Journal of Engineering and Applied Sciences 12 (15): 3851-3857, 2017
ISSN: 1816-949X

Software Project Management Approaches for Monitoring Work-In-Progress: A Review

Hamzah Ali Alawi Al-Aidaros and Mazni Omar
School of Computing, Universiti Utara Malaysia (UUM), 06010 Sintok, Kedah, Malaysia

Abstract: Several approaches have been established for the monitoring of the Software Development (SD) process. Monitoring a Software Development Project (SDP) is essential to ensure that the project progress is according to budget, schedule and quality expectations. Consequently, the lack of monitoring and control in SD leads to the failure of such projects. Currently, Agile Methods (AMs) have received wide recognition within the SD field due to their flexibility and effectiveness. Consequently, they have been extensively adopted and thus new versions of AMs have been developed, enhanced and integrated with other approaches to support the increased demand of various project environments. This study aims to review different approaches for monitoring Work in Progress (WIP) in different software project development. Research results have indicated that a strong project management approach is needed to adequately monitor the SD process because monitoring is vital to the success of such projects.

Key words: Software projects management, software development, agile methods, work-in-progress, established

INTRODUCTION

Software Project Management (SPM) encompasses knowledge, techniques and tools necessary to manage the development of software products. SPM is defined as the art and science of planning and leading software projects. It is a sub-discipline of project management in which software projects are planned, implemented, monitored and controlled (Tarom, 2014).

In addition, in order to deliver the software on the time, a software project manager is responsible for planning and scheduling of the Software Development Project (SDP) process (Kaur *et al*., 2013). Currently, Software Development Organizations (SDOs) attempt to deliver their software products quickly within the prescribed time frame and with the highest quality and lowest cost (Anand *et al*., 2014). However, delivering software projects according to specification on time within budget has proven to be a big challenge for them (Alette and Fritzon, 2012).

The standish group report on software projects showed that only 39% of projects are completed on time and on budget. The 43% of the projects are considered challenged that is they were completed but they were over-budget or late and they offered fewer features than were originally specified. However, 18% of the projects were cancelled at some point during the development cycle (TSG, 2013). Moreover, the results of a survey from 166 Information Technology (IT) leaders indicated that 89% of projects do not regularly meet their budget and 59% of projects are typically delivered late. Furthermore, 33% of IT leaders stated that at least 25% of their budget is spent addressing rework (Kasunic and Nichols, 2014).

Monitoring the development process of software project is essential to ensure that the project progresses according to budget, schedule and quality expectations. Doraisamy *et al.* (2014) stated that successful implementation of software projects depends entirely on successful monitoring mechanisms while the lack of monitoring SDP leads to the failure of such projects. Rupinder and Jyotsna (2013) have discussed the current process approaches and analysed on failure of SD. They proved that there is a need to develop a new approach to resolve the major issues of SDP.

During past decades, several approaches have been established for monitoring SDP process (Abrahamsson and Warsta, 2003). Currently, Agile Methods (AMs) have received wide recognition within the SD field due to their flexibility and effectiveness (Flora and Chande, 2014). In addition, AMs are extensively being adopted outside of their initial intended scope of small and collocated project teams. They have been implemented in all project sizes and in both distributed and non-distributed project environments as well as in different project domains such as engineering, banking, medical and manufacturing. Thus, new versions

Corresponding Author: Hamzah Ali Alawi Al-Aidaros, School of Computing, Universiti Utara Malaysia (UUM), 06010 Sintok, Kedah, Malaysia

of AMs have been developed, enhanced and integrated with other approaches to support the increased demand of various project environments (Hamid *et al*., 2015; Wang, 2014).

Chow and Cao conducted a survey to identify success factors in Agile projects. Consequently, they found that a strong project management process with a good progress tracking mechanism is needed, however, this is still a key factor in Agile project success (Chow and Cao, 2008; Cohen and Elalouf, 2014). Besides, the lack of Agile progress monitoring, poor communication and reporting on project status are commonly stated as contributing factors to project failure (Alette and Fritzon, 2012; Taherdoost and Keshavarzsaleh, 2015). Hence, developing a new Agile approach for progress monitoring is an increasingly needed in the area of SPM. Therefore, the aims of this study are to review the previous and current approaches that are being used to monitor WIP, discuss on the pros and cons of those approaches and identify their potential elements to improve the most recent used approach.

