I.J. Information Engineering and Electronic Business, **2018, 3, 23-35**
Published Online May 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2018.03.04

The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development

Aysha Abdullah Albarqi, Rizwan Qureshi
King Abdulaziz University/ Faculty of Computing & Information Technology, Department of Information Technology, Jeddah, Saudi Arabia
Email: albarqiaisha@yahoo.com, rmuhammd@kau.edu.sa

Received: 09 October 2017; Accepted: 23 December 2017; Published: 08 May 2018

***Abstract*—**Agile software development methodologies gaining the attention in the field of software engineering. There are several methods of agile such as Scrum, Lean, and Kanban. Scrum methodology divides the product into series of sprints. Lean is agile toolkit which has seven principles that facilitate: eliminating the wastes, delivering fast, and improving value for the final customer. Kanban is a visual method that can help in managing the production. To take the advantages of the following methodologies: Lean, Scrum, and Kanban we can integrate them together thus, the result will be a new methodology that can contribute in enhancing and improving the efficiency of the software development process, which is the aim of this thesis. An integrated methodology that integrating Scrum, Kanban, and Lean methodologies to yield a comprehensive agile methodology called L-ScrumBan has been proposed. The validation of the proposed methodology has been done through a survey by using a questionnaire; the survey results confirmed the efficiency of the proposed methodology.

***Index Terms*—**Software development, Agile, Scrum, Kanban, Lean, ScrumBan and L-ScrumBan

I. Introduction

Since a decade, Agile software development (ASD) has been in the software industry. Agile methodologies gained the attention in the field of software development where it is tailored to: manage the dynamic requirements, meet the customer satisfaction, and deliver the product in a way that suits the customer demands with the required quality [1-2], which in turn lead to achieve the aim of the software industry. The agile methodologies such as Scrum, Lean and Kanban have effective effect in the field of software engineering especially in software development [3].

Scrum methodology is an agile framework that has the feature of incremental and iterative, it's dedicated to software development and project management, its focus on delivering the desired product in the desired time with the maximum value [4-5].

The process of scrum starting with translates the customer's product into the product backlog which is the responsibility of the product owner. After the product backlog has been completed, the team starts the sprints (usually each sprint takes 2 - 4 weeks). The team has to hold a daily scrum meeting, and before establishing each sprint the team has to hold sprint planning to determine what it should be done by the end of this sprint, and by the end of each sprint, the team has to hold sprint review to gather the feedback from the customer. By the end of each sprint, there is a product or a feature has been completed and it's ready to be delivered to the final customer [6-8].

Lean is an agile toolkit that comprises several principles for managing software development where its aim as illustrated by Laura et al. [9]: to "eliminate any activities that considered as unnecessary activity for the product, maximize the value of the customer, build quality in, defer commitment, create knowledge, deliver fast, optimize the whole, and respect people". Lean proposed new tool that supports its ultimate goal by removing all the wastes and focus on the value maximization, which is called Kanban [10].

Kanban is "a Japanese word refers to a signboard" as it's defined by Marko et al. [11] which is a lean tool that has been applied as project management model that can monitor the workflow of the project [12]. Kanban aims to: monitor the workflow by visualizing the work, minimize the cost of production, increase quality, limit the work in progress(WIP), and minimize the cycle time which is in turn lead to deliver the product faster [8].

II. Related Work

Agile Software development methodologies are the hottest and recent trend in the software engineering domain. There are several researches and scientific studies on Agile methodologies that emphasized the success of implementing agile methodologies such as: Scrum, Lean Software Development, Kanban, and the combinations of these methodologies such as ScrumBan. Several scientific researches proved that by combining principles and practices of two or more methodologies with each other that will enhance the software development process [13].

A. Agile Software Development

In case of the dynamically requirements that have to meet changing in environments, delivering the software fast as possible with a high quality in a short time besides the customer satisfaction, the industry field need methods that can achieve the desired service at its best, by implementing agile software development methodologies all the software development demands are achievable.

Robert [14] described the proposing of the Agile manifesto as "A group of industry experts calling themselves the Agile Alliance met in early 2001 to create a statement of values and principles that would allow software teams to develop quickly and respond to change. The result was The Manifesto of the Agile Alliance".

Jeffrey [15] and Moniruzzaman et al. [16] introduced an introductions and overviews of agile software development and its methodologies, where their reports were summarized as the following, agile software development includes several methodologies that share common characteristics: iterative, adaptive (responding to the requirement changes), incremental, self-organizing team, fast delivering a high-quality software, and collaboration of the customer. In [3] by Markus et al. agile software development methodologies have been investigated whether these methodologies have a positive impact on the satisfaction of the customers in case of dynamic requirements. The study emphasized that agile software development methodologies have a significant impact on improving the customer satisfaction positively.

Agile has several methodologies for software development where each methodology has several principles and practices that distinguish it from the other.

B. Scrum

Scrum is an agile framework for software development and project management that inherited the “increment and iterative” and “self-organizing team” features from ASD. It comprises “two shared artifacts, three roles, and four types of meetings called ceremonies [17].

B.1. Scrum Artifacts

Scrum has two artifacts which are: Product backlog, sprint backlog.

1) Product Backlog: The product backlog is the most important tool which comprises all the tasks for the items of the customer product, these tasks have been prioritized according to the business value of the customer and have to be accomplished through the lifetime of the project [17]. The tasks in product backlog could be (enhancements, features, defects, …etc.) [18]. In the product backlog, each task should have a unique ID, and defined by its category (enhancement, feature, defect) [18].
2) Sprint Backlog: contains all the tasks that belong to the item should be delivered by the end of the sprint. There are no changes can be accepted after finalized the sprint backlog except in necessary and rare circumstances [17]. Each task in sprint backlog has to be defined with its estimated cost and the time that required to accomplish the task [18].

B.2. Scrum Roles

The team is consisting of all the members those who seek to accomplish the product, where it is divided into three roles which are the product owner, scrum master, and the workers [17]. The team number should be more than 4 but less than 9 [18].

1) The Product Owner: The product owner is the most important member of the scrum team which is responsible for the success of the product and its responsibility to: faciliatet the interactaction between the team member and the customer, translate the customer product into product backlog, and prioritize this product backlog according to the customer value [19].
2) Scrum Master: The scrum master is a leader, not manager where he/she is considered as one of the workers and is responsible to be sure that the team is working as planned in addition to its responsibility to remove any obstacles face them during the work [17].
3) The Workers: The workers which are the most important role, the team should be "cross-functional team" and capable of organizing and directing themselves without guiding [17].

B.3. Scrum Ceremonies

Scrum methodology has four types of ceremonies as the following: Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective [17]. These ceremonies facilitate the development process and make it more controller.

1) Sprint Planning: In sprint planning, the team decides what are the tasks have to be done in the sprint they will start it and how much time does it take [17].
2) Daily Scrum: The team members have to meet each other daily to discuss the workflow of the sprint, what they did yesterday, what they doing today, and what they will do tomorrow, this type of this ceremony called" Daily Scrum" [17].
3) Sprint Review: The third type of ceremonies is sprint Review which it takes its place by the end of the sprint to gather the feedback of the product that has been accomplished in the sprint [17].
4) Sprint Retrospective: The last one which is sprint retrospective has to be held to discuss what was good in the sprint and what needs to be done in a better way in the next sprint [17].

