AIP Conference Proceedings

Identification of criteria affecting software project monitoring task of Agile Kanban method

Cite as: AIP Conference Proceedings **2016**, 020021 (2018); https://doi.org/10.1063/1.5055423
Published Online: 27 September 2018

Hamzah Alaidaros, Mazni Omar, and Rohaida Romli

Identification of Criteria Affecting Software Project Monitoring Task of Agile Kanban Method

Hamzah Alaidaros[1, a)], Mazni Omar[2, b)], and Rohaida Romli[3, c)]

[1, 2, 3] *Human-Centred Computing Research Lab, School of Computing, Universiti Utara Malaysia 06010 UUM Sintok, Kedah, Malaysia*

[a)] Corresponding authors: m7amza7@yahoo.com
[b)]mazni@uum.edu.my
[c)]aida@uum.edu.my

Abstract. Currently, software development organizations (SDOs) attempt to deliver their software products quickly, within the prescribed period, and with the highest quality and lowest cost, yet this is proven a big challenge for them. Agile Kanban method recently is gaining increasing attention and popularity, due to its numerous advantages that make it performs better than other methods in terms of managing software projects. Beside of that, it has a board that used to visualize the workflow and monitor the project progress. However, various studies have shown that this method has significant challenges that negatively impact the scheduling of the development process. Consequently, late delivery of software projects may occur, thus the rate of projects' failures will be increased. This paper aims to identify the challenges and criteria that affect the progress monitoring task of Agile Kanban method using a narrative review method, whereby this review method is adopted to identify and summarize what has been previously published, avoid duplications, and seek new study areas which are not yet addressed. Finally, the findings of this study would be used to develop an improved software project monitoring task model of Agile Kanban method. That model would be capable to keep projects progress as it is planned, and thus leading to successful delivering for software projects according to its specifications, time, and budget.

INTRODUCTION

Software project management (SPM) is a sub-discipline of project management in which software projects are planned, implemented, monitored and controlled [1]. Currently, software development organizations (SDOs) attempt to deliver their software products quickly, within the prescribed period, and with the highest quality and lowest cost [2]. However, delivering software projects according to specifications, time, and budget is proven a big challenge for SDOs [3, 4]. Due to the above issue, one of the important tasks in SPM is progress monitoring, whereby it ensures that projects' plan is progressed according to budget, schedule, and quality expectations [5, 6]. In this regard, successful implementation of software projects depends entirely on successful monitoring mechanisms, while the lack of monitoring software development projects (SDPs) leads to the failure of such projects [7-9].

Agile methods (AMs) recently are gaining wide recognition within SDOs due to their flexibility and effectiveness. It provides a shorter cycle for the development process with higher customer satisfaction [10]. In particular, Scrum and Kanban methods are considered as the two powerful AMs that focus on managing software projects, this is because they can optimize the development process by setting-up teams, managing time more effectively, and identifying the tasks [11]. Besides that, Scrumban method combines the most important practices of Scrum along with the core principles of Kanban [12]. Nevertheless, various studies, such as [10, 11 , 13 , 14], reported that Kanban method, currently, has popularity among AMs. This is because it has numerous advantages that make it performs better than other AMs and has greater consistency in managing software engineering (SE) projects.

Anderson [15], father of Kanban method in software development (SD), described Kanban as an approach to incremental, evolutionary process, and systems change for organizations. In addition, he has defined five principles

Proceedings of the 3rd International Conference on Applied Science and Technology (ICAST'18)
AIP Conf. Proc. 2016, 020021-1–020021-7; https://doi.org/10.1063/1.5055423
Published by AIP Publishing. 978-0-7354-1734-2/$30.00

for Kanban method, which are limit work in progress (WIP), visualize workflow, measure and manage flow, make process policies explicit, and use models to recognize improvement opportunities. Kanban method can enhance understanding, visibility, and controlling the workflow, as well as support the management through two core principles, which are limiting WIP and visualizing workflow by using Kanban board [10, 16, 17]. Moreover, Kanban method has a board used to visualize the workflow and monitor the project progress as it is shown in Fig 1. Furthermore, Kanban board visualizes the activities of the development process and keeps WIP in control [15, 18].