Literature review

Agile project management approaches: Several software development approaches have been established since the inception of Information Technology (IT). These include traditional methods such as waterfall which have distinct phases and Agile methods which define a process of iteration but where design, construction and deployment of different pieces can occur simultaneously (Abrahamsson and Warsta, 2003). AMs have created a paradigm shift in the field of Software Engineering (SE) during the past two decades. Instead of fixed scope deliveries, budgets and schedules, AMs place more value on people, interaction and working software rather than on tools, contracts and plans (Wang, 2014).

Currently, AMs are being used to address the challenges of managing complex software projects during the development process (Kasunic and Nichols, 2014). They have received wide recognition within the SD field due to their flexibility and effectiveness. AMs provide a shorter development cycle, higher customer satisfaction and rapid changes to the business requirements in the SD environments (Flora and Chande, 2014). The common AMs and approaches are Xtreme Programing (XP), Dynamic Systems Development Method (DSDM), Scrum, Adaptive Software Development (ASD), Crystal, Lean Systems Development (LSD), Test-Driven Development (TDD), Feature-Driven Development (FDD) and Kanban (Ahmed *et al*., 2014). In particular, Scrum and Kanban are considered as the two powerful Agile methods in SDP. This is because they are optimizing the development process by identifying the tasks, managing time more effectively and setting-up teams (Lei *et al*., 2015).

The state of Agile development survey reported that scrum is the most followed closely method and 58% of the respondents practice it among other Agile methods (Anonymous, 2015). However, Kanban method has numerous advantages make it more popular in the area of software development. Guckenheimer and Loje (2012) stated that Kanban allows for continuous adjustment and is more prescriptive in managing queues while Scrum relies on the team commitment, reviewed at sprint boundaries.

In contrast to scrum, kanban does not use iterations but rather maximizes the workflow and shorten the lead time by limiting the amount of WIP and delivering software continuously (Flora and Chande, 2014; Michael, 2012). Moreover, Sjoberg *et al*. (2012) have compared the usage of Scrum and Kanban across a period of two years. Their study revealed that using Kanban instead of Scrum makes the company almost halved the lead-time, reduced the number of weighted bugs by 10%, improved productivity by 21% for implemented features and reduced bugs by 11%. Additionally, they argue that Kanban method eliminates the waste of periodic Scrum meetings and the rigidity of Scrum time boxing while keeping all other advantages of the Agile approach.

Furthermore, Lei *et al*. (2015) have been conducted study to see if there is a statistically difference between these two methods impacting a set of six factors of project management for SDPs which are schedule handling, budget handling, amount of available resources, risk control, quality of the project and clear project scope. The results showed that there is no statistically significant difference at the 95% confidence level between Kanban and Scrum for the aforementioned factors. However, it indicated that Kanban performs better than Scrum in term of managing project schedule. Besides, it suggested that projects that use Kanban method could experience greater consistency in term of the project management factors. Table 1 shows a comparative summary between Scrum and Kanban methods from different perspectives.

Table 1: A comparative summary between Scrum and Kanban

Perspectives	Scrum	Kanban
Time-boxed iterations	Prescribed	Optional
Default metric for planning and process improvement	Uses velocity	Uses lead time
Diagram	Burn down chart prescribed	No particular type of diagram
WIP limited indirectly	Per sprint	Per workflow state
Estimation	Prescribed	Optional
Adding items	Cannot add items to ongoing iteration	Can add new items
Prioritization	Prescribed	Optional

MATERIALS AND METHODS

Li *et al*. (2008) developed a method for monitoring software project progress based on the Earned Value Analysis (EVA) method and Use Case Points (UCP). This method has enabled management to plan accurate software projects and to improve the tracking process and results prognosis. It was based on two kinds of plans: the phase plan and the iteration plan. The purpose of the phase plan is to capture all information of a project per one cycle for example, required resources and minor milestones. The iteration plan is a fine-grained plan and there is one per iteration. In addition, each team member should assign data during each development cycle. Data such as planned start dates, completion dates and labor-day budgets are assigned to each subtask in accordance with the project schedule. Then the project data of task status, completion dates and spent dollars are regularly gathered and the earned values are converted into UCP for calculating EVA parameters. Moreover, the project progress tracking was visualized in an EVA diagram. The results of a controlled experiment indicated the effectiveness and usefulness of this method for tracking as well as controlling schedule and costs in software projects. However, this method needs more experiments with different project types and scales. This can help in classifying the data and defining different criteria to select them for a particular software project.

