Literature Survey on KANBAN: Opportunities and Challenges

Fizza Kamal
Department of Computer Software Engineering, Military College of Signals, NUST
Rawalpindi, Pakistan
Email: fizza.kamal10@gmail.com

DOI: 10.29322/IJSRP.10.11.2020.p10786
http://dx.doi.org/10.29322/IJSRP.10.11.2020.p10786

Abstract**—**Every software development team thrives for success by managing the workflow and team coordination for timely product delivery with optimum customer satisfaction. Although not a new concept, agile is a requirement for future sustainability. It can be shown that businesses that fail to adopt some form of agile methods for product development also fail to meet customer and market demands and become greatly disadvantaged. Kanban and Scrum are two most popular and widely adopted develop- ment methodologies. This paper focuses on Kanban as an agile software development methodology. In addition to its challenges and opportunities associated with application of Kanban as development methodology. Major benefits of using the Kanban as software development methodology are lead time improvement for software delivery, software quality improvement, effective communication, improved coordination, greater consistency in product delivery, and improved customer satisfaction.

Index Terms**—**KANBAN, agile methods, workflow, efficiency, challenges

I. Introduction

Manufacturing is the cradle of KANBAN and this enduring approach has been applied by Toyota in 1953. Slowly it is be-coming part of software development. Software development teams need to maintain high productivity to be competitive in the modern era of the fastmoving marketplace and cutthroat competition, which can only be achieved by managing the workflow during entire development process efficiently.

In traditional software development methodologies (water-fall model), it is quite not possible to achieve high level of productivity. Software development methodologies need to be customer-centric [2]. Also, those methodologies should be able to manage the changing requirements of the customer during development process.

According to a rough estimation, the agile software devel-opment method is used in around 85% of the leading software companies. As agile involves customer opinion with continual delivery of software products, thus customer satisfaction is acquired. KANBAN is an agile methodology that focuses on efficient teamwork and managing workflow effectively. KAN-BAN helps in achieving customer satisfaction, timely delivery of product, improved understanding and communication of team.

II. Literature Review

A qualitative case study was conducted in an organiza- tion to observe the service production of KANBAN been implemented in an IT organization with respect to change

management in [1], and how the operational process of the organization are being changed by the KANBAN. The results of their research conclude that there is minimal resistance to change implementation in Volvo IT. The changes in implementation, are effectively handled by KANBAN, done by the team members.

Another study [2] focuses on the combination of Kanban and scrum, concludes that "Scrum ban" helps in process and productivity yielding beneficial economic outcomes.

Key ultimatums for Agile KANBAN method are i.e. Shortage of advance tracking technique, it's hard to determine WIP limits due to lack of amp technique, and monitor project progress is not being provided by enough information and useful indicators [3].

In [4] the state of Baja California, using questionnaires for gathering data from different manufacturing companies, a model has been designed that depicts the influence of the planning phase and the advantages acquired in the enforcement of the KANBAN system.

The issues faced in the lean-agile software development environment. are addressed in [5]. The study suggests that in order to sustain in this competitive world of software development, one must refrain from using old traditional methods. New and efficient methods should be adopted in order to deliver the software on time.

Kanban by itself does not assure success as it is a comparatively basic flow tool that needs to be supported by other practices [6].

A systematic literature review [7] examines the current trends of Kanban usage in software development along with identification of the benefits and challenges involved. The study suggests that Kanban should be used because it's simple and it focuses on the workflow. Moreover, due to the usage of Kanban, customer satisfaction is achieved, and software quality is improved in addition to on-time delivery is done.

M. L. Mittal [8] examines the impact of lean methodology in software development projects and its status of implementation in software development projects. The results of the study conclude that the concepts of lean methods are evolving in SD projects. The problem faced in adopting lean method was more commonly found with customers, process and people.

Daryl J.Powell [9] aims to determine whether Kanban can be used for lean production in high mix, low volume environments. Strict limits are set for Work In Progress (WIP) after the

research by using two case studies on Toyotas manufacturing process using Con WIP principle. According to the research, by introducing an effective pull mechanism KANBAN board can be used to accelerate in high-mix, low-volume manufactur- ing environments based on available resource capacity rather than stock renewal. It also emphasizes daily standups.

Another study [10], focuses on the strengths and weaknesses of agile and traditional methodologies. The results of the study show that they're not the one for all methods for software development. The combination of processes has helped many organizations to overcome the weaknesses of a single process, that's why many organizations have adapted these combinations of processes.