B.4. Scrum Workflow

The required product is divided into tasks that will be accomplished in several iterations called “sprints” and several ceremonies, these tasks described in “product backlog”, and prioritized by the product owner, each member of the team choose the proper task to do it. The sprints have fixed length from one-to-four week per each sprint. The product backlog comprises “sprint backlog”, each sprint backlog contains relevant tasks that have to be accomplished in one sprint. The scrum process as shown in fig. 1 [20], it starts with translating the tasks into the product backlog by the product owner. After that, the team members hold a meeting called “Sprint planning” to plan for the sprint they will work on it. They discuss the tasks and their details and how they can accomplish them and what the time these tasks will take. After that, the scrum master and the workers start working on the selected tasks, they organize themselves as suits them.

Everyday the scrum master and the workers should meet each other in “Daily Scrum” meeting to follow up the progress of their work, and to answer the following questions: “what they did yesterday, what are they doing today, and what they will do tomorrow?”

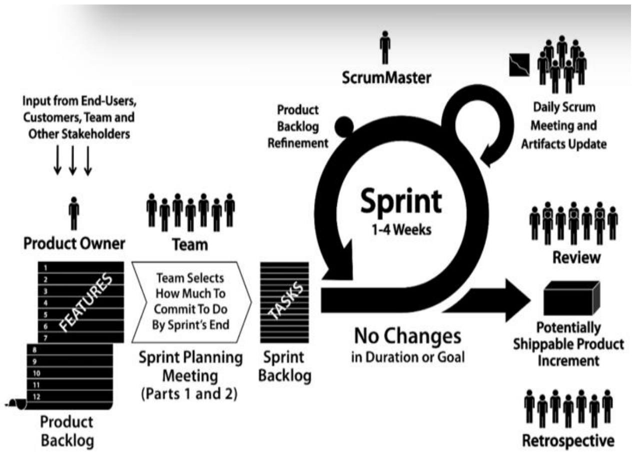
ScrumMaster
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Sprint
1-4 Weeks
Product Owner
Team
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes
in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting (Parts 1 and 2)
Sprint Backlog
Product Backlog
Retrospective

Fig.1. The scrum process

After accomplishing the tasks of the sprint the scrum master and the workers meet the customer and the product owner in a meeting called “sprint review” to gather the feedback about the produced features, if there were a comments about the features from the customer the team fix it as required then deliver it to the product owner to deliver it to the customer. After that, the scrum master and the workers hold a meeting called” sprint retrospective” to discuss what was good in the previous sprint and what needs to be done in a better way in the next sprint. By the end of each sprint, there are features ready to be delivered to the customer after the evaluation process [17].

To emphasize the efficiency of scrum, Chris and Frank [21] and Hu Guang [5] conducted studies that confirmed the efficiency of scrum methodology. Chris and Frank [21] conducted a case study and the result indicated that implementing scrum in a development process decreased the developer overtime at the same time increased the customer satisfaction, where Hu Guang [5] also conducted study that confirmed the success of scrum in the field of software development, this study imported scrum in the development process for parts of vehicle spare by using visual studio as a template for scrum management process. The study confirmed that by implementing scrum the productivity and customer satisfaction increased and the product accomplished with the desired quality at the required time [5]. Several studies showed that by implementing scrum the customer satisfaction will be increased and at the same time the productivity will be duplicated as the following study conclude. Jeff et al. [22] demonstrated how the scrum model duplicated the productivity at the same time reduced the work time.

C. Lean Software Development:

Lean software development (LSD) model is a toolkit of agile that aimed to raise the value of customer by minimizing the resources that can be used on activities that will not add any value for the customer’s product which that in turn, leads to minimize the lifetime of project, also concentrate to achieve the following: prevent and detect errors to avoid producing defects, automating the process of manufacturing partially and stop if there are any defects automatically, customer demands (Just In Time (JIT)-Pull system), and the continuously of improvement [23].

Lean software development has seven principles as the following: “Eliminate waste”, “Amplify learning”, “Decides as late as possible”, “Deliver fast as possible”, “Empowering the team”, “Build integrity in”, “See the whole” all of them can be applied in the field of software development, These principles have been mentioned in details by Poppendiecks [24] and Laura et al. [9].

LSD model has been evaluated through using guidelines of project management by conducting interviews and surveys with 17 persons all of them are professionals where 8 of them are corporate directors and the rest of them are PhDs and have expertise in this field. The aim of this work was to identify which the area of project management-as the guidelines of PMBoK has greatest crucial processes that associated with lean implementation. The processes in lean implementation that have been found as crucial processes and have to be improved were in the management of Purchases, communication, risks, the stakeholders, and human resources. The collected data have been analyzed by using “exploratory factor analysis”, “Multi-Dimensional Scaling”, and statistical methods.

Lean principles and practices are applied in various manufacturing fields such as manufacturing of air-conditioning coil, electronic, steel and automotive production [25-28] led to raise the productivity, reduce the wastes which in turn, lead to minimize lifetime of the project, and facilitate the continuation of improvement.

D. Kanban

Six years ago lean proposed a key project management tool and applied it to the software industry called

“Kanban”, it is Japanese expression that means “signboard” which is as defined by Marko et al. [12] “It is basically a flow control mechanism for pull-driven Just-In-Time production in which the upstream processing activities are triggered by the downstream process demand signals”. It facilitates improving, monitoring, and managing the workflow of the production by visualizing and representing the workflow on a board known as “Kanban board” or by sticky notes that known as “Kanban Cards” which are sticky notes on a whiteboard. The Kanban board shows progress of the work as shown in fig. 2 [29]. It is divided into columns those describe the progress of work, the ideal columns name that usually used are as the following: Tasks, In Progress, Code Review, Accept, Testing, and Done or TO DO, DOING, and DONE. The aim of this board is to help the team members to work effectively and limiting the work in progress (WIP), to avoid the bottleneck or remove it if it exists, the customer value will be maximized, the waste will be reduced. In addition, the kanban board can facilitate measuring the average time for accomplishing one feature and therefore, measuring the whole project time.

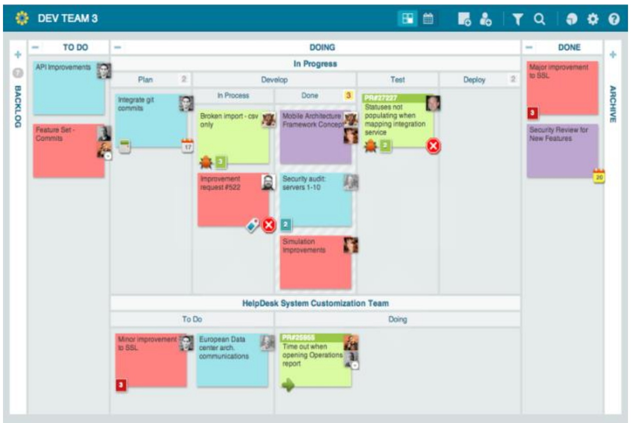

Fig.2. Kanban board

The set of features need to be accomplished are chosen and placed on the board, each column in the board has a limit of work cannot be bypassed, if there is a lot of unfinished works in one column then the team members have to collaborate and finish these unfinished works before starting new works.

