Towards an Improved Software Project Monitoring Task Model of Agile Kanban Method

Hamzah Alaidaros[#1], Mazni Omar[#2], Rohaida Romli[#3]

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

[1]m7amza7@yahoo.com
[2]mazni@uum.edu.my
[3]aida@uum.edu.my

***Abstract*— Agile Kanban method recently is gaining increasing attention and popularity in software development organizations (SDOs). This method has numerous advantages that make it performs better than other Agile methods in terms of managing software projects. However, different studies revealed that this method has significant challenges that negatively impact the scheduling of the development process. Therefore, late delivery of software projects may occur, thus the rate of projects' failures will be increased. In response, this paper aims to explicate the current challenges in progress monitoring task of Agile Kanban method. Accordingly, the results gave insights to bridge that gap by developing an improved software project monitoring task model of Agile Kanban method. To do so, we identified the components and criteria that affect software project monitoring task, and then an initial model has proposed. The initial model consists of three main components, which are (1) extending progress tracking, (2) generating optimum WIP limits, and (3) visualizing useful insights for workflow. Further research can be focused on developing and evaluating the proposed model through discussion with the knowledge and domain experts.**

***Keywords*—** *Software project management, Agile development method, Kanban method, Progress monitoring task.*

1. Introduction

Software project management (SPM) is the concept that involves knowledge, techniques, and tools, which are essential needed for managing the development process of software projects. SPM is a sub-discipline of project management in which software projects are planned, implemented, monitored, and controlled. In addition, in order to deliver the software on the time, a software project manager is responsible for monitoring the process of software development projects (SDPs) [1].

In SPM, monitoring the projects' progress is an essential task during the execution of any project. Besides being required to steer the project, timely and accurate reporting is important to keep the team and management up to date on the project's progress [2]. Progress monitoring task is carried out to ensure that projects' plan is progressed according to budget, schedule, and quality expectations. Thus, successful implementation of software projects depends entirely on successful monitoring mechanisms, while the lack of monitoring SDPs leads to the failure of such projects [3], [4]. Recently, software development organizations (SDOs) still have challenges in delivering their software projects according to their specifications, time, and budget [5]. The Standish Group Chaos report on software projects showed that failed and challenged projects represented approximately two-thirds of all project outcomes, whilst only about a third of the software projects were successful [6].

During the last two decades, Agile methods are being massively adopted for developing software projects due to their flexibility and effectiveness. Agile methods refer to the family of lightweight software development (SD) methods that define a process of iteration, where design, construction, and deployment of different pieces can occur simultaneously. Furthermore, it can provide a shorter development cycle, higher customer satisfaction, and rapid changes to the business requirements in the SD environments [7]. Recent survey [8] have conducted among IT professionals, and revealed that 60% are using Agile methods, while 38% are using a mix of Agile and another methods. Moreover, the tenth annual state of Agile survey has reported that 95% of respondents' organizations practice Agile methods [9].

Particularly, Scrum and Kanban are considered as the two powerful Agile methods that focus on managing software projects. This is because both can optimize the development process by identifying the tasks, managing time more effectively, and setting-up teams [10].

International Journal of Supply Chain Management
IJSCM, ISSN: 2050-7399 (Online), 2051-3771 (Print)

According to [9], Scrum is the most followed method and 58% of the respondents practice it among other Agile methods, while more than 39% of the respondents practice Kanban method within their organizations.

Despite of that, various studies reported that Kanban method, currently, is the contender among Agile methods because it has numerous advantages that make it performs better than Scrum and other Agile methods in terms of having experience greater consistency in managing software engineering (SE) projects [7], [10], [11]. However, Agile Kanban method has significant lacking in progress monitoring task during development process of software projects. This problem negatively affects the success of software projects because of lags in projects' scheduling that lead to late delivering [5], [12]-[14]. Therefore, this study aims to investigate the lacking in progress monitoring task by developing an improved model of Agile Kanban method to remedy that situation.

The rest of the paper is organized as follows. Section 2 introduces concepts of Agile Kanban method. Next, the current challenges of Agile Kanban method are provided in section 3. Section 4 discusses the criteria that affect software project monitoring task of Agile Kanban method. In section 5, the initial model is proposed and illustrated. Last section concludes this study and suggests some remarks for the future work.