Hamzah Alaidaros [11] observes the current tasks and examines the challenges which are in progress of Agile KANBAN method. Research reveals that Agile KANBAN technique lacks progress tracking mechanism; thus, this method needs to be combined with another method to be an effective method. The study suggests that a lot of changes must be made in Kanban methodologies so that it can be used more extensively.

III. Agile Methodology

Agile software development methodology revolves around the idea of iterative development, as shown in Fig. 1. In agile methodology, requirements and solutions are modified through teamwork of multi-functional and self-regulating teams. The essential aspect of Agile development is that it allows teams to deliver project more rapidly, with superior quality and certainty, and greater capacity to accept and respond to mod- ification or variation.

Fig. 1. Agile Methodology

The utmost admired agile methodologies are Kanban, Scrum, Feature Driven Development (FDD), Adaptive Soft- ware Development (ASD), eXtreme Programming (XP), Crys- tal, Dynamic Systems Development Method (DSDM) and Lean Software Development (LSD). Every single agile development framework ensures the successful release of the software product as a result of iterative planning, development, testing and integration

The key benefits of agile process framework are:

1)Customer Collaboration
2)Rapid Development.
3)Reduced eisk of error
4)Quick response to changes
5)Customer satisfaction

A. Strengths and Weaknesses of different agile process frameworks:

Table 1. Strengths and Weaknesses of Agile Software Processes [10]

Process Framework	Strength	Weaknesses
Scrum	• Effective and efficient communication among team members. • Continuous feedback from the customers. • Produces quality product with customer satisfaction. • Measuring the growth and productivity of the team and individual is easier with daily Scrum meetings and sprint meetings. • Quality product with customer satisfaction. • Can easily handle unclear and changing requirements.	• Employees lack knowledge of Scrum • Scrum lacks engineering practices • Simple to understand but difficult to master • Suitable for small projects
Kanban	• Helps in managing production of a product Increase in communication between the team and stalk holders • Positive impact on external quality of the system	• Lack of details about its implementation
Lean Software Development	• Eliminate waste • Maximize value of the product	• Does not cover technical and managerial issues • Lack of details about its implementation
FDD	• Adaptive and incremental in nature • Emphasis more on quality	• Needs special training to write requirement/user stories • Less responsiveness to change • Need of experienced and trained staff • Less appropriate for small scale projects
DSDM	• Users are highly involved in the development of the system so; they are more likely to get a grip on the software development project. • In this model, basic functionality is delivered quickly, with more functionality being delivered at frequent intervals	• Costly to implement, as it requires users and developers both to be trained to employ it effectively. • It may not be suitable for small organizations or one-time projects
Crystal	• Effective communication among team members • Projects can be clearly classified using Crystal methods	• Lacks system validation practices
Extreme Programming	• Pair programming and continuous integration improves productivity • Works well with simple and small-scale projects	• Lack of documentation • Poor architectural structure • Less focus on design • Pair programming requires mutual understanding and common skillset between two programmers
Test Driven Development	• Writes test cases and test code first using the requirements • Writes Lean code, removes duplicates	• Sometimes very time consuming due to repeated test failures • Specific knowledge and skill set required
Adaptive Software Development	• Focuses on collaboration and learning as a technique to build complex systems.	• Demands extensive user involvement, which can be difficult to facilitate. • Integrates testing into every stage, which can add to a project's costs. • Emphasis on rapid iterating and continuous feedback can lead to scope creep.

As shown in table 1, scrum helps in having continuous feedback from the customer. It helps in effective communication between the team. But the usage of scrum as development methodology is often limited due to lack of knowledge among developers. It does not apply to large projects. It is simple to understand yet challenging to master at it. Using scrum, the customer is highly satisfied, as daily scrum meetings are done with the customer which increases the quality of product.

Lean software development approach does not consider technical issues. It helps in eliminating the waste by focusing on only what is needed which maximizes the product value.

To implement the Feature Driven Development approach, one needs to acquire special training in writing user stories. Also, it requires an experienced and trained team to handle the requirements. It is much adaptive in nature and focuses mainly on quality of product.

Dynamic System Development Method is highly expensive, as it requires a trained and experienced team. It is suitable for large organizations only. In DSDM, users are involved in each step of software development. Using this approach, the basic functionality of the project is delivered earlier than rest of the modules which get delivered in intervals.