For confirming the positive effect of applying Kanban on software development project the authors in [11] investigated the impact of Kanban on the workflow of software development project by proposing framework that based on previous literature, this framework perceives 9 aspects of work of project as the following : documentation, problem-solving, visualization, understanding the whole, communication, embracing the method, feedback, approval process, and selecting work assignments where all these aspects have been investigated to verify if Kanban model can influence these aspects considerably, then this framework has been assessed by conducting an experimental investigation on software factory. The results illustrated that Kanban influences the nine viewpoints as described as follows: for documentation viewpoint regarding Kanban model only documenting what is producing the value; because the customers didn’t interest to read the document, regarding problem solving viewpoint the problems solved as they occurred where if there is any task with high priority did not finish yet the team can cooperate to accomplish it as soon as possible, for visualization viewpoint the Kanban board helped the workers to flow their works and monitor their progress, regarding understanding the whole viewpoint there is no clear influence of Kanban on this viewpoint where the workers can understand the whole by attend the demos of customer or carry out various tasks, communication was perfectly between the team and by the help of Kanban board the workers were able to visualize the problem if it existed then by communicating each other they can find solution or divide any big unsolved problem into tasks, for embracing the method viewpoint at the beginning converting the project ideas to the Kanban board will take a time and a little effort but with a little assistant and leading by the leader the Kanban board can be easily understood, for the feedback viewpoint the feedback can be obtained from demos of customers and the reviews meeting such as code review or testing, regarding the approval process viewpoint there is no higher authority to approve any task, the team made the rules for themselves to approve any tasks, where a developer with its specialty approve tasks for another developer and so on, for selecting work assignments viewpoint the tasks have been chosen based on independence and the priority of the tasks among business value and customer demands. The results also illustrated the adaptability of Kanban model in projects that related to software engineering and illustrated its benefits in managing the work in orderly and effectively manner besides coordinating the activities of the team members.

E. Combining Lean Thinking and Agile Methodologies

Combining lean thinking and agile approaches gained wide attention in recent years. Agile and lean thinking are complementing each other, combining lean thinking with agile methodologies that requiring from the organization’s managers a profound understanding about lean thinking and how it can be applied as Ladas [30] illustrated, where he provided a chain of essays that provide the necessary background to facilitate combining agile methodologies, for example: scrum with lean’s method such as kanban.

As it was mentioned above that agile approach has significant benefits including increasing the product quality and flexibility of development process besides reducing project lifetime, some agile methodologies found that cannot be scaled to the entire organization [31]. Lean is a promising tool that can enhance software development process where it concerns to eliminate the wastes which in turn, leads to decrease the project lifetime and increase the value of the customer, where this tool has the ability to scaling-up the agile methodologies. The differences between agile and lean practices have been demonstrated in detail by Perera et al.

[32] then the relationship between agile and lean has been emphasized based on examinations conducted in their lab, the collected data has been analyzed which indicated that the processes of agile software development can be improved by using principles of lean which that means a combination of agile processes and lean principles can drive the developer team to deliver a high-quality product.

Although few differences between agile and lean approaches but the combination of them can addressing different components of software development. Due to some common principles between agile and lean such as managing team, producing products with high quality, continuous improvement, …etc., lean and agile approaches can be complementing each other but with regard to some aspects such as customer priority of their needs and/or value and flexibility [33]. Poppendieck [34] stated: "Lean thinking a platform upon which to build agile software development practices" therefore, lean could be used as a platform to facilitate scaling agile. Adoption the combination of agile and lean can achieve the continuous of both learning and improvement.

The combination of agile and lean thinking, in software organizations, can be applied to develop software [33]. It is addressed that the basic factors have a significant impact on applying the combination of agile and lean. The objective of the study is seeking to identify the important tools and practices to facilitate applying the combination of agile and lean approaches in software development organizations. Empirical studies have been conducted on two organizations that were seeking to transform their processes from agile to lean approach. The results of the study confirmed that there are multiple compatibilities between agile and lean so that they can be combined in a single process. There are numerous benefits of this integration such as lean principles guided the using of agile methodologies through development process, the productivity increased, the satisfaction of customer has been met, fast response to the requests of customers. Another study [35] indicates that lean and agile combination in a single process brings into significant benefits to achieve economic goals including efficiency, quality and flexibility.

Agile and lean combinations can exist for numerous purposes in numerous manners in the field of software development, a collection of experience-reports that adopted agile and lean combinations in the real world have been analyzed by Xiaofeng [1] to help the researchers to understand these combinations. Other researcher addressing the combination of lean software development principles with the practices of a particular methodology of agile such as a combination of Scrum and lean as illustrated by Benjamin [36], where a combination of Scrum and lean model has been proposed to guide the companies of software development and production. The model has been evaluated by several developers (around a thousand) working in software companies, the evaluation results demonstrate that the proposed model achieved delivering the product in time with the desired budget and quality.

Other combinations of agile methodologies with lean have been proposed such as ExtremeBan (Extreme Programming and Kanban) [37], ScrumBan (Scrum and Kanban) [38]. Scrumban is the most common combination methodology where it is described in details in the next section.

F. Scrumban

Scrum and Kanban are "lightweight methods" considered as tools, where there are no tools are completely perfect or can guide the team all the time or in all the activities, they provide them multiple guidelines, principles, and constraints only. A sophisticated methodology can guarantees the highest product quality such as complementing the principles of scrum with practices of Kanban yields more sophisticated development methodology called ScrumBan, which can combine the benefits of both of them, scrum can effectively facilitate the managing of projects by providing principles urges the cooperation among the team to accomplish the required work that have been divided in several sprints with fixed length and the scrum's team should be "a cross-functional" where Kanban limits the work in progress, measures the lifetime of project, besides monitors and manages the workflow by applying Kanban board [39-40].

Knowledge, about how frameworks of Scrum and Kanban delivering value, is essential to combine and mix Scrum and Kanban effectively in order to design an efficient framework that meets the needs. Several organizations found their needs by combining Scrum and Kanban approaches, sometimes the development team members combine several tools during their work without being aware that they have hired principles or practices of another tool such as team of Kanban could meet each other daily which this practice belongs to scrum or the team of scrum limits the size of their activity per column (queue) which is practice of Kanban [39].

ScrumBan framework similar to Kanban in pulling the work according to the needs, continuous flow, measuring lead time, has Kanban board, has ready and open queues instead of product backlog as in scrum, limits the work in progress, prioritized the work based on demands and the class, and has diagram demonstrates the continuous flow, and like scrum in: having ceremonies such as daily meeting and retrospective meeting, the roles of the team which having product owner, scrum master, the workers and the team should have members with different specialties and expertise (cross-functional team) .

Katarzyna [38] worked with intelligent administrators of systems and tried to apply both of scrum and Kanban approaches separately in developing software projects processes, they came up with these results: the scrum approach was not good enough because the continuous changes in the environment and due to lacking the Kanban for boundaries. With these results, they decide to use the combination from both of Scrum and Kanban (ScrumBan) which achieved the required results with the high level of quality and visibility. The ScrumBan configuration has been applied is explained in details in his paper.

