the product backlog lie in the hands of the product owner, and with each item/feature of the product backlog, a priority and estimated effort is appropriated [1, 48].
Sprint Backlog [1]	A sprint backlog is described as a set of activities that needs performing throughout a sprint [1, 31, 34]. Every activity or set of activities is a representation of an item of product backlog that has to be included in the next release, to be introduced after the current sprint ends [1, 31].
Sprint Burn down Chart [1]	This chart represents the remaining sprint backlog work that has to be daily updated, within which the assessment of the progress are assessed against the iteration goals [1, 15]. Experts generally have to estimate the accomplishment of the remaining tasks/stories for every sprint backlog [1].
Effort Estimation [1]	The entire members that constitute the team using Scrum methods make use of difficulty level and effort level for work estimation [1, 31]. Effort scaling, an abstracted metric, is also used (e.g., numeric sizing or t-shirt sizing) [1].
Sprint [1]	In the heart of Scrum lies Sprint, which is referred to as a time-box of one month or less, and within the duration, a completed, useable and releasable product increment is developed [1, 36]. On the basis of this empirical research, the findings indicated that majority of informants that applied Scrum stayed within a two week duration sprint [1]. However, it can be a four or six months sprint [31, 36]. Moreover, changes cannot be made after the team decides on the sprint to be followed but in some instances, the product owner can cancel the said sprint although it may incur resources [1].
Sprint Planning [1]	Sprint planning primarily aims to reach a decision as to what can be delivered in the current 2 to 4 weeks sprint and the way the work will be achieved to deliver the increment [1, 36].
Daily Stand-up Meetings [1]	This reflects the meetings that are held daily for around 15 minutes with the aim towards analyzing the project progress, with positive and negative feedback [1]. Also, in the meeting, the unexpected issues that could delay the project are brought up, along with the work to be conducted prior to the next meeting [1, 32, 39]. For large teams, daily Scrum will not provide the expected value for the team members [36, 37].
Sprint Review [1]	After each sprint, the members of the team including the customer, hold a review meeting that is attended by the entire stakeholders, during which all the features of the new release are demonstrated [1, 31]. It was proven that customer feedback during sprint review meeting increased the quality of each sprint [33]. Hence, customer preferences and feedback is needed [44, 45].
Sprint Retrospective [1]	A sprint retrospective aims to promote discussion among the team members concerning the issues that they encounter and how they can be handled in the future sprints [1, 32]. This enables developers to hone their effectiveness through behavioral adjustment based on the circumstances encountered [1, 32, 33].

B. Kanban Method

The Kanban method was introduced by David J. Anderson in 2004 while he was attempting to assist a small Microsoft IT team from failing [13]. But Anderson only forwarded the details of Kanban to a software development after 6 years, in 2010 [16, 21]. Kanban primarily aims to safeguard the team from the unending tasks appropriated by management, with an attempt to realize ongoing development pace and adaption similar to other Agile methods, where little to no resistance to change arises [21, 27]. More specifically, Kanban method has its basis on the Just-in-Time (JIT) premise, particularly when it deals with software development, indicating that the team should refrain from development of software features prior to their demand. In other words, the team is not supported to develop or deliver features that are not required in order to mitigate waste, and maintain the same to be closer to zero [28].

The Kanban method makes use of visual Kanban board to improve software development through the display of different phases of the process of development. It is not as prescriptive as Scrum in that it does not have defined roles, no emphasis on meetings and no artifacts [5]. The major principles of Kanban is workflow visualization through the Kanban board, limiting work in progress by minimizing the number of features to be implemented, management and measurement of flow, making clear policies, implementing feedback, looping and enhancing collaboration in a continuous manner [16]. The practices and definition of Kanban are presented in table II as follows.

TABLE II. KANBAN PRACTICES DEFINITIONS.