An integrated method for software project success monitoring has proposed by Kazi *et al*. (2012). This method was based on Projects in Controlled Environments, Version 2 (PRINCE2) method and balanced scorecard method. PRINCE2 has been developed in 1975 with its latest version being launched in 2009. It is highly prescriptive, especially on process structure but adaptable to a project of any size. Additionally to this method, Key Performance Indicators (KPIs) have been defined as measures to be used while collecting data during project implementation. Although, the proposed method does not visualize the work, it could be useful in large SDOs. Nevertheless, this proposed method was neither developed nor presented.

Petersen and Wohlin (2011) integrated a set of metrics with Cumulative Flow Diagrams (CFD). The main contribution was an approach that contains a set of measures to track the progress of software product development and to achieve a higher throughput. Data are collected for measurements and visualization by CFD. Moreover, the current status is maintained in a database and documented by a time and date stamp. The results of this study indicated that this approach has reduced lead-time by removing waste-free resources that focused on value-adding activities. It also indicated that the visualization and measures are especially valuable when developing large-scale products with many teams and tasks being conducted at the same time. However, the results suggested the need for further study to implement AMs such as Kanban. Additionally, from an analysis perspective, other usefulness methods such as Value Stream Maps (VSM) could be used to allow management to achieve more benefits when they are used in SE.

A control chart is a useful approach for monitoring process variations over time and it can help practitioners to manage process performance quantitatively. Although, several studies have indicated the effectiveness of using control charts to monitor a SDP, some controversy surrounds the suitability of utilizing control charts directly in the software industry (Baldassarre *et al*., 2007). For instance, it requires a large number of observations to accurately estimate process parameters and establish valid control limits. However, this amount of process data is often unobtainable in SD environments. Therefore, errors may occur in statistical reasoning when utilizing a control chart. Thus, Chang and Tong (2013) proposed a Q chart as an alternative approach to the control chart to overcome such issues for the SDP. Due to its visualization property, it can help project managers simultaneously evaluate schedule and cost performance. The advantages of applying the Q chart to monitor SDP are early detection capability, real-time process monitoring and fixed control limits. The Q chart, however is limited to a normal process with unknown parameters for monitoring the process mean. Thus, future research was suggested to focus on the application of Q charts for attribute data such as nonconformities for various metrics collected in a software environment.

Zhang (2011) stated that Gantt charts, cumulative cost curves and resource load charts which are used in project monitoring and scheduling are considered as graphical approaches. These approaches, however provide only visual effects and contain no quantitative information to help in software project's monitoring. Therefore, he introduced a systematic model based on the EVA method which is a very effective project control method. EVA not only achieves the overall monitoring of the project but also provides a lot of quantitative information. EVA is used for cost and schedule monitoring. It needs three basic variables; Planned Value (PV), Actual Cost (AC) and Earned Value (EV). This approach is based on the combination of human factors such as fatigue and software project monitoring. However, it is only suitable for medium and small-sized SDOs. For large and complex software projects, study suggested using this approach in conjunction with the critical path method. A further study could be focused on other factors such as project deviation and schedule compression.

A review to the current studies on project monitoring approaches and analytical models has conducted by Hazir (2015). This review has focused on literature related to EVA, optimization tools and the design of Decision Support Systems (DSS) that could contribute to helping project managers monitor and control software projects. In particular, Hazir described EVA as the most widely used managerial control method and it uses monetary units as a common basis to measure and communicate project progress. EVA is also used to estimate the expected project time and cost using the project's current status (Vanhoucke, 2009). Thus, Hazir emphasized the integration of relevant optimization models into DSS and project management software. This needs, however to develop analytical models using EVA metrics to forecast project performance. Additionally, DSS should be model-driven, function as early warning systems and be integrated into commercial project management software. Even though EVA has been receiving increasingly more attention from project managers, there are some limitations in implementing EVA in practice. Hall (2012) stated them as critical and noncritical activities which are not differentiated, activities are assumed to be independent, behavioural aspects of management are not taken into account, quality of processes and output are not assessed and the information requirement is high.