III. Problem Definition And The Proposed Solution

What is the motivation behind combining Lean thinking, Scrum, and Kanban to yield L-ScrumBan methodology?

A. Problem Definition

Lean principles only provide the “behavioral approach” to guarantee the success of the development process. Lean does not cover the technical and managerial issues, and all its concern is about minimizing the wastes and improving the quality. Lean, therefore, is not considered as a complete model to be implemented in software development field [33].

Scrum illustrates the details of development process using the scrum methods including all the phases and practices that can be done, but there is no consideration to managing the workflow of daily tasks [40].

Kanban is just managing tool to monitor the project workflow, there is no sufficient practices or consideration to the phases or plan that describe how the development process can be done [29]. Thus, the answer to the question above is: “No Tool Is Complete, No Tool Is Perfect” [39].

So we combine Lean thinking, Scrum methodology, and kanban tool in a complete and a perfect methodology called L-ScrumBan, this methodology overcomes the weaknesses of the previous methodologies and comprises all their strengths together.

B. The Proposed L-ScrumBan Methodology

“Don’t limit yourself to only one tool. Mix and match the tools as you need” [39].

B.1. Introduction

L-ScrumBan methodology is an agile framework for managing software development process. The term L-ScrumBan is derived from Lean thinking, Scrum methodology, Kanban tool. Moreover, L refers to Lean thinking, Scrum refers to Scrum methodology, and Ban refers to Kanban tool.

It’s inherited several features from Agile software development “self-directing and organizing team” and “increment and iterative”. L-ScrumBan methodology has a distinctive feature so that customers can indulge in the process of developing the required project, and can follow the work progress of the project on General board that is managed by the product owner.

L-ScrumBan methodology process is accomplished in iterations called sprints. The process starts with the product owner who translates the customer requirements in the customer demands list so that each requirement is prioritized according to the customer value. Then the entire team meets each other to eliminate wastes from customer demands list to produce the product backlog.

There are two kinds of boards, the general board which is for the whole project. It is managed by the product owner to follow the progress of the work (this board is visible to the entire team and the customer). The other board is L-ScrumBan board that follows the work of the sprint and is managed by both the team master and the developers (this board has to be rest for each sprint).

The product owner puts the product backlog in the general board under TO DO column, at the same time, the team starts the sprint by planning which tasks from the product backlog have to be accomplished in this sprint. Then, writing the sprint with its tasks and details in their L-ScrumBan board under Ready column, after that, the product owner writes the tasks that the team will do of the sprint with their details under DOING column in general board. The team starts working on the tasks so that the selected tasks have to be written in DOING column on their board. After accomplishing the sprint, the team tests the produced features -these features have to be written in TESTING column on their board. if the test succeeds, the team moves these features to DONE column on their board and delivers the produced features to the product owner and customer.

The product owner moves the tasks of the sprint and their details to be done as a column on a general board and gathers the feedback from the customer after testing the required features. The product owner moves this sprint and their tasks to be delivered column in general board after the features to the customer are delivered. Then the team starts the next sprint and so on.

L-ScrumBan methodology has five tools: Customer Demand List, Product Backlog, Sprint backlog, General board, and L-ScrumBan board. Three roles: Product Owner, Team Master, and The Developers. Six meetings: Eliminate-waste meeting, Sprint Planning, Daily meeting, Quality-Test meeting, Sprint Review, and Sprint Retrospective.

B.2. L-ScrumBan Tools

L-ScrumBan methodology has five tools all of them visible and can be accessed by the team. The tools as the following: Customer Demands list, Product Backlog, Sprint Backlog, General Board, and L-ScrumBan Board.

1) Customer Demands list: It comprises a queue of all the items including features, functional requirements, non-functional requirements, enhancement, or bug fixes of customer demands in user stories form that provide a brief description for each task as the customer described. These items are divided into tasks then prioritized according to customer value; where these tasks have to be independent of each other. It is the responsibility of the product owner to collect the demands from the customer.
2) Product Backlog: it’s the customer demands list after eliminating the waste, in details: after dividing and prioritizing the customer demands list (tasks) in user stories form, the Team Master and the Developers (the team) have to hold an “eliminate-waste meeting” to eliminate the waste from these tasks. Then, they have to define and

identify each one of them with a unique identifier (ID), the priority, the category (enhancement, functional requirement, non functional requirement, feature, or bug fixes), cost and time estimation to distinguish each task.

3) Sprint backlog: it contains all the tasks and their details, the cost and time estimation for each requirement that has to be accomplished during one sprint. This sprint backlog is selected from the product backlog by the team in a meeting called sprint planning. The tasks with high priority have to be accomplished first.

4) General board: it is a physical (board with sticky notes) or digital board as shown in Fig. 3 below, managed by product owner to follow the work progress of the whole project. The customer, the product owner, the team Master, and the Developers can see this board to check the progress of the work. The board has four columns: TO DO (containing product backlog), DOING (containing the tasks the team working on it, this column has only the tasks of one sprint at a time), DONE (containing the accomplished tasks of the sprint that ready to be delivered to the customer), and DELIVERED (containing the tasks that are delivered to the customer).

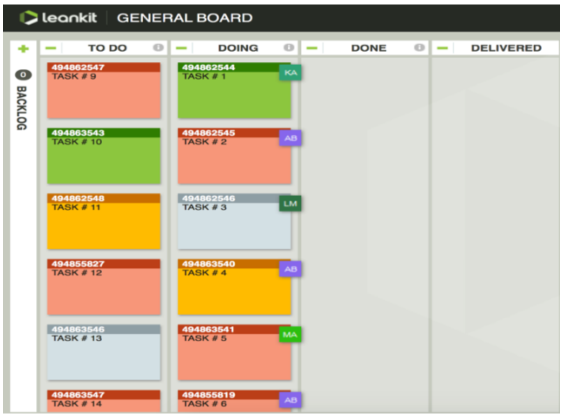

Fig.3. General Board

5) L-ScrumBan board: it is a physical (board with sticky notes) or digital board as shown in Fig. 4 below, managed by team master and the developers to follow the work of each sprint, after each sprint this board has to be reset. The ideal columns for the board are READY (containing sprint backlog), DOING (containing the tasks of the sprint the team works on it- this column may divide into several columns and several rows. The columns are designing and coding as needed, and each column should limit the number of the tasks (WIP) that can be accomplished at the same time). Also, for each column, there are several rows containing the name of developer with the tasks he/she will do it (each developer must have limited number of tasks he/she cannot exceed this limit). The next column is TESTING (each task has to be tested first by a test called Quality test before hold meeting with the customer to guarantee the quality and integrity of the delivered feature), this column is mandatory to guarantee the quality and integrity of the project. Finally, DONE column (containing the accomplished tasks after a successful testing that is ready to be delivered to the customer).