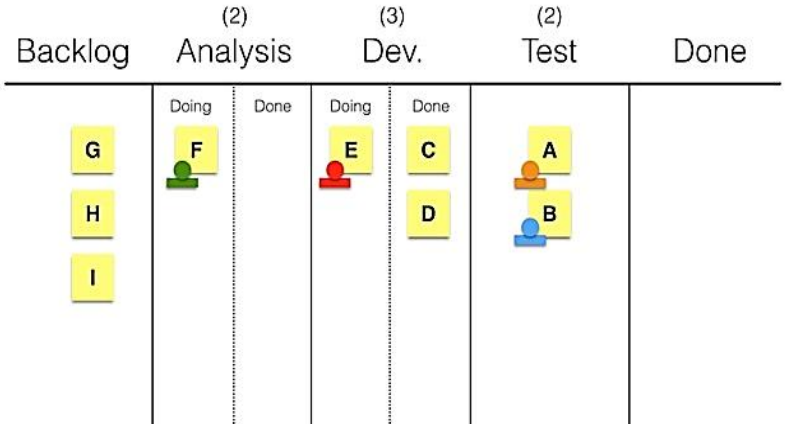

FIGURE 1. Agile Kanban board

Progress monitoring is an essential task during any project execution, and there is no exception for Agile projects. Besides being needed to monitor a project, accurate and timeliness reporting is important for keeping the management and team updated on the project's progress [5, 6]. However, the progress monitoring task of Agile Kanban method has significant lacks during the development process. This problem has negative impacts on software projects' success because the delays in project scheduling lead to late delivery [3], [13], [19], [20]. Therefore, a study that investigates the lacks in progress monitoring task by developing an improved model for Agile Kanban method could remedy this situation.

Based on the problem identified above, this paper aims to answer these questions: what are the current challenges in Agile Kanban method? and what are the criteria affecting software project monitoring task model of Agile Kanban method?. To achieve that, a narrative review method was used, whereby this review method is capable of criticizing and summarizing a body of literature and drawing conclusions about their research topics [21]. Moreover, Uman [22] stated that narrative review seeks to comprehensively use existing literature by focusing on a subset of information, and using a wide variety of sources that goes beyond the more rigid systematic review.

RESULTS AND DISCUSSION

The results of this study are provided and discussed in the next subsections.

The Current Challenges in Agile Kanban Method

This study revealed that ***Agile Kanban method has lack of mechanism for progress tracking***. Thus, it needs to be integrated with other methods because it does not have a standard definition for SD and its specific practices are not rigorously defined yet. In this regard, Flora and Chande [10] and Lindblom [23] claimed that Kanban method should be complemented or expanded by Agile method or other methods in SDOs to keep the project schedule as planned and work effectively. Consequently, this challenge has led to integrate Scrum with Kanban to introduce a new method called Scrumban [12], and integrate Kanban with a value stream mapping (VSM) [24]. Although previous studies have addressed this issue, these are few studies and still have limitations. Therefore, integrating a suitable monitoring method with Kanban method may contribute to improve the monitoring task of Agile Kanban method. In this context, recent study[14] has suggested to integrate earned value analysis (EVA) method and Agile Kanban method with the undertaking of the aforementioned studies limitations.

Another challenge concerns with limiting WIP principle, which is a core principle of Kanban method that defined the maximum count of tasks for each Kanban board stage. It is identified by the project manager to prevent roadblocks and make tasks move quickly on the board [18, 25-27]. Nevertheless, determining the WIP limits is proven a major challenge that is faced by software project practitioners, whereby ***there is a lack of adequate technique to determine the optimum number of WIP limits for each stage in Kanban board*** [19, 28]. In this study, the optimum WIP limits refer to suitable numbers for each stage that can monitor and control the team members with their tasks and ensure that project progress as planned. Typically, WIP limits are estimated; however, a bad estimation of initial WIP limits can be led to lags in scheduling of the development process of SD and failure in software delivery on the prescribed time [13, 19, 20, 28, 29]. Overall, even though Kanban method is good in monitoring project progress using WIP limits, it is still a challenge to determine the optimum WIP limits for each workflow stage in Kanban board.