2. Agile Kanban Method

Anderson [15], father of Agile Kanban method in SD, has defined five principles 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 is vital method for managing workflow and controlling waste. It does not push the work tasks to members, but it utilizes the pulling system. All team members must have only one task to work on at a specific time. After finishing that task, team member can pull another task [16].

Besides, 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 [7], [17]. However, these two core principles still having challenges and limitations that impact the monitoring task of Agile Kanban method. Kanban board is used to visualize the workflow and monitor the project progress by showing the activities of the development process and keeping WIP in control [15], [18]. Moreover, it allows developers to focus on a few tasks, and reduce the waste of time and resources because of switching from one item to another. A Kanban board is divided into several stages in a vertical direction. Each stage indicates the state of the task. The tasks are represented by cards and attached on the board, thus to represent the current state of the tasks. Along with the changes in the state of the task, the task card is moved on the Kanban board from left to right [19]-[22]. Figure 1 shows Agile Kanban board.

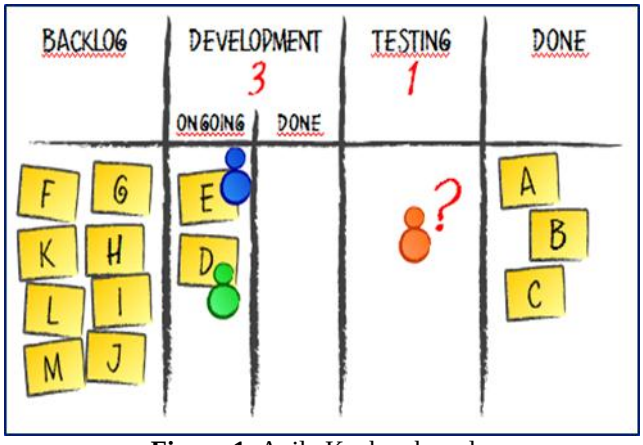

Figure 1. Agile Kanban board

3. Challenges of Agile Kanban Method

Although Agile Kanban method is gaining increasing attention and popularity in SDOs, this method still has significant challenges in its progress monitoring task during the development of software projects. This section presents the current challenges of Agile Kanban method.

3.1 Progress Tracking

In fact, Kanban is a lightweight method, and does not has a clear formal definition in order to promote changes and modifications as it has been acknowledged in [15]. In this context, [7] stated that applying Kanban method to SD, either as a standalone method or in combination with other methods, has been a highly pertinent topic for software researchers and practitioners. As such, [12] claimed that Kanban method needs another supporting method to work effectively. Likewise, [23] argued that Kanban method should be complemented or expanded by Agile methods or another methods in SDOs to keep schedule of the project progresses as it is planned. In the same vein, recent study [24] suggested to integrate Kanban method with earned value analysis (EVA) method in order to make the most of both.

Accordingly, this issue has led to integrate Scrum with Kanban to introduce a new method called Scrumban [25], and to integrate Kanban with value stream mapping (VSM) [26]. Even though previous studies addressed this issue, these are few studies and have different limitations. For instance, Scrumban method still facing challenges with progress tracking and managing WIP [27]. However, the integration of Kanban with VSM was not to improve Kanban method, instead to improve some areas that

constrain and harm workflow in the value stream, whereby VSM still lacking to capture the dynamic nature of the software process to evaluate improvements [22].

To sum up, Agile Kanban method has lacking in progress tracking mechanism, thus this challenge give a motivation to integrate Kanban method with another effective method. Therefore, integrating and applying a suitable monitoring method with Kanban method may contribute to improve monitoring task of Agile Kanban method with undertaking of the aforementioned studies limitations.

3.2 Determining WIP Limits

Limiting WIP is a core principle of Kanban method, which is defined as the maximum number of tasks for each stage on the Kanban board. This number is identified by project manager in order to prevent road blocks and make tasks flow faster through the board. By limiting WIP, the development process is kept under monitoring using a card system in Kanban board. The cards enable team members to monitor WIP and to self-organize by assigning their own tasks [15], [18]-[21].

In spite of aforementioned benefits, determining the WIP limits is proved as a major challenge faces software project practitioners, whereby no formula for find out the optimum WIP limits for each stage. The optimum WIP limits refer to suitable numbers for each stage in Kanban board that can monitor and control team members with their tasks, and thus ensuring that project is progressed as it is planned. Typically, in order to set WIP limits, it needs to start with initial number, and after some of time, this initial number needs to be adjusted as the project is being progressed. However, a bad estimation for initial WIP limits can be painful, and will impact the project progress. Thus, it can cause throughput decrease and lead time to increase and versa vice. Consequently, this situation will lead to lags in scheduling of the development process of software projects and failing to deliver software products on the prescribed time [11]-[14].