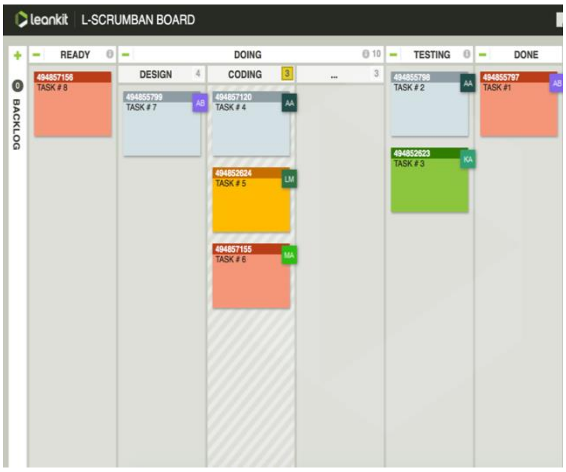

Fig.4. L-ScrumBan Board

B.3. L-ScrumBan Roles

L-ScrumBan team involves all the members that aim to accomplish one purpose, which is the customer product, and the team is divided into product owner, team master, and the developers. The ideal number of L-ScrumBan methodology team is from four people to nine.

1) Product Owner: its responsibility is to guarantee success of the required project and to meet the customer, then to translate the requirements of the required project in customer demands list, after that, to provide the appropriate priority for each requirement according to the customer value- where the requirement with high value should be on the top. The product owner should facilitate communications between customers and the team if needed.

2) Team Master: is a team leader towards achieving the purpose. Its responsibility is to assist the team by making sure they are working on the right way as planned, and by removing any difficulties that can face them during work. The team master has no authority over the team, he/she is just one of them working and participating in developing the product. The team organizes and manages itself with a very high degree of both autonomy and accountability.

3) The developers are the members who are working on product development process. They are the core of the L-ScrumBan team and they should be "cross-functional team". There is no authority over the developers, they have the authority to do their work in their proper way that suits them to achieve the purpose. The team can manage and organize themselves as appropriate for developing the product.

B.4. L-ScrumBan Meetings

L-ScrumBan Methodology has six types of meetings: Eliminate-waste meeting, sprint planning, daily meeting, quality-test meeting, sprint review, and sprint retrospective. The software development process will become easier and more manageable by holding these meetings.

1) Eliminate-waste meeting: this meeting has to be held after the product owner has met with the customer to make the customer demands list, and before starting any work on the product. The product owner, team master, and the developers have to attend this meeting. The purpose of the meeting is to remove any waste from customer demands list that will not add any value to the customer; to save time, effort and cost on something do not add any value. The result of this meeting is Product Backlog.
2) Sprint Planning: is a meeting has to be held after producing the product backlog to discuss and choose the appropriate tasks that will be accomplished during the coming sprint. They have to discuss the tasks details and estimate the sprint time. All L-ScrumBan team members should attend this meeting to share their suggestions and opinions about how to do the work. This meeting has to be held before starting any new sprint.
3) Daily meeting: it is fifteen minutes meeting (“stand-up meeting”), the team master and the developers attend this daily meeting to discuss and follow their work and to make sure the work is proceeding as planned by answering these questions: what did they work yesterday? what are they working on today? what are they going to do tomorrow?
4) Quality-Test meeting: it is a meeting after accomplishing the sprint. The team master and the developers attend it to test the feature that the team has produced from this sprint before delivering it to the product owner and the customer. It is a quality test to guarantee the quality and integrity of the produced feature. This meeting has to be held after accomplishing any sprint.
5) Sprint Review: it is a meeting concludes all the team members including the product owner with the customer; to deliver the produced features to the customer and to gather the feedback from him/her about the produced features so that the customer can test its feature, which is called customer test, then provide the team with his/her feedback. This meeting takes its place after the team has done the quality-test meeting.
6) Sprint Retrospective: this meeting happens after finishing the sprint and before starting a new sprint. The aim of this meeting is to discuss how was their work in the previous sprint, what were the mistakes they have done, and how can they overcome such mistakes in the next sprint. It’s a meeting for the team master and the developers to discuss their work enhancements as well.

B.5. L-ScrumBan Process:

The process of L-ScrumBan methodology starts as shown in Fig. 5 below, with the product owner. The product owner meets the customer, and writes down all the requirements of the customer about the desired product; this meeting could in be several days until all the requirements of the customer become clear. The product owner writes down all the requirements, but no decisions are taken until the last moment of his/her meeting with the customer, which is known in Lean thinking as “Decide as late as possible”.

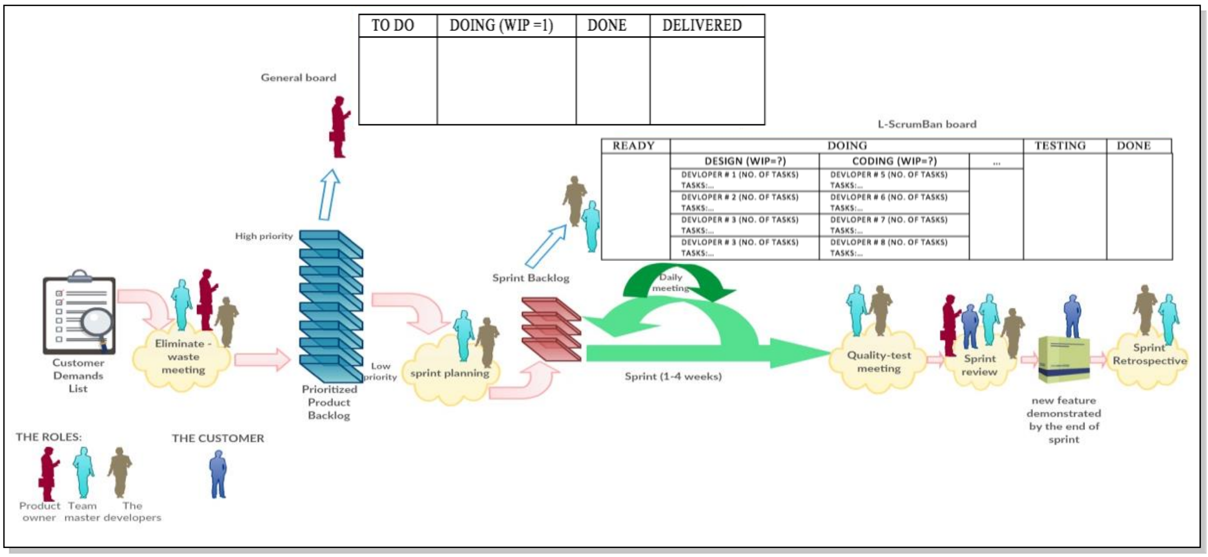
TO DO
DOING (WIP =1)
DONE
DELIVERED
General board
L-ScrumBan board
READY
DOING
TESTING
DONE
DESIGN (WIP=?)
CODING (WIP=?)
...
DEVLOPER # 1 (NO. OF TASKS) TASKS:...
DEVLOPER # 5 (NO. OF TASKS) TASKS:...
DEVLOPER # 2 (NO. OF TASKS) TASKS:...
DEVLOPER # 6 (NO. OF TASKS) TASKS:...
DEVLOPER # 3 (NO. OF TASKS) TASKS:...
DEVLOPER # 7 (NO. OF TASKS) TASKS:...
DEVLOPER # 3 (NO. OF TASKS) TASKS:...
DEVLOPER # 8 (NO. OF TASKS) TASKS:...
High priority
Sprint Backlog
Daily meeting
Customer Demands List
Eliminate - waste meeting
Low priority
Prioritized Product Backlog
sprint planning
Sprint (1-4 weeks)
Quality-test meeting
Sprint review
Sprint Retrospective
new feature demonstrated by the end of sprint
THE ROLES:
THE CUSTOMER
Product owner
Team master
The developers

Fig.5. L-ScrumBan process

After that, the product owner makes the customer demands list and provides the priority for each requirement according to the customer value. The product owner holds a meeting “Eliminate-waste meeting” with the team (team master and the developers). In this meeting, they look for the wastes in the customer demands list and discuss how to eliminate them and how to keep their work clean from any waste, the aim of this step is to save their time and effort from doing something that does not add any value for the customer. There are several types of waste as the following:

- Producing extra-features the customer does not a concern with it or producing two features that have the same functionality.
- Any extra-process that does not add a value for the customer is considered a waste. For example, the paperwork does not add a value for the customer, so documenting only what is really matter for the customer.
- Any defects produced during the work of the team it is considered a waste, so the team has to guarantee the quality and integrity of their work by making a test for each sprint in “quality-test meeting”.