Visualizing the workflow is also another core principle of Kanban method, which is defined as the process of highlighting the mechanisms, interactions, waiting, queues, and delays, which are involved in implementing a part of valuable software. A Kanban board is used to visualize workflow and monitor project progress by showing the development process activities [15, 18]. Within Kanban method, Cumulative Flow Diagram (CFD) is used to depict the average lead-time and WIP in order to show the issues and bottlenecks during the development process [17]. Nevertheless, Kanban board and CFD neither report on how much work is left nor provide some indications of where the project ought to be [28]. Generally, Kanban method is good in visualizing the workflow and monitoring project progress using Kanban board and CFD; however, both do not show target information, and fail to relate it to how much have been accomplished if the project is to meet its commitments. Thus, ***Agile Kanban method still has lack in visualizing sufficient information and useful indicators to monitor project progress***.

From the above discussion, results revealed that there are three significant challenges in Agile Kanban method, which are summarized in the following three points:

- Agile Kanban method has lacking of mechanism for progress tracking.
- Practitioners of Agile Kanban method have a major challenge in determining the optimum WIP limits for each stage in Kanban board.
- Kanban board does not report target information or quantitative indications about the projects' progresses.

Therefore, those challenges give a significant impact to develop a model for improving software project monitoring task of Agile Kanban method.

Criteria Affecting Software Project Monitoring Task of Agile Kanban Method

This section presents the results of criteria that affect software project monitoring task of Agile Kanban method. Results are categorized into three subsections based on the three challenges that have previously discussed.

Criteria Affecting Progress Tracking

During the development process, project data are collected and used as the foundation and measurements for progress monitoring task [30-33]. For instance, data such as start dates, completion dates, and cycle time, are assigned to each task of the project in accordance with the project schedule. Controlling cost and schedule using methods, values, or measures helps to deliver products according to its expectations [6, 24]. In this context, Zhang [34] and Li, Ma and Dong [31] claimed that EVA is the suitable method for monitoring cost and schedule. Besides, it needs identifying the variables: planned value (PV), actual cost (AC), and earned value (EV) in order to generate project status, thus current status of project is maintained in database and documented by a time and date stamp to help project manager to track and report the project progress [6, 33]. Moreover, Ong, Wang and Zainon [32] argued that calculating Estimate At Complete (EAC) is used for reporting project progress. In addition, Hazır [6] and Li, Ma and Dong [31] have claimed to prepare an accurate planning and forecast the project performance for development process of software project. Along with that, an early warning property for slight deviation in project schedule could be added in order to improve the progress monitoring task [6].

Criteria Affecting Determining WIP Limits

Al-Baik and Miller [20] have conducted a systematic literature review (SLR) to investigate the concept of limiting WIP. The results showed that majority of studies suggested that organizations set WIP limit by experiment. In this regard, some studies, such as [35-37], have emphasized to start with lenient number, and in this case, the common

situation for the limits will be wrong. Afterwards, limits need to be altered and adjusted as project progresses based on the experience of the project manager or team members. Moreover, Benson and Barry [38] stated that even though setting WIP limits is difficult in the beginning stages, after discovering that prioritizing some tasks over others ultimately leads to complete all tasks in shortest time. However, this challenge can be resolved by selecting an initial estimate on the basis of a common agreement between development teams [39]. Radigan [40] argued that teams match the amount of WIP to the team's capacity. However, determining the WIP limits depends on the team capacity and resources also, such as numbers of workers, technology settings etc., of the SDOs as stated by [15]. Setting WIP limits needs to know how many people on the team and how many tasks that team to work on at the same time [41]. Additionally, Halasz [42] claimed that the maximum number of tasks cannot be more than three tasks per person to ensure that the team is not overloaded, while the minimum number of tasks is twice the team size. Cork [43] has emphasized the use of Little's Law to determine WIP limits as suggested by Thomas [44]. It is often written in software circles as:

