Suez Canal University

Faculty of Faculty of Computers and

Discuss the difference between Scrum and Kanban, "in detail". Illustrate your answer with an example.

Student name	Amir Haytham Muhammad Salama
Faculty	Faculty of Computers and Informatics
Level	Third Level
Department	Computer Science
National ID	29906261900473
Student code	1714103022
Program	Computer Science Program
Course	Software Engineering

1. Introduction

Agile software development methodologies gaining the attention in the software engineering field. There are several agile methods like Scrum, Lean, and Kanban. Scrum methodology divides the product into sprints sequences. Lean is an agile toolkit that has seven concepts to facilitate: reducing waste, delivering efficiently, and improving end-customer value. Kanban is a visual approach which can help with the production management. In order to take advantage of the following methodologies: Lean, Scrum, and Kanban, we can therefore integrate them together, resulting in a new methodology that can contribute in enhancing and improving the efficiency of the software development process, which is the aim of this research. An integrated methodology has been proposed which integrates Scrum, Kanban, and Lean methodologies to create 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. Studies show that over 60 percent of software projects end in failure. [1:4]

Agile Development is one of the best way to improve the chances of your project being a success. In this research, we will take a look at the evolution of software development, and the theories and values behind Agile Software Development. We take a quick look at Kanban, and we take a deep dive into Scrum, one of the most popular Agile frameworks. We will look at the roles, events, artifacts, and values of Scrum in detail, and we will learn about the product owner role and responsibilities in Scrum. If you want to know how Agile development and Scrum can turbo charge the performance of your team, let's discuss that in detail. We live in an age We live in an age where companies work in an incredible world of unprecedented technological advancement and where consumers have more options than ever. Businesses want to deliver benefits to customers better and faster than their competitors, but they will face many challenges in the process: global competition, an evolving business ecosystem, rapidly-changing market conditions. Why do we create a product that consumers enjoy while helping an organization achieve its targets? This requires to be able to cope with a rapidly changing world, adapting to changing customer needs, experimenting, and learning quickly. Agile Methodologies has become a common option in product creation for project management frameworks in product development. [5]

Agile software development (ASD) has been around in the software industry for a decade. Agile methodologies have gained the attention in the field of software development recognition where they are targeted to: handle the dynamic requirements, meet the customer satisfaction, and deliver the product in a way that suits the customer demands with the required quality, which in turn lead to achieve the aim of the software industry. In the field of software engineering, agile methodologies like Scrum, Lean and Kanban have an effective effect , especially in software development. 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 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.[4]

2. Research items

- Agile Software Development
- The concepts and values behind Agile development
- Differentiation between 2 kinds of Agile framework: SCRUM, Kanban.
- The theory and values behind SCRUM.
- The SCRUM members, artifacts and events.
- Case Study Example to Explain the best or the proper Agile framework.

3. Research

- ## Agile Software Development [1:7]

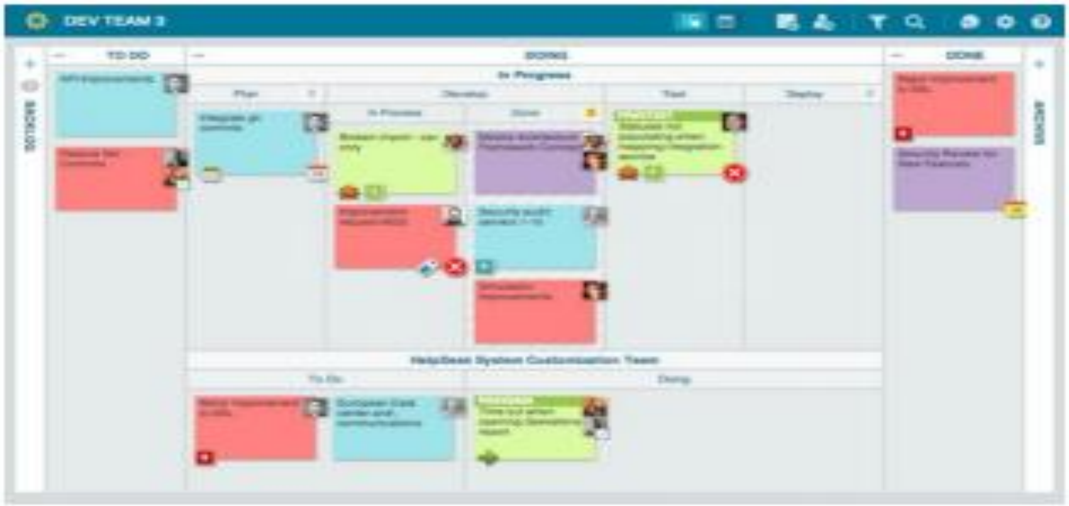
In case of the dynamically requirements that have to meet changing in environments, delivering the software fast as possible with a high quality in a short time besides the customer satisfaction, the industry field need methods that can achieve the desired service at its best, by implementing agile software development methodologies all the software development demands are achievable [10].