Ong *et al*. (2016) integrated the two efficient planning and monitoring approaches EVA and Gantt charts in order to develop a new approach named earned value Gantt chart (EV-Gantt) for portfolio planning and monitoring. Basic project information such as contract sum, payment to date, planned schedule and actual progress are collected as the foundation. Also, the Estimate At Complete (EAC) is calculated at the date of reporting progress. EV-Gantt adapts the Gantt chart to visualize multiple individual projects into a portfolio interface and uses EVA to analyse the performance of each individual project. This approach has been tested through scenario analysis and a pilot run using a portfolio of real cases. The results indicated that EV-Gantt performed satisfactorily in resource allocation, systematic reporting and monitoring enhancement. However, EV-Gantt must be enhanced by developing other relevant dimensions such as quality control.

The Scrum and Kanban methods have been integrated to contribute a new method called Scrumban. This method combines the most important Scrum practices which are release and sprint planning, regular delivery of increments and frequent feedback with basic Kanban principles which are visualization of workflow, limiting the WIP and change management. In order to visualize the workflow, Kanban board is used. It is suitable for complex development projects (Flora and Chande, 2014; Kniberg and Skarin, 2010; Ladas, 2009).

However, Scrumban method has such challenges with managing WIP and needs specific skills and training (Leskinen, 2015).

Cohen and Elalouf (2014) proposed a method for mitigating the dilemma between running a project in accordance with the AM and the need of a manager to monitor a project's progress. The method which combines the Kanban method with the principles of Earned Value Management (EVM), was based on calculating a Schedule Performance Index (SPI) for each evaluation point. SPI is calculated as the ratio between the project's actual earned value at the evaluation point and its planned value for that point. The planned value can be manually adjusted based on project resource allocation. At specific points in time, the project manager calculates an overall current value for the project, determined according to the current status of the various tasks. This method also uses Kanban board to visualize the workflow. However, this method is limited to simulating and predicting the planned values which can be more challenging in projects that are less stable and tend to have frequent plan changes.

Furthermore, the Kanban method and Value Stream Mapping (VSM) have been integrated to improve productivity and quality in SDP (Raju and Krishnegowda, 2014). Function Point Analysis (FPA) has been adopted to monitor the performance of SDP. FPA can help significantly improve the probability of successfully completing a software project, on time and within budget (George and Vogt, 2008). For all iterations, different function points are planned. Then the Mean First Time Yield (Mean FTY) over four iterations is calculated. In this approach, Kanban board was the core of visualizing and VSM visualizing the SD process, showing productive times and waste/wait time. Therefore, the real-time updates were made available to all team members. Furthermore, reduced cost helped to allocate optimum resources to deliver value to the customer. The results of this study indicated that this integration could positively affect the productivity and quality in SDP. Although, in a few studies, the VSM method was successfully applied in a SE context, some shortcomings have been observed, in particular, failing to capture the dynamic nature of the software process to evaluate improvements (Ali *et al*., 2015).

Table 2 summarizes the previous studies on monitoring approaches that have been used in SPM. This summary is made according to four main attributes: parameters, project type, resource and visualization technique. Each approach either receives a check mark in case of it has this attribute or an "x" mark in the other case. Parameters refer to values that are used as inputs or requirements such as start date, end date, input values, the number of WIP limits. However, project type refers to project scale including small, medium, large, simple or

Table 2: Summary of previous studies of SPM monitoring approaches with their attributes

		Attributes			
Researcher(s) and years	Used approach(es)	Input parameters	Project type	Resources	Visualization
Li *et al.* (2008)	UCP and EVA	√	X	X	√
Kazi *et al.* (2012)	PRINCE 2 and (BSC)	√	√	√	X
Petersen and Wohlin (2011)	A set of metrics integrated with (CFD)	√	√	√	√
Chang and Tong (2013)	Control and Q chart	X	X	√	√
Zhang (2011)	EVA	√	√	√	X
Hazir (2015)	EVA	√	X	√	X
Ong *et al.* (2016)	EVA and Gantt chart	√	X	√	√
Ladas (2009), Kniberg and Skarin (2010) and Mahnic (2014)	Scrum and Kanban	√	√	X	√
Cohen and Elalouf (2014)	Kanban and EVM	√	X	√	√
Raju and Krishneg (2014)	Kanban and VSM	√	X	√	√

complex. Resources refer to cost, time, duration and team size whereas visualizing refers to used technique such as diagram, chart or board. Table 2 summary of previous studies of SPM monitoring approaches with their attributes.