WIP=Throughput * Cycle Time

whereas Throughput is the number of tasks per time, and Cycle Time is the desired time for work items that would lead to successfully meeting budget and schedule goals. Little's Law can be a powerful demonstration of how reducing WIP can reduce cycle time. However, when WIP dropped below the limits, the team could continue to hit cycle times, but would fall short of the total throughput number. Therefore, when using Little's Law, it is important that the formula be adjusted periodically as WIP limits change [43, 45]. On the other hand, the commercial tools that implement Kanban method have different settings for WIP limits. For instance, Leankit Kanban tool also uses Little's Law to set WIP limits [46], while Visual Studio tool depends on the number of team members and maximum number of tasks per a member [47]. For KanbanTool, it limits WIP based on maximum tasks per a time and the number of team members [48].

Criteria Affecting Visualizing the Workflow

Typically, the basic project data are collected before and during software project implementation in order to visualize the workflow and monitor projects' progress [33]. Data are updated concurrently to present and report useful information. In Kanban method, Boeg [49] stated that workflow demonstration makes Kanban a powerful method in making informed decisions, whereby data presentation on the Kanban board can easily assists project managers and team members to make a factual-based decision. By looking at Kanban board, management can get information on resource capacity and availability that helps in resource assignment and scheduling.

Graphical approaches, such as Gantt charts, cumulative cost curves, and resource load charts, are used in project monitoring and scheduling. Zhang [34] claimed that these approaches provides only visual effects, thus it must show quantitative information in order to help the project manager for progress monitoring of software projects. Moreover, using control charts to monitor a SDPs can help practitioners to manage process performance and progress monitoring quantitatively [50]. Likewise, a Q chart can help project managers simultaneously monitor and evaluate schedule and cost performance, whereby it has early detect capability and real-time process monitoring [51].

Table 1 summarizes the results of this study by showing the current challenges along with their criteria that affect software project monitoring task of Agile Kanban method.

TABLE 1. Summary of Challenges and Criteria that affect Monitoring Task of Agile Kanban Method

Challenges	Criteria	Resources
Progress Tracking	Data collection	31 32 33
	Cost and schedule controlling	6 24 31 34
	Current status maintaining	6 33 34
	Planning and forecasting	6 31
	Schedule deviation	6 34
Determining Optimum WIP Limits	Experiment and experience	20 36
	Task prioritizing	38
	Agreement between team members	39
	Team members	40
	Team members and resources	15
	Team members and max tasks per a member	42
	Cycle time &throughput	43
	Team members and throughput	48
Visualizing Useful Insights for Workflow	Data collection	33
	Data presentation	49
	Real time updating	51
	Quantitative information displaying	34
	Progress status reporting	17
	Understanding the Visualized Elements	20

CONCLUSION AND FUTURE WORK

This paper has clarified that Agile Kanban method has problem in progress monitoring task. This problem negatively affects the scheduling of the development process, thus the failures rate of in software projects is increased due to late delivery. Thus, a narrative review has conducted through revision of the journals, proceeding papers, books, thesis, documents, blogs, and reports. However, this review has focused on identifying the current challenges along with their criteria that affect software project monitoring task of Agile Kanban method. In additions, this review has limited to researches published in the last 10 years (2007–2017).

The results revealed that Agile Kanban method has three key challenges, which are lacks of mechanism for progress tracking, lacks of adequate technique to determine WIP limits, and lacks of visualizing sufficient information and useful indicators to monitor project progress. Therefore, those challenges give a significant impact to improve Agile Kanban method in terms of software project monitoring task. Besides the challenges, this paper has identified various criteria that affect the progress monitoring task of Agile Kanban method.