Hence, even though Agile Kanban method is good in monitoring project progress by using limit WIP principle, it is still a challenge and difficult to determine the optimum WIP limits for each workflow stage in Kanban board. Therefore, there is a need to generate the optimum numbers of WIP limits.

3.3 Visualizing Workflow

Visualizing the workflow is also another core principle of Kanban method, which is defined as the process of highlighting the mechanisms, interactions, queues, waiting, and delays that are involved in the implementing of a part of valuable software. A Kanban board is used to visualize the workflow and monitor the project progress by showing the activities of the development process [15], [18]. Besides that, data such as lead time, cycle time, number of bugs, throughput, and so on, are usually shown in diagrams, affixed to the walls of the workplace, or in any case continuously updated and made public [17], [28].

Within Kanban method, the cumulative flow diagram (CFD) has used to show WIP and average lead time, and to highlight issues and bottlenecks [17]. The CFD is useful for thinking of workflow states as queues, understanding the queues behavior, and diagnosing problems and taking meaningful decisions [29], [30]. Nevertheless, CFD is cited in almost publications about the Kanban method, it seems to be provided only by very few commercial tools. For instance, Kanbanery tool uses CFD just to report some information, such as average lead time and cycle time by using date filtering [17], [31]. Moreover, Kanban board and CFD neither report how much of work is left nor provide some indications of where the project ought to be or have it progressed at a constant rate [11].

Generally, in spite of Agile Kanban method is good in visualizing workflow and monitoring projects' progress by using Kanban board, however, it does not show target information, and fail to relate it to how much should have been accomplished if the project is to meet its commitments. Thus, there is a need to identify alternative and extra visualization criteria for Kanban method that may provide useful insights and information for helping project managers to take meaningful decisions regarding to the projects' progress.

Overall, it can be concluded that Agile Kanban method has lacking in progress monitoring task during development process of software projects. Thus, this method needs to be improved by integrating it with another method to be an effective method, whereby it is a lightweight method. In addition, determining the optimum WIP limits for each stage in Kanban board is proved as a major challenge faces software project practitioners. Consequently, assigning incorrect numbers for WIP limits causes lags in project scheduling in turn to lead to late delivery and software project failures. Furthermore, Kanban board neither reports target information or quantitative calculations about how much of work progress is left nor provides some indications of where the project is being progressed, which could be useful for progress monitoring task. This gap gives significant insight to develop a model for improving software project monitoring task of Agile Kanban method. Therefore, the criteria that affect software project monitoring task are presented in the next section.

4. Criteria Affecting Software Project Monitoring Task of Agile Kanban Method

In order to address the previous problem, this section presents the criteria that affect software project monitoring task of Agile Kanban method. These criteria have carried out and categorized into three subsections based on the three challenges that have previously discussed.

4.1 Criteria Affecting Progress Tracking

During the development process, project data are collected and used as the foundation and measurements for progress monitoring task. 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 [32]-[35]. Controlling cost and schedule using methods, values, or measures helps to deliver products according to its expectations [2], [26] . In this context, [36] and [33] 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 [2], [35]. Moreover, [34] argued that calculating Estimate At Complete (EAC) is used for reporting project progress. In addition, [2] and [33] 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 [2]. Table 1 shows the criteria with their descriptions that affect progress tracking during the development of software projects.

Table 1. Criteria that Affect Progress Tracking

Criteria	Description	Resources
Data collection	Basic data are collected and used as the foundation for progress monitoring before and during software project implementation.	[33] [34] [35]
Cost and schedule controlling	Controlling cost and schedule using methods, values, or measures helps to deliver products according to its expectations.	[2] [26] [33] [36]
Current status Maintaining	Maintaining the current status of project helps to estimate the expected project time and cost.	[2] [35] [36]
Planning and forecasting	Preparing an accurate planning and forecasting the project performance.	[2] [33]
Schedule deviation	An early warning system for slight deviation in the project schedule.	[2] [36]

4.2 Criteria Affecting Determining WIP Limits

A systematic literature review (SLR) has been conducted by [13] 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 [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. In this direction, [38] stated that setting WIP limits is difficult in the beginning stages, whilst after discovering prioritizing of 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]. Further, [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 in [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]. As such, [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. [43] has emphasized the use of Little's Law to determine WIP limits as suggested by [44], whereby this law 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].