RESULTS AND DISCUSSION

This study reviewed several SPM approaches that were developed and recently used in SDOs. The findings clearly confirm that there is a need for a strong Agile approach with a good progress monitoring mechanism, however, this is still a key factor in the success of SDP (Chow and Cao, 2008; Cohen and Elalouf, 2014; Taherdoost and Keshavarzsaleh, 2015).

Agile approaches have received wide recognition within the SD field due to their flexibility and effectiveness. However, several studies have confirmed that failure of Agile project development is due to the lack of progress monitoring, poor communication and reporting on project status (Alette and Fritzon, 2012; Taherdoost and Keshavarzsaleh, 2015). In spite of considering that Scrum and Kanban are the two most powerful AMs (Lei *et al.*, 2015), Kanban, however, performs better than Scrum in terms of managing project schedule and queues (Guckenheimer and Loje, 2012). Additionally, Kanban does not use iterations but maximizes the workflow by limiting the amount of WIP and continuously delivering software (Flora and Chande, 2014; Michael, 2012). Moreover, the use of Kanban instead of Scrum has almost halved the lead time, reduced the number of weighted bugs, improved productivity and reduced bugs of the SDOs (Sjoberg *et al.*, 2012).

Mahnic (2014) stated that Kanban is still in the initial phases in the area of SE and has several limitations and challenges. For instance, motivating team members to use Kanban has been challenging due to the organizational culture and people's adhering to other familiar SD methods. It also has challenges with managing WIP. In addition, the lacks of specialized skills and training and the misunderstanding of core principles have been reasons for failing to adopt Kanban. Furthermore, it needs other supporting methods or tools to work effectively (Ahmad *et al.*, 2014; Hamid *et al.*, 2015), therefore, it has been integrated with Scrum (Mahnic, 2014), EVM (Cohen and Elalouf, 2014) and VSM (Raju and Krishnegowda, 2014). Additionally, Kanban still has challenges in projects that are less stable and tend to have frequent plan changes (Cohen and Elalouf, 2014). Moreover, Petersen and Wohlin (2011) argued to implement Kanban because it is an area for future research. Furthermore, Raju and Krishnegowda (2014) suggested that the integration of Kanban method with any existing approach could generate a strong approach for SPM and enhance the adoption of Agile methods.

From Table 2, it can be seen that the EVA demonstrates the best approach to monitor project, due to its numerous properties and it embraces the most attributes. EVA is an efficient project monitoring method that used to estimate the expected project time and cost. However for large and complex software projects, EVA must be used in conjunction with other methods (Hazir, 2015; Ong *et al.*, 2016; Zhang, 2011).

Finally, the graphical approaches are limited to provide only visual effects and do not contain quantitative information to help in software project's monitoring. While other approaches are complex and required a large number of observations, more experiments with different scales of software projects and further development and presentation (Chang and Tong, 2013; Kazi *et al.*, 2012).

CONCLUSION

Monitoring a software project development is an essential task to ensure that the project progresses according to budget, schedule and quality expectations. In this study, reviews of the current progress monitoring approaches in SPM have been presented with their attributes, strengths and weaknesses. Recently, AMs have caught the attention and received wide recognition within the SD field due to their ability to address the challenges of managing complex software projects. However, various studies showed that AMs have lacking in progress monitoring. Results of this study have indicated that a strong Agile approach is needed to adequately monitor the SD process because monitoring is vital to the success of such projects. Despite arguments that Agile Kanban method is still in initial phases this study, however, emphasized that Kanban has influenced on the area of SDPs due to its numerous attributes and strengths elements. Additionally, results revealed that EVA is the most monitoring approach that has used in SD.