Practices	Definition
Visualize the workflow [16]	Visualizing the workflow entails making the work required to be visible for the entire team members as invisible work contributes to the risk of project implementation. Through the Kanban board, the workflow can be clearly visualized [16, 23].
Limit work in progress [16]	This is the main practice in Kanban that refers to limiting the amount of work in progress (WIP) or curtailing the WIP features [16, 41]. In this regard, it was evidenced that when the amount of multitasking is reduced, the time to deliver the features is also reduced [8, 16].
Manage and measure flow [16]	Flow management primarily aims to finish the work in progress in a way that each state in the workflow is monitored and reported in what is known as measuring flow. Therefore, the movement will be expedient and smooth, creating timely value, mitigating risk and avoiding cost of delay [16].
Make policies explicit [16]	Majority of organizations possess implicit policies considering the various work types to be handled. Kanban teams lay down the policies explicitly for their consistent application [8, 16, 23].
Implement feedback [16]	Kanban needs feedback loops to work and as such [16, 41], it uses standup meeting, service delivery review, operations review and risk review to enable the comparison between expected outcomes and actual outcomes, and to make the necessary adjustments [16].
Continuous improvement [16]	The clarified and shared understanding regarding work theories, workflow, process and risk help the team members to develop a shared understanding of a problem and to suggest enhancements that can be agreed on unanimously [16]. The team should employ a Kaizen mind [29, 42] that is deemed to be a major part of Kanban's effective use.

III. RESEARCH METHOD

This research is conducted mainly to deliberate on the Kanban and Scrum methods and determine their suitability in projects. Thus, a review was performed in order to ascertain the pertinent and available studies in describing the criteria of the methods. Then, the differences and similarities of both methods are highlighted. This research also looks into the factors that impact the selection of both methods. The review was according to keywords, study type, language and year of publication.

The keywords in the search include selection factors, selection criteria, Kanban and Scrum. As for the inclusion criteria for each category, the key points shown below were used:

- Studies elaborating the methods of Scrum and Kanban.
- Studies illustrating a comparison between diverse methods of Scrum and Kanban.
- Studies exploring the practices, roles and/or responsibilities of the methods of Scrum and Kanban.
- Studies exploring the factors impacting the selection of the methods of Scrum and Kanban.

The publications as early as year 2001 were selected in this study. The year 2001 was when Agile methods became the subject of interest within the research community and industry. Qualitative analysis was performed on the selected studies; in particular, content analysis was used through the use of NVivo software. Content analysis comprises a scientific method, where a summary and analysis of textual messages obtained was used in the comparison of the Scrum and Kanban methods in-depth. Content analysis entails the direct compression of words in a text. Thus, according to the explicit coding rules, the established content categories were fewer in terms of amount [30]. During the process of coding, labeling was done on each text segment. The range of text segment could be from a few words to the entire paragraph. Through coding, interrelated words, sentences or paragraphs could be rearranged and integrated. This would create meaningful data [30]. In the context of this study, the results entail the comparisons of Scrum and Kanban as well as the factors impacting the selection of both methods.

IV. FINDINGS AND DISCUSSION

Both Kanban and Scrum methods could be used in different contexts as shown in Table III. Based on this review it was found that the Scrum method has been extensively adopted more than Kanban [9-11]. The differences between Kanban and Scrum methods as highlighted in literature are enumerated in Table III below.

The comparison between Scrum and Kanban shows that the former is a prescribed method with distinct roles and responsibilities for the development team [1, 4, 5]. Added to this, in Kanban, the size of the team is more flexible that in Scrum, with the batch size of the Scrum requirement being one, two or four weeks sprint, whereas Kanban batch size is comparatively small (it can be as small as hourly batch) [6]. In relation to this, Kanban allow prioritized requirements daily,

while requirements have to be prioritized on the basis of the length of the sprint when following Scrum [6]. Moreover, the size of the feature is very small in Kanban compared to that of Scrum [8], and the lead time (the length of time between proposing a new feature/making a request and its deployment to the environment of the customer) for one organization is mitigated by 50% when shifting from Scrum to Kanban [8]. Literature also shows that estimates and timeboxes that are not accurate provided longer lead times, which cause waste [21]. In addition, Kanban focuses of cutting cost more than Scrum [11] especially for operations [12]. It also focuses more on improving quality [12, 21]. As for

Their similarity, both Scrum and Kanban do not have technical practices.

TABLE III. SCRUM AND KANBAN PRACTICES DEFINITIONS.