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 [45], while Visual Studio tool depends on the number of team members and maximum number of tasks per a member [46]. For KanbanTool, it limits WIP based on maximum tasks per a time and the number of team members [47].

Table 2 shows the criteria with their descriptions that affect the determining WIP limits during the development of software projects.

Table 2. Criteria that Affect Determining WIP Limits

Criteria	Description	Resources
Experiment and experience	Project manager starts with lenient number, and then limits need to be adjusted as project progresses.	[13] [37]
Task prioritizing	The prioritization of some tasks over others ultimately leads to complete all tasks in shortest time.	[38]
Agreement between team members	Selecting an initial estimate based on a common agreement between development team members.	[39]
Team members	The number of team members.	[40]
Team members and resources	The number of team members, and the resources of the SDOs.	[15]
Team members and max tasks per a member	The number of team members and the max number of tasks per a member.	[42]
Cycle time and throughput	Cycle Time is the desired time for work items, while the throughput is the number of tasks per time.	[43]
Team members and throughput	The number of team members and the number of tasks per time	[47]

4.2 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 [35]. Data are updated concurrently to present and report useful information. In Kanban method, [48] 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. In this vein, [36] 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 [49]. 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 [50].

Table 3 shows visualization criteria with their descriptions, which are essentially required for progress monitoring task during SDPs.

Table 3. Criteria that Affect Visualizing the Workflow

Criteria	Description	Resources
Data collection	Basic data are collected before and during software project implementation in order to visualize the workflow and monitor projects' progress.	[35]
Data presentation	Data presentation on the Kanban board can easily assists project managers and team members to make a factual-based decision.	[48]
Real time updating	Updating project data in real time can help project managers simultaneously monitor and evaluate schedule and cost performance.	[50]
Quantitative information displaying	using charts to show quantitative information can help practitioners to manage process performance and progress monitoring quantitatively	[36]
Progress status reporting	Visualizing indications and reports on the progress, where the project ought to be, or have it progressed are essential for progress monitoring.	[17]
Understand-ing the visualized elements	Understanding the Kanban board and its different visualized elements to ensure that used effectively	[13]

5. The Initial Model

After identifying the criteria that affect software project monitoring task of Agile Kanban method, this section introduces the initial model for improving software project progress monitoring task of Agile Kanban method. The initial model was build based on the original Kanban model that is usually represented by Kanban board, which is shown in Figure 1. Considering the current situation, Agile Kanban method needs to improve three components that play roles in the task of progress monitoring. The initial model consists of three main components, which are (1) extending progress tracking, (2) generating optimum WIP limits, and (3) visualizing useful insights for workflow. Besides that, the criteria that influence each component must be involved during the development the proposed model. Figure 2 shows the initial model for an improved software project progress monitoring task of Agile Kanban method.

The first component is to extend the progress tracking of Agile Kanban method. This extension may apply EVA method, whereby this method can improve the progress tracking process of Agile Kanban method as suggested in [24]. EVA is an efficient method used for tracking the projects' progress and estimate the expected project time and cost using the project's current status. The equations of EVA method are used to keep tracking of the project schedule progresses as it is planned. These equations need three basic data, which are PV, AC, and EV. These data and results of the calculations will be stored in data store, and then will be visualized in Kanban board in order to support the tracking process of the projects' progress.

The second component is to generate the optimum WIP limits. This component can use a formula that dynamically generates the optimum WIP limits for each stage in Kanban board based on the need of projects' progress. This formula will be constructed by using algorithm. The basic project data, such as the number of tasks, number of team members, throughput, lead time, cycle time, and so on, will be used as key inputs for the algorithm. Different mathematical calculations, based on the aforementioned data, can be performed to generate the optimum WIP limits for each workflow stage in Kanban board. The output will be different WIP limits that will be assigned to predefined stages and will be stored in database store. Indeed, these limits will be dynamically changed based on the updates on the project status and releasing of tasks.

The third component is to visualize useful insights for workflow. In addition to the general Kanban board, this component will use project data along with the results of EVA calculations in order to show useful insights and quantitative information. Thus, project manager and team members can monitor the projects' progress easily, whereby information such as where the project ought to be, or have it progressed will be provided. Besides that, real time updates for indicators and reports on the projects' progress will be visualized.