The crystal family lacks the validation approach. It helps in classifying the projects. The team members communicate with each other more effectively.

In eXtreme Programming, there is no need for documentation, as pair programming helps more in getting along with the understanding of the projects. It focuses less on design. But continuous integration of modules improves the productivity.

Test Driven Development approach can be time-consuming because of continuous test failures. Test cases are generated first using the requirements. It requires definite knowledge and set of skills to implement it.

In Adaptive Software Development, user involvement is required at an extensive level, which can be a challenging task. It focuses on immediate iteration and constant feedback from the user/customer. As it involves testing at every stage, the cost of the product can increase.

IV. KANBAN

Kanban is one of the most popular agile approaches that is used by practitioners in software development. The idea of Kanban is to deliver the software product continuously and incrementally without over burdening the team. The development team keeps a track of the work progress by using a Kanban board.

A. Kanban Board

The Kanban board comprises of the labelled columns which help in keeping the track of the progress of work. Each column has sticky notes that represent the task on which the team must work.

Kanban handles the Work In Progress (WIP) limit, so each column has a limit of tasks indicated at the top of each column. That abstains the team from overburdening themselves.

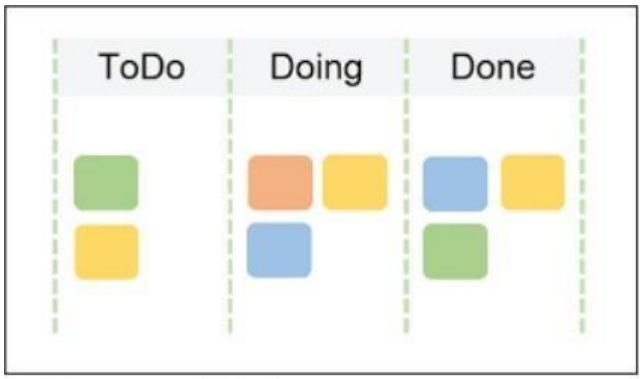
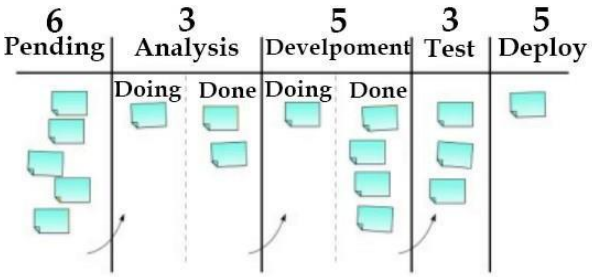


Fig. 2. Simple Kanban Board Example(1)

As shown in Fig. 2., it is the simplified form of Kanban board in which the columns are labelled as 'ToDo', 'Doing', and 'Done'. When any task newly arrives, it is categorized in the 'ToDo' column. Those tasks which are under process and the team is working upon them, are catalogued in 'Doing' column. 'Done' column comprises of those tasks which are completed.

Fig. 3. Simple Kanban Board Example(2)



In Fig. 3., the Kanban columns are labelled as 'Pending', 'Analysis', 'Development', 'Test', and 'Deploy'. The analysis and development columns are further divided into 'Doing' and 'Done' columns. The limit of tasks that each column can have, is specified at the top of the respective column. At most 6 tasks can be placed in the 'Pending' column. There can only be 3 tasks in the 'Analysis' column. For 'Development', 'Test' and 'Deploy', the WIP limit is 5, 3 and 5 respectively.

The tasks that newly arrive are classified into the pending column. Then, if there is a vacancy for a new task, only then the new tasks are pulled by the system and can be categorized as 'Doing' in the analysis column. Once the analysis is completed, that task is labelled as 'Done' by analyst.

That task is the pulled by the system into the next phase that is the development phase. Those tasks which are under process, are labelled as 'Doing' and those which are completed,

are labelled as 'Done'. Then the next phases are testing and deployment phases. There can only be at most 3 modules for testing. And deployment can have 5 modules.

B. Motivation to use Kanban

There are several factors that motivates a practitioner to use Kanban approach for their software product development, some are enlisted below:

1)Simplicity
2)Visualize workflow
3)Focuses on workflow
4)Customer satisfaction
5)Improved Software quality
6)No massive documentation
7)Adaptive to changes in requirements
8)Continuous flow of work
9)WIP limit

Kanban is the most preferable agile process work due to its simplicity. It uses Kanban board on which the work progress can be visualized by categorizing it into 'to do', 'in progress', 'testing' and 'done'. It makes it easier for the team to keep track of processes completed or in progress. Also, it shows the roles and responsibilities of each team member. Which makes it easier for the project manager to keep track of work progress of each module.