Criteria	Scrum	Kanban
Method Prescription	Prescribed in detail with transparency [1, 39]	Not prescribed and is characterized by more flexibility compared to Scrum [4, 6, 39]
Roles & Responsibilities	Predefined roles and responsibilities for every team member, including product owner, Scrum master and others [1, 7, 38, 39, 42]	No predefined roles and responsibilities [5, 6, 38, 39, 42]
Adoption time	Challenges in transition may be encountered [8]. However, most of the organizations adopt Scrum before Kanban [12]	Easier transition [8], particularly when teams that maintain and operate projects want to shift from traditional methods [12]
Team size	A team comprising of 5-11 development team members with the inclusion of both product owner and Scrum master [1, 36]. If the team is large, it will be hard to provide value to all Agile team members [36]. Hence, for large teams, Kanban is better than Scrum [15]	Kanban has more flexibility when compared to Scrum [15], in that team can have less than 5 team members [14], or it can be more than 11 (up to 14 team member) [15]
Batch Size (WIP size)	Large batch size (less than one month sprint) is considered to be big especially when dealing continual changes [6, 12]. Team should commit on delivering sprints on time [6, 38, 39]	It has to have small batches with small amount of WIP [6, 8, 12, 38]. Thus, Kanban does not have one, two, or four weeks sprint like Scrum. This indicates that daily/hourly delivery of important items can be done [6]. Commitment is not compulsory [6, 38, 39]
Requirements prioritization	The requirements need prioritization on the basis of the length of the sprint, every two, three or four weeks [6, 42]	The prioritization of requirements is done continuously, which can be daily/hourly [6, 42]
Feature size	It is characterized by small-feature size [8, 12]	The feature can be categorized into smaller pieces, which can prove useful when the experienced team wants faster feedback [8, 12]
lead time	Scrum avoids cutting lead time unlike Kanban [21]	Kanban cuts lead time by steering clear of multi-tasking and limiting the WIP features [12, 13, 22, 39]
Technical practices	Scrum has no technical practices [24, 42]	Like Scrum, Kanban has no technical practices [25, 42]
Cost	Unlike Kanban, Scrum avoids cost saving but it focuses more on knowledge, experience and decision making based on what is known [11]	Focuses on cutting cost when compared to Scrum [11], especially for operations [12]
Quality	Sprint review meeting is the main practice for improving quality in the Scrum method [26]	Kanban focuses more on improving quality when compared to Scrum [12, 21]

Based on table III above, the following paragraphs illustrate the suitability of each method and how the selection could be done in both cases (Kanban and Scrum).

Case 1: Kanban Selection

When the Agile team members do not prefer to follow the prescribed methods, they do not prefer to follow the predefined roles and responsibilities, they will consider the transition into Kanban easy. Kanban is suitable if the team size comprises of 3-14 development members and they face difficulties in batching the work into one, two, or four weeks sprint or there is a need to batch daily/hourly and to do requirements prioritization on a daily/hourly basis. In selecting Kanban, the features can be categorized into smaller pieces, while delivering value in less than one week (daily/hourly). Also, if there is a need to focus more on cutting lead time, on improving quality, and on cutting cost, then Kanban is more appropriate.

Case 2: Scrum Selection

When the Agile team members prefer to follow the prescribed methods; they prefer to follow the predefined roles and responsibilities, they will consider the transition into Scrum easy. With Scrum, the team size comprises of 5-11 development members and the batching size for one, two, or four weeks sprint, is considered to be easy in terms of estimation. If the requirements prioritization flows into one, two or four weeks sprint and the feature size is characterized by small-feature size which can be done in one, two or four weeks sprint, then Scrum is the suitable option. The cutting lead time and cost saving are not emphasized in Scrum and the team focuses more on knowledge, experience and decision making based on what is known. Also, if quality is not highly emphasized, then Scrum is more appropriate.

V. CONCLUSIONS

The Agile team members of software development are met with challenges in their adoption of Scrum and Kanban methods. Agile team members find it challenging to choose

Scrum over Kanban, or Kanban over Scrum or these two methods in combination, when they are implementing a specific project. Consideration has to be made on a range of selection factors during the selection of the Scrum and Kanban methods. Agile team members should be assisted when making this crucial decision. Thus, in-depth review of literature was conducted in this study to explore the selection factors for Scrum and Kanban. Both Kanban and Scrum can function in tandem. However, the method prescription, roles and responsibilities, adoption time, team size, batch size, requirements prioritization, feature size, lead time, technical practices, cost and quality are the major factors that can help the Agile team members in their selection for a multitude of purposes. These factors lead to minimized waste and the possibility of project delay. An empirical investigation, of how and when to use each method, for different projects will be undertaken by the researchers to make accurate selections.

ACKNOWLEDGMENT

We acknowledge the Faculty of Information Science and Technology at Universiti Kebangsaan Malaysia (UKM) for supporting.

REFERENCES

[1] J. Sutherland and K. Schwaber, "The SCRUM guide. The definitive guide to SCRUM: The rules of the game," SCRUM. orgOctober, 2013.

[2] A. Alliance, "Agile manifesto," Online at http://www. agilemanifesto. org, vol. 6, 2001.