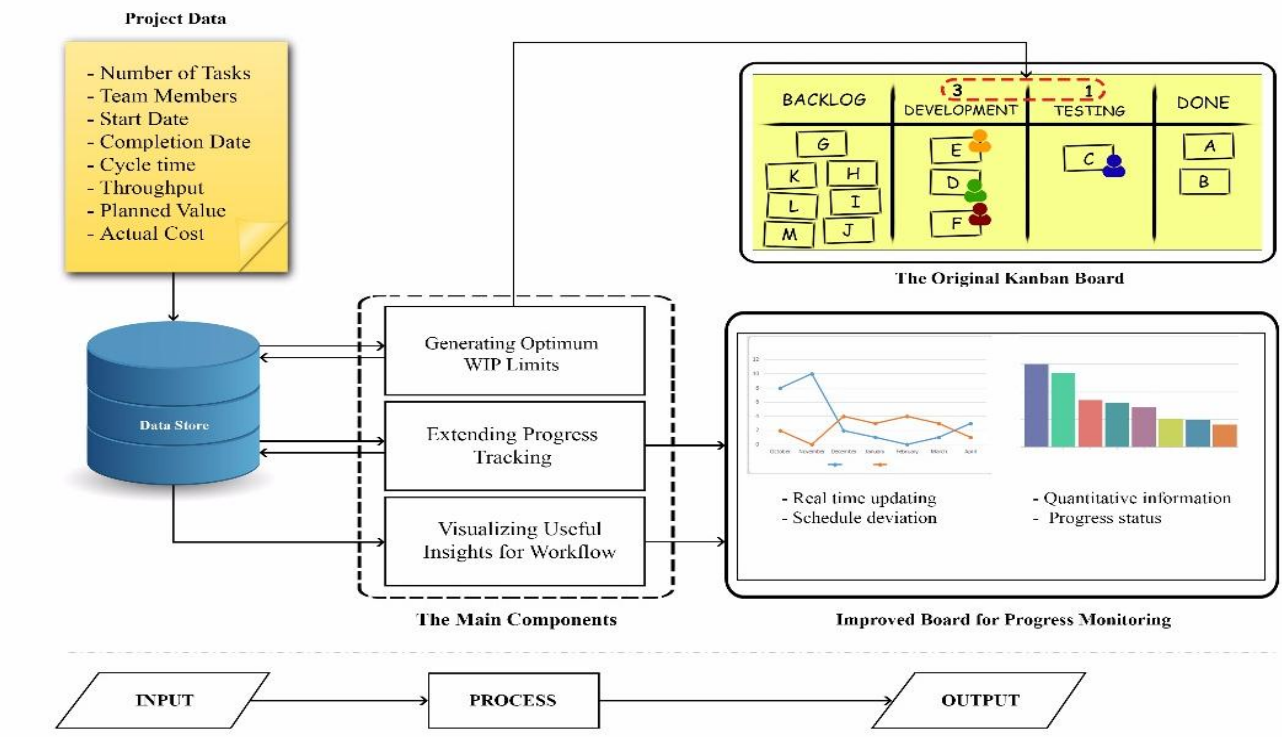

Figure 1. The initial model

6. Conclusion And Future work

Despite of that Agile Kanban method is gaining increasing attention and popularity in SDOs, this paper has emphasized that this method still has significant challenges related to its progress monitoring task. It clarified that one of the current challenges Agile Kanban method is the lacking of progress tracking mechanism, thus this method needs to be integrated with another method to be an effective method. Additionally, determining the optimum WIP limits for each stage in Kanban board is proved as a major challenge faces software project practitioners. Consequently, assigning incorrect numbers for WIP limits causes lags in project scheduling in turn to lead to late delivery and software project failures. Moreover, Kanban board neither reports target information or quantitative calculations about how much of work progress is left nor provides some indications of where the project is being progressed, which could be useful for progress monitoring task. Accordingly, these challenges have paid attention to develop an improved software project monitoring task model of Agile Kanban method. However, this paper has not limited to determine the components and criteria affect the progress monitoring task, but only to propose an initial model.

Therefore, future research could be directed to develop a complete model, by explicating the detailed process of each component and how it works with other components. Besides that, the developed model needs to verify its understandability, relevance, feasibility, organization, and comprehensiveness through knowledge and domain experts. Thereafter, the validity of that model will have to be tested via real projects data using case studies and focus group.

Acknowledgment

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] L. D. Popescu, "The Project Team Management", International Scientific Conference" Strategies XXI", Bucharest, Vol. 3, pp. 234-238, ,2016.