SUGGESTIONS

Future research should be focused on examining the elements of Kanban method to understand and improve that method in the context of SD. This is because Kanban is still lacking in progress monitoring and difficult to determine the ideal number of WIP limits. Moreover, the researches on the EVA method is scattered, scarce and lacking a unifying overview. Therefore, an exploratory study of the EVA fundamentals and how it tracks the progress of a software project is vital. Furthermore, it is worth to improve Kanban method as this method still has certain limitations. It could be done by integrating it with an EVA method in order to make the most of both. However, the validity of this proposed method has to be verified in the forthcoming works.

ACKNOWLEDGEMENT

The researcher wish to thank the Ministry of Education Malaysia for funding this study under Fundamental Research Grant Scheme (FRGS), S/O Project code: 12818.

REFERENCES

Abrahamsson, P. and J. Warsta, 2003. New directions on agile methods: A comparative analysis. Proceedings of the 25th International Conference on Software Engineering, May 3-10, Portland, Oregon, IEEE Computer Society, pp: 244-254.

Ahmad, M., J. Markkula, M. Oivo and P. Kuvaja, 2014. Usage of Kanban in software companies an empirical study on motivation, benefits and challenges. Proceedings of the 9th International Conference on Software Engineering Advances, October 12-16, 2014, Nice, France, ISBN:978-1-61208-367-4, pp: 1-7.

Ahmed, T., J. Cox and L. Girvan, 2014. Developing Information Systems: Practical Guidance for IT Professionals. The Chartered Institute for IT, Swindon, England, ISBN:9781780172453, Pages: 320.

Alette, T. and V.E. Fritzon, 2012. Introducing product and process visualizations to support software development. MSc Thesis, Chalmers University of Technology, Gothenburg, Sweden.

Ali, N.B., K. Petersen and B.B.N.D. Franca, 2015. Evaluation of simulation-assisted value stream mapping for software product development: Two industrial cases. Inf. Software Technol., 68: 45-61.

Anand, G., A. Chandrashekar and G. Narayanamurthy, 2014. Business process reengineering through lean thinking: A case study. J. Enterp. Transformation, 4: 123-150.

Anonymous, 2015. 10th annual state of agile development survey. VersionOne Inc, Alpharetta, Georgia.

Baldassarre, M.T., D. Caivano, B. Kitchenham and G. Visaggio, 2007. Systematic review of statistical process control: An experience report. Proceedings of the 11th International Conference on Evaluation and Assessment in Software Engineering, April 2-3, 2007, IET, Lucknow, India, pp: 83-93.

Chang, C.W. and L.I. Tong, 2013. Monitoring the software development process using a short-run control chart. Software Qual. J., 21: 479-499.

Chow, T. and D.B. Cao, 2008. A survey study of critical success factors in agile software projects. J. Syst. Software, 81: 961-971.

Cohen, N. and A. Elalouf, 2014. How to manage an Agile/Kanban software project using EVM. Intl. J. Comput. Inf. Technol., 3: 1-5.

Doraisamy, M., S. Ibrahim and M.N.R. Mahrin, 2014. Formulation of metric based software project performance monitoring model: A roadmap. Proceedings of the IEEE Conference on Open Systems (ICOS), October 26-28, 2014, IEEE, Subang, Malaysia, ISBN:978-1-4799-6368-3, pp: 48-53.

Flora, H.K. and S.V. Chande, 2014. A systematic study on agile software development methodologies and practices. Intl. J. Comput. Sci. Inf. Technol., 5: 3626-3637.

George, R. and T. Vogt, 2008. Illustrative example of a function point analysis for the NASA crew exploration vehicle guidance, navigation and control flight software. Johnson Space Center, Houston, Texas.

Guckenheimer, S. and N. Loje, 2012. Agile Software Engineering with Visual Studio: From Concept to Continuous Feedback. Addison-Wesley Professional, Boston, Massachusetts, ISBN-10:0-321-68585-7, Pages: 289.

Hall, N.G., 2012. Project management: Recent developments and research opportunities. J. Syst. Sci. Syst. Eng., 21: 129-143.

Hamid, S.S., M.H.N.M. Nasir, M.K. Othman and R. Ahmadi, 2015. Factors limiting the implementations of agile practices in the software industry: A pilot systematic review. Indian J. Sci. Technol., Vol. 8, 10.17485/ijst/2015/v8i30/87111.

Hazir, O., 2015. A review of analytical models, approaches and decision support tools in project monitoring and control. Intl. J. Project Manage., 33: 808-815.