After the team has eliminated the waste, the team produces the “Product Backlog” in form of user stories (tasks) and provides points and cost for each task. This product backlog is prioritized by the product owner, and then the team has to identify each one of the requirements with a unique identifier (ID), and to write the category for each one of them (enhancement, functional requirement, non functional requirement, feature, or bug fixes). After that, the product owner writes the product backlog with the details in the column titled with TO DO in the general board, managed by him/her, to follow the work progress of the team. Now the product backlog is ready, the team holds a meeting called sprint planning to discuss the sprint they will start, tasks that have to be accomplished in this sprint, and to estimate the cost and time of this sprint. After that, the product owner puts the tasks and details of the sprint that has been chosen from the product backlog in the column titled with DOING in the general board; this column has only the tasks of one sprint at a time. At the same time, the master team and the developers start the sprint by filling the L-ScrumBan board with the details of the sprint they will do it in the column titled with READY. Then, the team organizes themselves to do the selected tasks and to fills out the DOING column in L-ScrumBan board with the chosen tasks under the proper sub-column of the column DOING. Daily, the team has to attend the daily meeting to discuss their progress about work. During the teamwork, if there is any delay that has been noticed on the board, the team members contribute to overcoming this delay to accomplish work as planned then to go back again to complete their specific assignments. When the sprint is accomplished, the team puts the accomplished features of the sprint in TESTING column of L-ScrumBan board, and holds “quality-test meeting” to test the produced features before their meeting with the customer and product owner- to guarantee the integrity and the quality which is called in Lean thinking “Build the Integrity in”. If there was any bug, it has to be fixed. If there were no any bugs, then they move the accomplished features of the sprint to DONE column of L-ScrumBan board. After that, the team holds a meeting called “Sprint Review” with the product owner and the customer- to deliver the produced features to the customer. At this time, the product owner has to move the accomplished sprint and their accomplished tasks from DOING column to DONE column on the general board, and the customer can make the test on these features to make sure this is the required. The team gathers the feedback from this meeting and if there were any objection from the customer, the team has to fix it before moving to the next sprint. If the produced features satisfy the customer, the product owner delivers these features to him/her and moves the features from DONE column of the general board to Delivered column. After delivering the features that have been produced during the sprint, the team (master team and developers) holds the last meeting for this sprint called” Sprint Retrospective” to discuss how was their work in the sprint (what was good and what was bad?), and how can they improve their work in the next sprint?

IV. Validation Of The Proposed Solution

The validation of the proposed methodology has been done through a survey as a research methodology. The validation is done by using a questionnaire; the questionnaire was designed to evaluate L-ScrumBan methodology against five goals. It was distributed among eight software development companies who are involved in Agile methodologies. The questionnaire is divided according to five goals:

Goal 1: Agile methodologies can improve the software development process and can achieve the overall goals of the software development and business field.

Goal 2: Scrum is the most appropriate methodology for software development process but it’s insufficient regarding the management of daily tasks and visualizing the workflow.

Goal 3: Lean principles can help in the software development process but it’s insufficient regarding the technical and managerial issues, and therefore is considered as an incomplete model.

Goal 4: Kanban is helpful but it’s not enough to work alone because it’s limited to being a management tool for visualizing the work nothing more.

Goal 5: The sufficiently of L-ScrumBan methodology by offering practices, principles, management tools that can guide the team through the whole lifetime of the project.

A. Cumulative Statistical Analysis of Goal 1

The goal is to determine that how much agile methodologies can improve the software development process and achieve the overall goals of the software development and business field. As represented in Table 1 that 90% of the respondents were very highly agreed that agile methodologies have a very high effect on improving the software development process and achieve the overall goals of the software development and business field. It is also, represented that 8% of the participants were highly agreed that agile methodologies have a high effect on improving the software development process to achieve the overall goals of the software development and business field. 2% of the participants report the nominal effect of agile methodologies on improving the software development process and achieving the overall goals of the software development and business field. None of the respondents, however, believed that the agile methodologies had a low or very low effect on improving the software development process and achieving the overall goals of the software development and business field.

Table 1. Cumulative Statistical Analysis of Goal 1

Q. No.	Very Low	Low	Nominal	High	Very High
1	1%	1%	1%	9%	89%
2	1%	1%	1%	6%	92%
3	0%	0%	2%	9%	89%
Total	1%	1%	5%	23%	270%
Avg.	0%	0%	2%	8%	90%

B. *Cumulative Statistical Analysis of Goal 2*

Determine the insufficiency of scrum in two issues: the management of daily tasks and visualizing the workflow although this methodology considered as the most appropriate methodology for software development process. As represented in Table 2 that 73% of the participants were very highly agreed that scrum is insufficient with regard to the management of daily tasks and visualizing the workflow, where 8% of them were highly agreed with the respective goal. The table also, represented that 2% of the participants believed that scrum has the nominal effect regarding the management of daily tasks and visualizing the workflow, where 3% of them were agreed to the low effect and 14% of them to the very low effect.

Table 2. Cumulative Statistical Analysis of Goal 2

Q. No.	Very Low	Low	Nominal	High	Very High
4	1%	0%	2%	10%	88%
5	1%	0%	1%	9%	89%
6	1%	1%	1%	9%	89%
7	1%	0%	3%	11%	85%
8	0%	1%	4%	9%	86%
9	82%	15%	1%	1%	2%
Total	85%	17%	11%	48%	440%
Avg.	14%	3%	2%	8%	73%

C. *Cumulative Statistical Analysis of Goal 3*

Goal 3 emphasizes that Lean principles can help in the software development process but it's insufficient regarding the technical and managerial issues, and therefore is considered as an incomplete model. As represented in Table 3 that 88% of the participants are very highly agree that Lean principles is insufficient with regard to the technical and managerial issues, and therefore is considered as incomplete model, where 9% of them were highly agreed with this goal and 1% of them favoring the nominal effect of goal 3. 2% of the respondents emphasize that Lean principles have very low efficiency regarding the technical and managerial issues and none of them emphasize the low effect of this goal.

Table 3. Cumulative Statistical Analysis of Goal 3

Q. No.	Very Low	Low	Nominal	High	Very High
10	1%	0%	2%	9%	89%
11	3%	0%	1%	10%	86%
Total	4%	0%	3%	19%	175%
Avg.	2%	0%	1%	9%	88%

D. *Cumulative Statistical Analysis of Goal 4*

Goal 4 emphasizes that Kanban is helpful but it's not enough to work alone because it's limited to being a management tool for visualizing the work nothing more. As represented in Table 4 that 89% of the respondents are emphasized that Kanban is limited to being a management tool for visualizing the work nothing more.