[2] Ö. Hazır, "*A review of analytical models, approaches and decision support tools in project monitoring and control*", International Journal of Project Management, Vol. 33, pp. 808-815, 2015.

[3] M. Doraisamy, S. b. Ibrahim, and M. N. r. Mahrin, "*Metric based Software Project Performance Monitoring Model*", the Conference on Open Systems (ICOS), IEEE, Bandar Melaka, Malaysia, pp. 12-17, 2015.

[4] D. S. Nguyen, "*Workplace Factors that Shape Agile Software Development Team Project Success*", American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS), Vol. 17, pp. 323-391, 2016.

[5] R. Skinner, L. Land, W. Chin, and R. R. Nelson, "*Reviewing the Past for a Better Future: Re-evaluating the IT Project Retrospective*", the International Research Workshop on IT Project Management (IRWITPM), Texas, pp. 110-119, 2015.

[6] Chaos Report Chaos report, http://www.standishgroup.com/outline/, Last access 29-04-2018.

[7] H. K. Flora and S. V. Chande, "*A Systematic Study on Agile Software Development Methodologies and Practices*", International Journal of Computer Science and Information Technologies, Vol. 5, pp. 3626-3637, 2014.

[8] Interop ITX Research Report, http://www.interop.com/, Last access 07-03-2018

[9] Version One, Annual State of Agile Development Survey, https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf, Last access 07-03-2018.

[10] 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.

[11] K. Karunanithi, "*Metrics in Agile and Kanban Software Measurement Techniques*", Fullerton: California State University, 2016.

[12] M. Ahmad, J. Markkula, and M. Oivo, "*Insights into the Perceived Benefits of Kanban in Software Companies: Practitioners' Views*", the international Conference on Agile Software Development, Cham, pp. 156-168, 2016.

[13] O. Al-Baik and J. Miller, "*The Kanban approach, between agility and leanness: a systematic review*", Empirical Software Engineering, Vol. 20, pp. 1861-1897, 2015.

[14] N. Kirovska and S. Koceski, "*Usage of Kanban methodology at software development teams*", Journal of Applied Economics and Business, Vol. 3, pp. 25-34, 2015.

[15] D. J. Anderson, *Kanban*, Blue Hole Press, 2010.

[16] G. Concas, M. I. Lunesu, M. Marchesi, and H. Zhang, "*Simulation of software maintenance process, with and without a work-in-process limit*", Journal of Software: Evolution and Process, Vol. 25, pp. 1225-1248, 2013.

[17] E. Corona and F. E. Pani, "*A review of lean-Kanban approaches in the software development*", WSEAS Transactions on Information Science and Applications, Vol. 10, pp. 1-13, 2013.

[18] D. Dennehy and K. Conboy, "*Going with the flow: An activity theory analysis of flow techniques in software development*", Journal of Systems and Software, Vol. 133, pp. 160-173, 2017.

[19] M. Hammarberg and J. Sunden, *Kanban in action*, Manning Publications Co., 2014.