Kasunic, M. and W. Nichols, 2014. When measurement benefits the measured. Master Thesis, Carnegie-mellon university, Pittsburgh, Pennsylvania.

Kaur, R., S. Singh and M. Rakshit, 2013. A review of various software project scheduling techniques. Intl. J. Comput. Sci. Eng. Technol., 4: 877-882.

Kazi, L., B. Radulovic and Z. Kazi, 2012. Performance inicators in software project monitoring: Balanced scorecard approach. Proceedings of the IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics (SISY), September 20-22, 2012, IEEE, Subotica, Serbia, ISBN:978-1-4673-4751-8, pp: 19-25.

Kniberg, H. and M. Skarin, 2010. Kanban and Scrum-Making the Most of Both. C4Media Inc., Toronto, Ontario, Canada, ISBN:978-0-557-13832-6, Pages:107.

Ladas, C., 2009. Scrumban: Essays on Kanban Systems for Lean Software Development. Modus Cooperandi Press, Seattle, Washington, Raleigh, North Carolina, ISBN:978-0-578-00214-9, Pages: 89.

Lei, H., F. Ganjeizadeh, P.K. Jayachandran and P. Ozcan, 2015. A statistical analysis of the effects of Scrum and Kanban on software development projects. Rob. Comput. Integr. Manuf., 43: 59-67.

Leskinen, M., 2015. Towards a flow efficient ICT development process with Kanban: A case study. Master Thesis, Aalto University, Espoo, Finland.

Li, J., Z. Ma and H. Dong, 2008. Monitoring software projects with earned value analysis and use case point. Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science ICIS 08, May 14-16, 2008, IEEE, Portland, Oregon, ISBN:978-0-7695-3131-1, pp: 475-480.

Mahnic, V., 2014. Improving Software Development through Combination of Scrum and Kanban. In: Recent Advances in Computer Engineering, Communications and Information Technology, Music, J. (Ed.). WSEAS Press, Benidorm, Spain, ISBN:978-960-474-361-2, pp: 281-288.

Michael, H.T., 2012. Agile ALM: Lightweight Tools and Agile Strategies. Manning Publisher, Greenwich, Connecticut,.

Ong, H.Y., C. Wang and N. Zainon, 2016. Integrated Earned Value Gantt chart (EV-Gantt) tool for project portfolio planning and monitoring optimization. Eng. Manage. J., 28: 39-53.

Petersen, K. and C. Wohlin, 2011. Measuring the flow in lean software development. Software Pract. Experience, 41: 975-996.

Raju, H. and Y. Krishnegowda, 2014. Value stream mapping and pull system for improving productivity and quality in software development projects. Intl. J. Recent Trends Eng. Technol., 11: 24-24.

Rupinder, K. and S. Jyotsna, 2013. Software process models and analysis on failure of software development projects. Intl. J. Sci. Eng. Res., 2: 1-4.

Sjoberg, D.I., A. Johnsen and J. Solberg, 2012. Quantifying the effect of using Kanban versus scrum: A case study. Software IEEE., 29: 47-53.

TSG., 2013. CHAOS manifesto 2013-think big, act small. The Standish Group International Inc, Yarmouth, Massachusetts.

Taherdoost, H. and A. Keshavarzsaleh, 2015. A theoretical review on it project success/failure factors and evaluating the associated risks. Proceedings of the 14th International Conference on Telecommunications and Informatics, September 20-22, 2015, Michigan State University, East Lansing, Michigan, pp: 1-9.

Tarom, T.D., 2014. Software project management using 8D application techniques. Software Eng. Technol., 6: 179-182.

Vanhoucke, M., 2009. Measuring Time: Improving Project Performance using Earned Value Management. Vol. 136, Springer Science & Business Media, Berlin, Germany,.

Wang, H., 2014. The review and research on agile oriented method in the pilot industry system. Comput. Modell. New Technol., 18: 895-901.

Zhang, J.G., 2011. Schedule management method study of middle and small software projects. Proceedings of the 2nd IEEE International Conference on Artificial Intelligence, Management Science and Electronic Commerce AIMSEC, August 8-10, 2011, IEEE, Dengleng, China, ISBN:978-1-4577-0535-9, pp: 1495-1498.