Another motivation to use Kanban is that it limits the work in progress. It sets the limit of work to be done in one each status of the workflow. It is called a push and pull system. A new process is pulled by the system if there is vacancy/ space for new work. Otherwise, it is not pushed into the system unless the existing process is completed.

As it involves team cohesion and collaboration, so it is guaranteed to have improved software quality and customer satisfaction.

Kanban is highly adaptive to changes in requirements. The changes are added to the process as they are needed, and the team do not have to wait unlike in scrum.

Kanban focuses on the development of those items which are needed and requested. Due to this approach, the workflow is balanced and remains maintained which ensures the constant releases of work items to the customer.

Kanban is preferred because it uses shorter feedback loops from the customer. And those feedbacks help to remove bugs and errors at an early stage and the developer doesn't require to have a hard time debugging. Also, customer satisfaction is acquired.

C. Benefits of kanban

Along with several other benefits of Kanban, some of them are listed below:

1)Customer Satisfaction
2)Improved software quality
3)Improved lead time delivery
4)Earlier feedback
5)Reduction in customer error reporting
6)Improved communication between stakeholders
7)Increased developer motivation
8)Bugs fixed more quickly
9)Better understanding of project
10)Reduced cycle time
11)Ensured team training
12)Improve transparency
13)Strategic alignment
14)Changes are added as needed

By using the Kanban process framework, one can achieve many benefits for their software product.

Top of all, customer satisfaction is acquired because of improved software quality and continuous collaboration with the customer for testing. Bugs and errors are detected at an earlier stage, that reduces the time to market for the software product. Communication and collaboration between the customer and team are improved. It increases the predictability in the delivery of the final products and more precise estimate of the work [6].

D. Challenges in Kanban

The most challenging aspect of using Kanban is that it cannot be used alone, and it requires supporting approaches like scrum. So, the combination of two is called "Scrumban". Another challenging thing about Kanban is that the team can overload/overcomplicate the Kanban board. They need to keep it simple for rest of the team to understand it easily, but some member can learn "new tricks" and apply those on Kanban board which hides the important information.

It is hard to manage WIP, as there is no time frame for any process. Only the progress is labelled as 'in progress', 'to do', 'completed'. Time frame is not defined, so we don't know how much time it will take for a task to be completed.

Getting the whole team to agree on one point is quite a difficult task. So, changing the organizational culture is quite a challenging thing when you are using Kanban.

Sometimes collaboration and communication issues occur which create a really challenging environment to work in.

E. When to use Kanban

Kanban is recommended if any of the following conditions occur

1) When it is required to include stories or alter sprints on the fly
2) When iterations are not required.
3)When there is no need to estimate.
4) When you need more control over the process and the capability to deliver the product at any time.
5) When your focus is on continuous improvement.
6)When your team is not adaptive to extreme changes
7)When you want to enhance delivery flow
8)When you want your system to be easy to grasp

V. Combination of Scrum and Kanban

It is challenging to implement Kanban as the only development methodology, so it is used in combination with other

process frameworks like scrum, which is called Scrumban. Fig. 4. depicts the methodology of Scrum. Scrumban comprises of the finest suitable policies from Scrum and Kanban. It contains the best features from Scrum and Kanban. Those are enlisted below:

A. Features of Scrum:

1)Iterations
2)Stand-ups
3)Prioritization
4)Decide length of sprint

B. Features of Kanban:

1)WIP limit
2)Pull system
3)Continuous flow of work
4)Short lead times

In Scrumban, there is no need for a precise number of members in a team. Also, they do not require to specify their roles. Scrumban uses acceptably rigorous rules and grants a possibility for a controlled Kanban process to some extent. The work cycles do not exceed more than 2 weeks. Daily stand-up meetings are carried out.