[20] V. R. Nanduri, "*Lean+ Agile vs Seven Wastes in Software Development*", M.Sc. thesis, 2014.
[21] S. Nakazawa and T. Tanaka, "*Development and Application of Kanban Tool Visualizing the Work in Progress*", the 5th International Congress on Advanced Applied Informatics (IIAI-AAI), IEEE, pp. 908-913, 2016.
[22] N. B. Ali, K. Petersen, and B. B. N. de França, "*Evaluation of simulation-assisted value stream mapping for software product development: Two industrial cases*", Information and Software Technology, Vol. 68, pp. 45-61, 2015.
[23] M. Lindblom, "*Agile Development in a Lonely Environment: How to Develop Software Using Agile Techniques within Small-Scale Projects*", M.Sc. thesis, KTH Royal Institute of Technology, 2015.
[24] H. Alaidaros and M. Omar, "*Software Project Management Approaches for Monitoring Work-In-Progress: A Review*", Journal of Engineering and Applied Sciences, Vol. 12, pp. 3851-3857, 2017.
[25] V. Mahnic, "*Improving Software Development through Combination of Scrum and Kanban*", the Recent Advances in Computer Engineering, Communications and Information Technology, Espanha, pp. 281-288, 2014.
[26] H. Raju and Y. Krishnegowda, "*Value Stream Mapping and Pull System for Improving Productivity and Quality in Software Development Projects*", International Journal on Recent Trends in Engineering & Technology, Vol. 11, pp. 24-38, 2014.
[27] M. Leskinen, "*Towards a flow efficient ICT development process with Kanban: A Case Study*", M.Sc. thesis, Aalto University, 2015.
[28] V. T. Heikkilä, M. Paasivaara, and C. Lassenius, "Teaching university students Kanban with a collaborative board game," in *Proceedings of the 38th International Conference on Software Engineering Companion*, 2016, pp. 471-480.
[29] K. Power and K. Conboy, "*A metric-based approach to managing architecture-related impediments in product development flow: an industry case study from Cisco*", the Proceedings of the Second International Workshop on Software Architecture and Metrics, IEEE Press, Italy, PP. 15-21, 2015.
[30] D. Reinertsen, "*The principles of product development flow: second generation lean product development*". Celeritas, Redondo Beach, 2009.
[31] Kanbanery, https://kanbanery.com/, Last access 26-02-2018.
[32] L. Kazi, B. Radulovic, and Z. Kazi, "*Performance indicators in software project monitoring: Balanced scorecard approach*", 10th Jubilee International Symposium on Intelligent Systems and Informatics (SISY), IEEE, Subotica, Serbia, pp. 19-25, 2012.
[33] J. Li, Z. Ma, and H. Dong, "*Monitoring software projects with earned value analysis and use case point*", the Seventh IEEE/ACIS International Conference on Computer and Information Science ICIS 08, IEEE, Portland, pp. 475-480, 2008.
[34] H. Y. Ong, C. Wang, and N. Zainon, "*Integrated Earned Value Gantt Chart (EV-Gantt) Tool for Project Portfolio Planning and Monitoring Optimization*", Engineering Management Journal, Vol. 28, pp. 39-53, 2016.
[35] K. Petersen and C. Wohlin, "*Measuring the flow in lean software development*", Software: Practice and experience, Vol. 41, pp. 975-996, 2011.
[36] J. Zhang, "*Schedule management method study of middle and small software projects*", the 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), IEEE, China, pp. 1495-1498, 2011.
[37] N. Oza, F. Fagerholm, and J. Münch, "*How does Kanban impact communication and collaboration in software engineering teams?*", the 6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), IEEE, San Francisco, CA, USA, pp. 125-128, 2013.
[38] J. Benson and T. D. Barry, "*Personal Kanban: Mapping Work*", Navigating Life Vol. 218: Modus Cooperandi Press, 2011.
[39] N. Tripathi, P. Rodríguez, M. O. Ahmad, and M. Oivo, "*Scaling Kanban for Software Development in a Multisite Organization: Challenges and Potential Solutions*", Agile Processes in Software Engineering and Extreme Programming, Springer, pp. 178-190, 2015.
[40] Atlassian, https://www.atlassian.com/agile/kanban, Last access 29-01-2018.
[41] Smart Sheet, https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban, Last access 28-01-2018.
[42] How to set WIP limits in Kanban, https://www.linkedin.com/pulse/power-work-in-progress-limits-kanban-boards-tibor-halasz, Last access 09-02-2018.
[43] P. Cork, "*Empirical study of project management practices*", M.Sc. thesis, Tampere University of Applied Sciences, 2015.
[44] Little's Law - the basis of Lean and Kanban, http://itsadeliverything.com/littles-law-the-basis-of-Lean-and-Kanban, Last access 26-02-2018.
[45] Leankit Inc, https://leankit.com/learn/kanban/lean-flow-metrics/, Last access 30-03-2018.
[46] Visual Studio, https://www.visualstudio.com/en-us/docs/work/kanban/wip-limits, Last access 30-03-2018.
[47] Kanban Tool, http://kanbantool.com/kanban-wip-limits, Last access 31-03-2018.
[48] J. Boeg, "*Priming Kanban: A 10 step guide to optimizing flow in your software delivery system*", Copenhagen: Trifork, Denmark, 2012.
[49] M. T. Baldassarre, D. Caivano, B. Kitchenham, and G. Visaggio, "*Systematic review of statistical process control: An experience report*", the 11th International Conference on Evaluation and Assessment in Software Engineering, BCS, Keele UK, pp. 119-129, 2007.
[50] C.-W. Chang and L.-I. Tong, "*Monitoring the software development process using a short-run control chart*", Software Quality Journal, Vol. 21, pp. 479-499, 2013