In the future, the findings of this study could be used to develop an improved software project monitoring task model of Agile Kanban method. As a suggestion, the improved model might be consisted of three main components, which are (1) progress tracking, (2) optimum WIP limits, and (3) useful insights for workflow. Along with that, the criteria affecting each component would be involved within developing the proposed model. After the development stage, the model will be evaluated through two different stages, which are verification and validation. Firstly, the model will be verified based on the comprehensiveness, understandability, feasibility, and organization by the knowledge and domain experts. Secondly, the model will be validated by conducting case studies in order to prove its applicability and feasibility.

ACKNOWLEDGEMENT

The authors wish to thank the Universiti Utara Malaysia for funding this study under University Research Grant Scheme, S/O project code: 13853.

REFERENCES

1. B. B. Agarwal and S. D. S. P. Tayal, *Software Project Management*. (Laxmi Publications, 2011).
2. G. Anand, A. Chandrashekar and G. Narayanamurthy, Journal of Enterprise Transformation **4** (2), 123-150 (2014).
3. R. Skinner, L. Land, W. Chin and R. R. Nelson, in *International Research Workshop on IT Project Management (IRWITPM)* (2015).
4. A. Kumar, D. S. Salaria and R. Gangwar, Global Journal of Computer Science and Technology **14** (4-C), 21 (2014).
5. D. Mihai Liviu, Database Systems Journal **5** (3), 37-56 (2014).
6. Ö. Hazır, International Journal of Project Management **33** (4), 808-815 (2015).
7. M. Doraisamy, S. b. Ibrahim and M. N. r. Mahrin, in *Confernece on Open Systems (ICOS)* (IEEE, Bandar Melaka, Malaysia, 2015), pp. 12-17.
8. M. A. Ahmad, N. H. Ubaidullah and M. Lakulu, World of Computer Science & Information Technology Journal **4** (5) (2014).
9. D. S. Nguyen, American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS) **17** (1), 323-391 (2016).
10. H. K. Flora and S. V. Chande, International Journal of Computer Science and Information Technologies **5** (3), 3626-3637 (2014).
11. H. Lei, F. Ganjeizadeh, P. K. Jayachandran and P. Ozcan, Robotics and Computer-Integrated Manufacturing **43**, 59-67 (2017).
12. V. Mahnic, in *Recent Advances in Computer Engineering, Communications and Information Technology* (WSEAS Press Espanha, 2014), pp. 281-288.
13. N. Kirovska and S. Koceski, Journal of Applied Economics and Business **3** (3), 25-34 (2015).
14. H. Al-Aidaros and M. Omar, Journal of Engineering and Applied Sciences **12** (15), 3851-3857 (2017).
15. D. J. Anderson, *Kanban*. (Blue Hole Press, 2010).
16. M. Ahmad, J. Markkula, M. Oivo and B. Adeyemi, in *10th International Conference on Software Engineering Advances ICSEA* (IEEE, Barcelona, Spain, 2015).
17. E. Corona and F. E. Pani, WSEAS Transactions on Information Science and Applications **10** (1), 1-13 (2013).
18. D. Dennehy and K. Conboy, Journal of Systems and Software **133**, 160-173 (2017).
19. M. Ahmad, J. Markkula and M. Oivo, in *nternational Conference on Agile Software Development* (Springer International Publishing, Cham 2016), pp. 156-168.
20. O. Al-Baik and J. Miller, Empirical Software Engineering **20** (6), 1861-1897 (2015).
21. P. Cronin, F. Ryan and M. Coughlan, British journal of nursing **17** (1), 38-43 (2008).
22. L. Uman, Journal of the Canadian Academy of Child and Adolescent Psychiatry **9** (1), 57–59 (2011).
23. M. Lindblom, M.Sc. thesis, KTH Royal Institute of Technology, 2015.
24. H. Raju and Y. Krishnegowda, International Journal on Recent Trends in Engineering & Technology **11** (1), 24 (2014).
25. K. Power and K. Conboy, in *Proceedings of the Second International Workshop on Software Architecture and Metrics* (IEEE Florence, Italy, 2015), pp. 15-21.
26. K. Petersen, P. Roos, S. Nyström and P. Runeson, Journal of Software: Evolution and Process **26** (12), 1150-1171 (2014).
27. M. Hammarberg and J. Sunden, *Kanban in action*. (Manning Publications Co., 2014).
28. K. Karunanithi, (California State University, Fullerton, 2016), pp. 1-19.
29. P. Klipp, *Getting started with Kanban*. (kanbanery, 2014).
30. L. Kazi, B. Radulovic and Z. Kazi, in *2012 IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics (SISY)* (IEEE, Subotica, Serbia, 2012), pp. 19-25.
31. J. Li, Z. Ma and H. Dong, in *Seventh IEEE/ACIS International Conference on Computer and Information Science ICIS 08* (IEEE, Portland, 2008), pp. 475-480.
32. H. Y. Ong, C. Wang and N. Zainon, Engineering Management Journal **28** (1), 39-53 (2016).
33. K. Petersen and C. Wohlin, Software: Practice and experience **41** (9), 975-996 (2011).
34. J. Zhang, in *2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)* (IEEE, China, 2011), pp. 1495-1498.