Table 4. Cumulative Statistical Analysis of Goal 4

Q. No.	Very Low	Low	Nominal	High	Very High
12	1%	0%	2%	9%	89%
13	1%	1%	1%	9%	89%
Total	1%	1%	3%	18%	178%
Avg.	1%	0%	1%	9%	89%

9% of the participants were highly agreed with this goal, where 1% of them were agreed with the nominal effect of the respective goal. None of the participants agreed with the low effect, where 1% of them were agreed to the very low effect.

E. *Cumulative Statistical Analysis of Goal 5*

Goal 5 determines the efficiency of L-ScrumBan methodology by offering practices, principles, managing tools that can guide the team through the whole lifetime of the project. As represented in Table 5 that 89% of the participants were very highly agreed that L-ScrumBan methodology is highly efficient by offering practices, principles, managing tools to guide the team through the whole lifetime of the project, where 8% of them were highly agreed with this goal. Only 2% of the participants believed that L-ScrumBan methodology has the nominal effect, where none of the participants, however, believed that the L-ScrumBan methodology has efficiency with low or very low effect.

Table 5. Cumulative Statistical Analysis of Goal 5

Q. No.	Very Low	Low	Nominal	High	Very High
14	0%	0%	2%	5%	93%
15	1%	0%	2%	13%	85%
16	1%	0%	2%	9%	89%
17	0%	1%	2%	9%	89%
18	0%	0%	2%	5%	93%
19	0%	0%	2%	13%	85%
20	1%	1%	1%	7%	91%
21	1%	0%	2%	7%	90%
22	1%	1%	4%	9%	85%
23	1%	0%	3%	10%	86%
24	1%	0%	2%	5%	92%
Total	5%	2%	24%	92%	978%
Avg.	0%	0%	2%	8%	89%

F. *Final Cumulative Evaluation of all Goals*

Final cumulative statistical analysis results of the five goals as shown in Table 6 represent that only 4% of the respondents report that the five goals have a very low effect to improve the software development process, where 1% of them have agreed that the goals have the low effect. 2% of the participants were neutral. 8% of the respondents report that the five goals have the high effect to improve the software development process, where the rest of them which are 85% emphasize that five goals have a very high effect. Therefore, 93% of the respondents support that L-ScrumBan methodology has the ability to improve the efficiency of the software development process. The graph in fig. 6 shows the results of all the goals.

Table 6. Final Cumulative evaluation of all Goals

Q. No.	Very Low	Low	Nominal	High	Very High
1	1%	1%	1%	9%	89%
2	1%	1%	1%	6%	92%
3	0%	0%	2%	9%	89%
4	1%	0%	2%	10%	88%
5	1%	0%	1%	9%	89%
6	1%	1%	1%	9%	89%
7	1%	0%	3%	11%	85%
8	0%	1%	4%	9%	86%
9	82%	15%	1%	1%	2%
10	1%	0%	2%	9%	89%
11	3%	0%	1%	10%	86%
12	1%	0%	2%	9%	89%
13	1%	1%	1%	9%	89%
14	0%	0%	2%	5%	93%
15	1%	0%	2%	13%	85%
16	1%	0%	2%	9%	89%
17	0%	1%	2%	9%	89%
18	0%	0%	2%	5%	93%
19	0%	0%	2%	13%	85%
20	1%	1%	1%	7%	91%
21	1%	0%	2%	7%	90%
22	1%	1%	4%	9%	85%
23	1%	0%	3%	10%	86%
24	1%	0%	2%	5%	92%
Avg.	4%	1%	2%	8%	85%

V. CONCLUSION

In this paper, an integrated methodology that integrating Scrum, Kanban, and Lean methodologies to yield a comprehensive agile software methodology called L-ScrumBan has been proposed. The methodology is proposed to improve the efficiency of agile software development and to overcome the weaknesses of the previous methodologies and comprises all their strengths together. The validation of the proposed methodology has been done through a survey as a research methodology. The validation is done by using a questionnaire; the questionnaire was designed to evaluate the proposed L-ScrumBan methodology against five goals. The survey results confirmed the efficiency of the proposed methodology in the software development field and its capability to overcome the weaknesses of the previous methodologies, besides its ability to guide the team during the whole lifetime of the project. The results of the questionnaire are represented using bar charts and frequency tables. A support of 93% is achieved to emphasize the efficiency of the proposed L-ScrumBan methodology as shown in fig. 6 and Table 6. It is emphasizing that the L-ScrumBan methodology has the ability to improve the efficiency of the software development process and achieve the overall goals of the software development and business field.

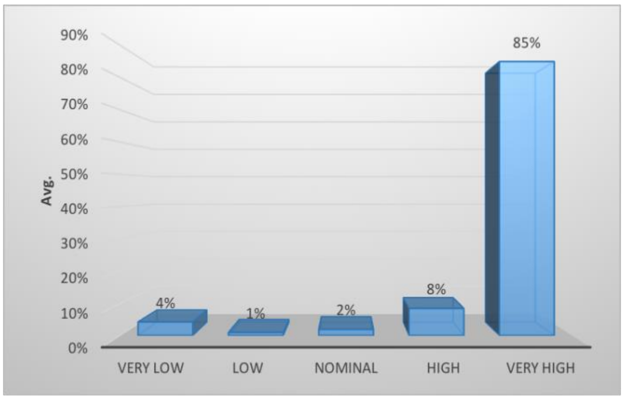

Fig.6. Cumulative results of all the goals

ACKNOWLEDGEMENTS

Sincerely thanks my academic supervisor Dr. M. Rizwan Jameel Qureshi for his invaluable support that helps me finish my thesis and I thank all the participators those who cooperated with me in answering the survey.

REFERENCES

[1] Xiaofeng, "The Combination of Agile and Lean in Software Development: An Experience Report Analysis", *IEEE*, Vol. 36, pp 1–9, 2011.

[2] Devedžić, Vladan, and Saša R Milenković, "Teaching Agile Software Development: A Case Study", *IEEE Transactions on Education,* Vol. 54, No. 2, pp273–78, 2011.

[3] Kohlbacher, Markus, Ernst Stelzmann, and Sabine Maierhofer, "Do Agile Software Development Practices Increase Customer Satisfaction in Systems Engineering

Projects?", *In Systems Conference (SysCon), 2011 IEEE International,* pp168–72, 2011.

[4] Nuevo, Eva del, Mario Piattini, and Francisco J. Pino, "Scrum-Based Methodology for Distributed Software Development", *IEEE*, Vol.23, pp66–74, 2011.

[5] Guang-yong, "Study and Practice of Import Scrum Agile Software Development", *In Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference,* pp. 217–220, 2011.

[6] Scharff, Christelle, S.Heng, and V.Kulkarni, "On the Difficulties for Students to Adhere to Scrum on Global Software Development Projects: Preliminary Results", *In Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD), IEEE,* pp. 25–29, 2012.