Fig. 4. Scrumban Methodology

C. Benefits of Scrumban:

The key benefits of using Scrumban are as follows:

1) The quality of product is increased as there is 25-70% reduction in errors and defects.
2) The productivity is increased by 20-45%.
3)Time to market is increased by 30-70%
4)Employees are more motivated and happier.
5)Waste is minimized

D. When to use Scrumban:

Scrumban can be used when following conditions occur:

1)When you want flexibility of workflow-based approach along with scrum structure.
2)When you want to switch between scrum and Kanban

VI. KANBAN ANALYSIS

F. Comparision between Kanban and other Agile Methods:

Table 2. Comparison of Kanban with other Agile Methods [12]

Criteria		Kanban	Scrum	Extreme programming	FDD	Crystal Family	DSDM	Lean software development	ASD
Model-based		NO	NO	NO	NO	NO	NO	NO	NO
Number of team members		Not Specified	5-9	2-10	4-20	All members	2-10	Not specified	Moderate
Scope of project		Entire	Entire	Narrow	Broad	Entire	Entire	Entire	Narrow
Daily meeting		Defined	Defined	Defined	Not defined	Not defined	Not defined	Not defined	Defined
Documentation		Not defined	Simple	Simple	Essential	Essential	Exists	Not Defined	Simple
Risk Mitigation	High Risk	Not defined	Defined	Not defined	Not defined	Defined	Defined	Not defined	Defined
	Medium Risk	Defined	Not defined	Defined	Defined	Not defined	Not defined	Defined	Not defined
Process centric		Defined	Not defined	Not defined	Defined	Not defined	Defined	Defined	Not defined
Virtual team support		Defined	Defined	Not Defined	Defined	Defined	Not Defined	Not Defined	Defined
Face to face meetings		Not Defined	Not Defined	Not Defined	Not Defined	Defined	Not Defined	Defined	Not Defined
Iteration length (week)		1	4	2	2	Project specific	Not Specified	Not Specified	4-8
People centric		Not Defined	Defined	Defined	Not Defined	Defined	Not Defined	Not Defined	Defined
Roles and responsibilities		Defined	Defined	Defined	Defined	Not Defined	Defined	Not Defined	Not Defined
Information sharing (through document)		Not Defined	Not Defined	Not Defined	Defined	Not Defined	Defined	Not Defined	Not Defined

As shown in table 2, Kanban is a lean method to organize and ameliorate work across human systems. This technique focuses on how to manage work by stabilizing requirements with available capacity, and by enhancing the handling of system-level. The number of team members is not specified with which they work. The entire team is responsible for the project's success.

In Kanban, daily meetings are prescheduled and when they're having their regular meetups so, documentation is not required. Kanban is a non-riotous evolutionary change management system this shows that the process is improved in small steps. By implementing many minor changes, the risk to the overall system is reduced.

Kanban is a process-centric methodology which includes the support of the virtual team as it is needed to ensure the teams' success.

Face to face meetings are not defined as people are working on their tasks. You may have a prioritization process and not need to hold discussions on it; you may find that you can do perfectly well with a weekly team meeting instead of a daily one. Only a week is given for the iteration. Roles and responsibilities are defined in it. Information sharing through documentation is not pre-defined in Kanban.

According to our analysis, though Kanban is beneficial due to its simplicity and other facts, but Kanban cannot be used alone. It must be used in combination with other methodology. Research on Kanban in Software engineering has been a quite popular topic among researchers during year 2018.

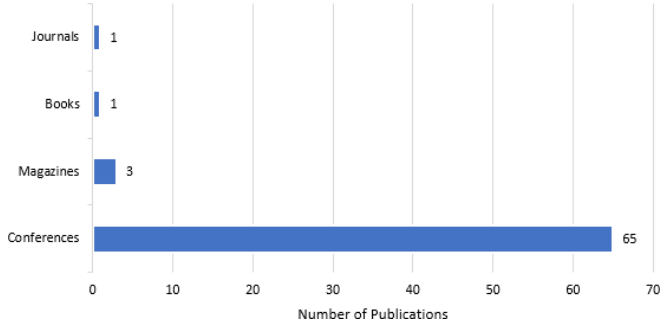


Fig. 5. IEEE Xplore Publications on Kanban since 2016 to 2019

As Fig.5 shows that total 70 publications regarding Kan- ban have been published in IEEE XPLORE since 2016 till 2019. Out which those 70 publications, 65 are published in conferences, 3 in magazines, 1 in Book and Journal each.