35. M. Ikonen, E. Pirinen, F. Fagerholm, P. Kettunen and P. Abrahamsson, in *2011 16th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)* (IEEE, Las Vegas, USA, 2011), pp. 305-314.
36. N. Oza, F. Fagerholm and J. Münch, in *6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)* (IEEE, San Francisco, CA, USA, 2013), pp. 125-128.
37. J. Reveco, M. Mora, T.-C. Shen, R. Soto, J. Sepulveda and J. Ibsen, in *Proc. SPIE 9152, Software and Cyberinfrastructure for Astronomy III* (International Society for Optics and Photonics, Montréal, Quebec, Canada, 2014), Vol. 9152, pp. 1-10.
38. J. Benson and T. D. Barry, *Personal Kanban: Mapping Work, Navigating Life*. (Modus Cooperandi Press, 2011).
39. N. Tripathi, P. Rodríguez, M. O. Ahmad and M. Oivo, in *Agile Processes in Software Engineering and Extreme Programming* (Springer, 2015), pp. 178-190.
40. D. Radigan. *How the kanban methodology applies to software development,* https://www.atlassian.com/agile/kanban, (2016).
41. Smart Sheet, Common *Questions About Kanban*, https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban, (2016).
42. T. Halasz. *How to set WIP limits in Kanban,* https://www.linkedin.com/pulse/power-work-in-progress-limits-kanban-boards-tibor-halasz, (2016).
43. P. Cork, M.Sc. thesis, Tampere University of Applied Sciences, 2015.
44. S. Thomas. *Little's Law the basis of Lean and Kanban,* http://itsadeliverything.com/littles-law-the-basis-of-Lean-and-kanban, (2015).
45. C. M. Shinkle, in *Agile Conference, AGILE '09* (IEEE, USA, 2009), pp. 186-191.
46. Leankit Inc. 7 *Lean Metrics to Improve Flow*, https://leankit.com/learn/kanban/lean-flow-metrics/, (2016).
47. Visual Studio. *Determine WIP limits*, https://www.visualstudio.com/en-us/docs/work/kanban/wip-limits, (2016).
48. Kanban Tool, *Setting WIP limits*, http://kanbantool.com/kanban-wip-limits, (2016).
49. J. Boeg, *Priming Kanban: A 10 step guide to optimizing flow in your software delivery system*, Second ed. (Copenhagen: Trifork, Chronografisk, Denmark, 2012).
50. M. T. Baldassarre, D. Caivano, B. Kitchenham and G. Visaggio, in *Proceedings of the 11th international conference on Evaluation and Assessment in Software Engineering* (British Computer Society, UK 2007), pp. 94-102.
51. C.-W. Chang and L.-I. Tong, Software Quality Journal **21** (3), 479-499 (2013).