[7] Potineni, Sindhura, S.Bansal, and A.Amresh, "ScrumTutor: A Web-Based Interactive Tutorial for Scrum Software Development", *In Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on IEEE*, pp. 1884–90, 2013.

[8] Bougroun, Zeaaraoui, and Bouchentouf, "The projection of the specific practices of the third level of CMMI model in agile methods: Scrum, XP, and Kanban", *in information science and technology(CIST) on IEEE,* pp.174-179, 2014.

[9] Bocock, Laura, and A.Martin, "There's Something about Lean: A Case Study", *IEEE*, pp. 10–19, 2011.

[10] Andrea, "A Guide to Lean Software Development in Action," *In Software Testing, Verification and Validation Workshops (ICSTW), 2015 IEEE Eighth International Conference on IEEE,* pp. 1–2, 2015.

[11] Marko, Pirinen, Fagerholm, Kettunen, and Abrahamsson, "On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation", *IEEE*, Vol.37, pp. 305–314, 2011.

[12] Ikonen, Marko, Kettunen, Oza, and Abrahamsson, "Exploring the Sources of Waste in Kanban Software Development Projects," *IEEE*, pp376–381, 2010.

[13] S. Ashraf and Sh. Aftab, "Scrum with the Spices of Agile Family: A Systematic Mapping", M*odern education and computer science (MECS)*, Vol. 9, No.11, pp58-72, 2017.

[14] Martin, Robert C., and M.Martin, "Agile Principles, Patterns, and Practices in C#, " *Prentice Hall*, 2007.

[15] Livermore, Jeffrey, and others, "Factors That Impact Implementing an Agile Software Development Methodology", *IEEE*, pp82–86, 2007.

[16] Moniruzzaman, A. B. M., and Dr S.Hossain, "Comparative Study on Agile Software Development Methodologies", *Global Journal of Computer Science and Technology (c),* Vol. 13 Issue 7 Version I, 2013.

[17] The Scruminc website. Available: https://www.scruminc.com, online January 2017.

[18] Williams, "A survey of agile development methodologies," 2007.

[19] Almutairi and Qureshi R., "The Proposal of Scaling the Roles in Scrum of Scrums for Distributed Large Projects", *In modern education and computer science (MECS)*, Vol. 7 No. 8, pp. 68-74, 2015.

[20] S. Ashraf and Sh. Aftab, "IScrum: An Improved Scrum Process Model", *In modern education and computer science (MECS),* Vol.9 No.8, pp16-24, 2017.

[21] Mann, Chris, and Maurer, "A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction," *In IEEE*, pp. 70–79, 2005.

[22] Sutherland, Jeff, and I.Altman, "Organizational Transformation with Scrum: How a Venture Capital Group Gets Twice as Much Done with Half the Work", *In System Sciences (HICSS), 2010 43rd Hawaii International Conference on IEEE,* pp. 1–9, 2010.

[23] The Lean Production website. Available: http://www.leanproduction.com/lean-glossary.html , online December 2016.

[24] Poppendieck, Mary, and Tom Poppendieck, "Lean Software Development: An Agile Toolkit: An Agile Toolkit", *Addison-Wesley,* 2003.

[25] Biman, Venkatadri, and Pandey, "Applying Lean Manufacturing System to Improving Productivity of Airconditioning Coil Manufacturing", *The International Journal of Advanced Manufacturing Technology*. Vol. 71, No. 1–4, pp. 307–23, 2014.

[26] Chan, F.T.S., H.C.W. Lau, R.W.L. Ip, H.K. Chan, and S. Kong, "Implementation of Total Productive Maintenance: A Case Study." *International Journal of Production Economics 95, no. 1*, pp71–94, 2005.

[27] Abdulmalek, Fawaz A., and J.Rajgopal, "Analyzing the Benefits of Lean Manufacturing and Value Stream Mapping via Simulation: A Process Sector Case Study," *International Journal of Production EconomicsI,* Vol. 107, No. 1, pp. 223–236, 2007.

[28] Krishnan. Pv, V.Ramnath.B, M.Pillai, "Work In Process Optimization Through Lean Manufactring," *KRISHNAN. PV. B et.al., Int. J. Eco. Res.*, Vol. 2 No. 2, pp19-25, 2011.

[29] The leankit website. [Online]. Available: http://leankit.com/kanban/online-kanban-board/.

[30] C. Ladas, "*Scrumban-essays on kanban systems for lean software development*", Modus Cooperandi Press Publisher, 2009.

[31] Abrahamsson, Pekka, and N.Oza, "Lean Enterprise Software and Systems," *First International Conference, LESS 2010, Helsinki, Finland, Proceedings. Lecture Notes in Business Information Processing, Springer,* 2010.

[32] Perera, GIUS, and M. S. D. Fernando, "Enhanced Agile Software Development—hybrid Paradigm with LEAN Practice," *In Industrial and Information Systems, International Conference on IEEE*, pp. 239–44, 2007.

[33] Pilar, "Combining Lean Thinking and Agile Software Development", *University of Oulu, Finland,* 2013.

[34] Poppendieck and Poppendieck, "Implementing Lean software development" , Software Design, Testing & Engineering, Software Development, 1st edition, Addison-Wesley Professional Publisher, pp. 19-42, 2006.

[35] Pilar, Markkula, Oivo, and Garbajosa, "Analyzing the Drivers of the Combination of Lean and Agile in Software Development Companies", *In Product-Focused Software Process Improvement, Springer*, pp. 145–59, 2012.

[36] Blau, Benjamin, and Hildenbrand, "Product Line Engineering in Large-Scale Lean and Agile Software Product Development Environments - Towards a Hybrid Approach to Decentral Control and Managed Reuse", *Sixth International Conference on Availability, Reliability and Security on IEEE,* pp. 404–408, 2011.

[37] Han, and Xie, "Practical Experience: Adopt Agile Methodology Combined With Kanban For Virtual Reality Development", *Software Engineering and Management Thesis, University of Gothenburg, Goteborg, Sweden,* 2012.

[38] Terlecka, "Combining Kanban and Scrum -- Lessons from a Team of Sysadmins", *Agile Conference on IEEE*, pp. 99–102, 2012.

[39] Kniberg, and Skarin, "Kanban and Scrum: Making the Most of Both". C4Media, Publisher of InfoQ.com.

[40] Nikitina, Mattsson, and Strale, "From Scrum to Scrumban: A Case Study of a Process Transition", In Proceedings of

the International Conference on Software and System Process on IEEE, pp. 140–149, 2012.

Authors' Profiles

Aysha Abdullah Albarqi Master student of Information Technology in King Abdul-Aziz University.

Dr. M. Rizwan Jameel Qureshi received his Ph.D. degree in Computer Science from National College of Business Administration & Economics, Pakistan 2009. He is currently working as an Associate Professor in the Department of IT, King Abdulaziz University, Jeddah, Saudi Arabia. This author is the best researcher awardees from the Department of Information Technology, King Abdulaziz University in 2013 and 2016. He is also honoured as the best researcher from the Department of Computer Science, COMSATS Institute of Information Technology, Pakistan in 2008.

How to cite this paper: Aysha Abdullah Albarqi, Rizwan Qureshi," The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.10, No.3, pp. 23-35, 2018. DOI: 10.5815/ijieeb.2018.03.04