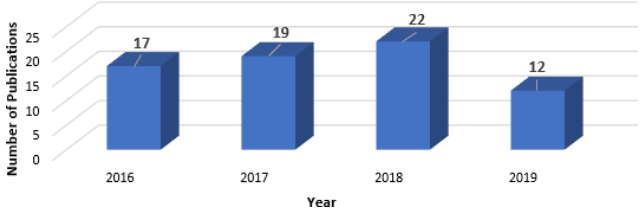


Fig. 6. IEEE Xplore Publications on Kanban Each Year

Fig.6 depicts the number of publications in IEEE XPLORE each year since 2016 to 2019. In year 2016, total of 17 papers have been published in IEEE XPLORE regarding Kanban. Whereas in 2017, 19 papers were published. Kanban has been a trending topic in year 2018 as total 22 papers have been published in IEEE XPLORE.

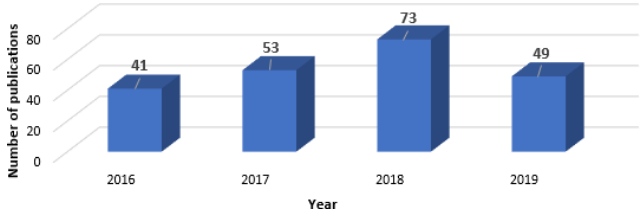


Fig. 7. ACM Digital Library Publications on Kanban Each Year

In ACM Digital Library, total 216 papers have been published regarding Kanban during years 2016 to 2019. The graph in Fig. 7 depicts the number of papers published each year since 2016. During year 2016, total 41 papers were published in ACM Digital library that were related to Kanban. Whereas, the count increased next year in 2017 and total 53 papers were published. Kanban has been a hot topic in 2018 as total 73 papers were published in ACM Digital Library during 2018. This rapid increase in publications, decreased abruptly and reached to 49 in year 2019.

VII. Conclusion

The aim of this paper was to focus on Kanban as an agile software development methodology. Along with challenges and opportunities associated with applications of Kanban as development methodology. The key advantage of applying the Kanban as software development methodology are lead time improvement for software delivery, software quality improvement, effective communication, improved coordination, greater consistency in product delivery, and improved customer satisfaction. Though, the advantages of Kanban identified in this paper are more than the challenges, Kanban alone does not assure success as it is comparatively a basic flow tool that needs a support of other process frameworks. The combination of Scrum and Kanban is found to be beneficial as it increases the quality of product as there is 25-70

References

[1] MAHGOL AMIN, "KANBAN Implementation from a Change Management Perspective: A Case Study of Volvo IT", School ofBusiness, Society and Engineering, Mälardalen University (2014)
[2] Sanjay Pandit Patil, Dr. Jitesh R. Neve, "Productivity Improvement of Software Development Process through Scrumban: A Practitioner's Approach", International Conference on Advances in Communication and Computing Technology (ICACCT)(2018)
[3] Hamzah Alaidaros, "Identification of criteria affecting software project monitoring task of Agile Kanban method", Published by the American Institute of Physics,2016
[4] Maria Mojarro, "Impact of the Planning from the Kanban system on the company's Operating Benifits", Article in sustainability. July 2018.
[5] Udit Kumar Nath, "Issues of lean-agile software development environment", International Journal of Engineering and Technology, 7 (2.33)(2018) 432-437

[6] Muhammad Ovais Ahmad, Denis Dennehy, “Kanban in software engineering: A systematic mapping study”, The Journal of Systems and Software 137 (2018) 96–113
[7] Ahmad, M. O., Markkula, J., and Oivo, M. (2013, September).“Kanban in software development: A systematic literature review”, In Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on (pp. 9-16). IEEE.
[8] RamKaran Yadav, M. L. Mittal, and Rakesh Jain, “Lean practices in software development projects: A literature review”, AIP Conference Proceedings 2148, 030044 (2019)
[9] Daryl J.Powell, “ Kanban for Lean Production in High Mix, Low Volume Environments”, IFAC PapersOnLine 51-11 (2018) 140–143
[10] Mahrukh Sameen Mirza, “Strengths and Weakness of Traditional and Agile Processes ; A Systematic Review”, Journal of Software, Volume 14, Number 5, May 2019
[11] Hamzah Alaidaros, “Towards an Improved Software Project Monitoring Task Model of Agile Kanban Method”, Int. J Sup. Chain. Mgt, Vol. 7, No. 3, June 2018
[12] Soukaina Merzouk, Sakina Elhadi, “A Comparative Study of Agile Methods: Towards a New Model-based Method”, International Journal of Web Applications Vol. 9, No., 4 December 2017
[13] agilemanifesto.org, “Principles behind the Agile Manifesto”, [online]. Available : http://agilemanifesto.org/principles.html. [Accessed: 29-Dec- 2019]