[3] S.W. Ambler, “The Non-Existent Software Crisis: Debunking the Chaos Report,” Available: http://www.drdobbs.com/architecture-and-design/the-non-existent-software-crisis-debunki/240165910, 11 September, 2014.

[4] N. Nikitina and M. Kajko-Mattsson, "Developer-driven big-bang process transition from SCRUM to Kanban," in Proceedings of the 2011 international conference on software and systems process, 2011, pp. 159-168.

[5] H. Kniberg and M. Skarin, Kanban and SCRUM-making the most of both: Lulu. com, 2010.

[6] R. Cuellar, "Kanban for Help Desks: Managing the Unplannable," Cutter IT Journal, vol. 24, p. 23, 2011.

[7] R.H. Al-Ta’ani and R. Razali, "A Framework for Requirements Prioritisation Process in an Agile Software Development Environment: Empirical Study," International Journal on Advanced Science, Engineering and Information Technology, vol. 6, pp. 846-856, 2016.

[8] A. Shalloway, "Demystifying Kanban," Cutter IT Journal, vol. 24, p. 12, 2011.

[9] A.M.M. Hamed and H. Abushama, "Popular agile approaches in software development: Review and analysis," in Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on, 2013, pp. 160-166.

[10] R.V. Anand and M. Dinakaran, "Popular Agile Methods in Software Development: Review and Analysis," International Journal of Applied Engineering Research, vol. 11, pp. 3433-3437, 2016.

[11] A.S. Campanelli and F. S. Parreiras, "Agile methods tailoring–A systematic literature review," Journal of Systems and Software, vol. 110, pp. 85-100, 2015.

[12] D. Verweij and O. Maassen, "Kanban at an Insurance Company in the Netherlands," The Viral Growth of Kanban in the Enterprise, Cutter IT Journal, 2011.

[13] D.J. Anderson and A. Roock, "An agile evolution: why Kanban is catching on in Germany and around the world," Cutter IT Journal, vol. 24, p. 6, 2011.

[14] P. Brodzinski, “When Kanban is the Best Choice”. Available: http://brodzinski.com/2010/06/kanban-best-choice.html. (2010,April).

[15] V. Moskalenko, “SCRUM vs. Kanban: Which to Choose for Agile Development?" Available: https://www.luxoft.com/blog/vmoskalenko/SCRUM-vs-kanban-which-to-choose-for-agile-development/. (2015, March).

[16] D.J. Anderson, "Kanban," ed: Blue Hole Press Sequim, WA, 2010.

[17] Z. Mansor, S. Yahya, and N. H. Arshad, "Towards the Development of Success Determinants Charter for Agile Development Methodology," International Journal of Information technology and Engineering, vol. 2, pp. 1-7, 2011.

[18] M.K. Alqudah and R. Razali, "Key Factors for Selecting an Agile Method: A Systematic Literature Review," International Journal on Advanced Science, Engineering and Information Technology, vol. 7, pp. 526-537, 2017.

[19] M. Alqudah and R. Razali, "A Review of Scaling Agile Methods in Large Software Development," International Journal on Advanced Science, Engineering and Information Technology, vol. 6, 2016.

[20] A.A. Hamid and Z. Mansor, "Client’s Readiness Assessment Success Factors for Outsourcing Software Projects," International Journal on Advanced Science, Engineering and Information Technology, vol. 6, 2016.

[21] D.I. Sjøberg, A. Johnsen, and J. Solberg, "Quantifying the effect of using kanban versus SCRUM: A case study," IEEE software, vol. 29, pp. 47-53, 2012.

[22] S. Govindaraj and S. Tadipatri, "Use of Kanban in distributed offshore environments," The Viral Growth of Kanban in the Enterprise, Cutter IT Journal, 2011.

[23] H. Lei, F. Ganjeizadeh, P.K. Jayachandran, and P. Ozcan, "A statistical analysis of the effects of SCRUM and Kanban on software development projects," Robotics and Computer-Integrated Manufacturing, vol. 43, pp. 59-67, 2017.

[24] S.W. Ambler and M. Lines, "Going Beyond SCRUM: Disciplined Agile Delivery," Disciplined Agile Consortium. White Paper Series, 2013.

[25] A. Bolboaca, “Agile Challenge: Adopting Technical Practices,” Available: http://mozaicworks.com/blog/agile-challenge-adopting-technical-practices/. February, 2013.

[26] F. Kanwal, K. Junaid, and M.A. Fahiem, "A hybrid software architecture evaluation method for fdd-an agile process model," in Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on, pp. 1-5, 2010.

[27] A. Bolaji, "A cross-disciplinary systematic literature review on Kanban," Master’s Thesis. University of Oulu. 62 p. Available at: http://jultika. oulu. fi/files/nbnfioulu-201502111073. pdf, 2015.

[28] T. Skeie, "Does Limit on Work-In-Progress (WIP) in Software Development Matter?," 2014.

[29] B. Estácio, R. Prikladnicki, M. Morá, G. Notari, P. Caroli, and A. Olchik, "Software kaizen: Using agile to form high-perfomance software development teams," in Agile Conference (AGILE), pp. 1-10, 2014,.

[30] K. Krippendorff, “Content analysis: An introduction to its methodology: Sage,” 2012.

[31] V.T. Heikkila, M. Paasivaara, and C. Lassenius, "Scrumbut, but does it matter? a mixed-method study of the planning process of a multi-team scrum organization," in Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on, 2013, pp.85-94.

[32] Y. Andriyani, "Knowledge Management and Reflective Practice in Daily Stand-Up and Retrospective Meetings," in International Conference on Agile Software Development, 2017, pp. 285-291.

[33] A. Khosravi, T.J. Gandomani, and H. Fahimian, "Introduction of scrum in an elite team: A case study," Journal of Software, vol. 12, 2017, pp. 173-180.

[34] S. Ashraf and S. Aftab, "Latest Transformations in Scrum: A State of the Art Review," International Journal of Modern Education and Computer Science(IJMECS), Vol.9, No.7, pp.12-22, 2017.DOI: 10.5815/ijmecs. 2017.

[35] S. Denning, "Why Agile can be a game changer for managing continuous innovation in many industries," Strategy & Leadership, vol. 41, 2013, pp. 5-11.

[36] J. López-Martínez, R. Juárez-Ramírez, C. Huertas, S. Jiménez, and C. Guerra-García, "Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review," in Software Engineering Research and Innovation (CONISOFT), 2016 4th International Conference in, 2016, pp. 141-148.

[37] V.G. Stray, Y. Lindsjorn, and D.I. Sjoberg, "Obstacles to efficient daily meetings in agile development projects: A case study," in Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on, 2013, pp. 95-102.

[38] M.O. Ahmad, P. Kuvaja, M. Oivo, and J. Markkula, "Transition of software maintenance teams from Scrum to Kanban," in System Sciences (HICSS), 2016 49th Hawaii International Conference on, 2016, pp. 5427-5436.

[39] M. Alqudah and A.A. Abdallah, "Implementing computer-aided language learning tool using hybrid agile method: A case study," in Informatics and Creative Multimedia (ICICM), 2013 International Conference on, 2013, pp. 174-180.

[40] M. Alqudah and A. Abdulsalam, "Basic english language tools for beginners: Using animations and audio," International Journal of Scientific Engineering Research, vol. 4, 2013, pp. 228-233.

[41] M.O. Ahmad, J. Markkula, and M. Oivo, "Kanban in software development: A systematic literature review," in Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on, 2013, pp. 9-16.

[42] G.S. Matharu, A. Mishra, H. Singh, and P. Upadhyay, "Empirical study of agile software development methodologies: A comparative analysis," ACM SIGSOFT Software Engineering Notes, vol. 40, 2015, pp. 1-6.

[43] R. H. AL-TA'ANI and R. Razali, "Process Model for Systematic Requirements Prioritisation Process in an Agile Software Development Environment Based on 5S Approach: Empirical Study," Journal of Theoretical & Applied Information Technology, vol. 95, 2017.

[44] M. Alqudah, Y. Yusof, S. A. M. Noah, and A. Almabhouh, "Incorporating Prioritized User Preferences in Search System," International Journal on Advanced Science, Engineering and Information Technology, vol. 2, pp. 401-404, 2012.

[45] M. K. Al-Qudah, "A Weighted-Based Approach To Prioritize User Preferences In Information Retrieval," Universiti Utara Malaysia, 2008.

[46] O. N. Al-Allaf, "The Adoption of Agile Processes in Large Web Development Enterprises: A Survey in Jordan," IJWA, vol. 2, pp. 206-216, 2010.

[47] F. Ahmad, F. Baharom, and M. Husni, "Agile development methods for developing web application in small software firms," in Knowledge Management International Conference (KMICe), 2012.

[48] C. Kaur1st and V. Kumar 2nd, "Product Backlog Prioritization in Scrum: A Review," International Journal of Modern Computer Science (IJMCS) vol. Volume 3, Issue 2, June, 2015.