Agile has several methodologies for software development where each methodology has several principles and practices that distinguish it from the other, and an 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. Also, 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 [7]

- # Differentiation between 2 kinds of Agile framework: SCRUM, Kanban: [1,5,4]

1. 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 defined as It is essentially a pull-driven, and just in time flow control system which activates the upstream processing operation via 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 the following figure:



It is divided into columns describing work progress, the ideal column name usually used is as follows: Tasks, In Progress, Code Review, Accept, Test, and Done or TO DO, DOING and DONE. The purpose of this board is to help team members work effectively and to limit the work in progress (WIP), In order to eliminate a bottleneck or minimize it if it exists, the consumer benefit is maximized and the waste can. Alternatively, the kanban board can allow the calculation of the average time taken to accomplish one function and thus calculate the entire time of the project. The set of features to complete are selected and put on the board, each column in the board has a limit of work can not 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 beginning new works. To confirm the positive effect of applying Kanban to software development project, we need to look at Kanban's effects on the software development project workflow by proposing a framework that, based on previous literature, perceives 9 aspects of project work 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 [6] 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 System perspective initially transferring the ideas of the project 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. [2]

2. SCRUM: [4,6,7]

Scrum is an agile software development and project management framework that inherited ASD's features of increment and iterative and self-organizing team. It contains two objects exchanged, three positions, and four forms of meetings called ceremonies.

2.1. 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. The tasks in product backlog could be (enhancements, features, defects, …etc.). In the product backlog, each task should have a

2.3 Scrum Ceremonies:

Scrum methodology has four types of ceremonies as the following: Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective. 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.

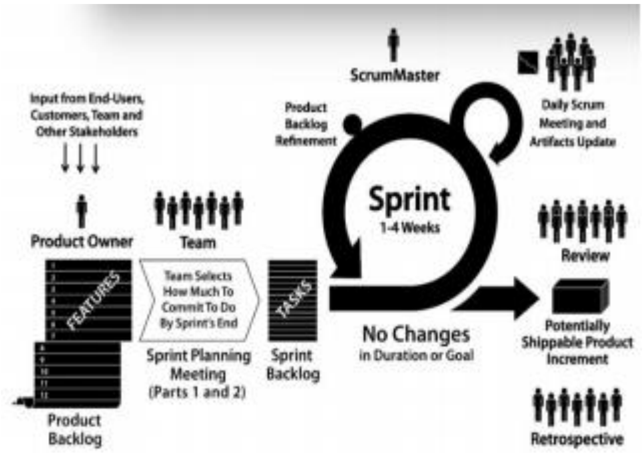
2) **Daily Scrum:** The team members have to meet each other daily to discuss the workflow of the sprint, what they did yesterday, what they doing today, and what they will do this kind of ceremony tomorrow called "Daily Scrum."

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.

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 .

2.4 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 Owner of a product, each team member chooses the right 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 the following figure, 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?"


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

After accomplishing [1,2,3] 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.

To emphasize the efficiency of scrum, the efficiency of scrum methodology. Chris and Frank 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 also the success of scrum in the field of software development, this study imported scrum in the development process for parts of vehicle spare, using visual studio as a scrum management process template. 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. 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.[1,4,5,7]

- # Case Study: How an engineering company reshaped its culture with Kanban [2,4:7,10]

Introduction:

Somabe is an industrial engineering firm designing and producing individual machinery. Every order of machines is different, and their creation is handled as a project.
Six years ago, the company was facing several problems:

• Tasks were verbally assigned, and there was no transparency regarding their states.

• There was a lack of awareness of the actual capacity of each department.

• Planning was based on estimates of ad-hoc effort, set in the budget, and must-meet.

• Work was flowing like a stream of big batches through departments.

• Problems were resolved thanks to heroic personal efforts, which, at some moments, created tension between people.

Before Kanban:

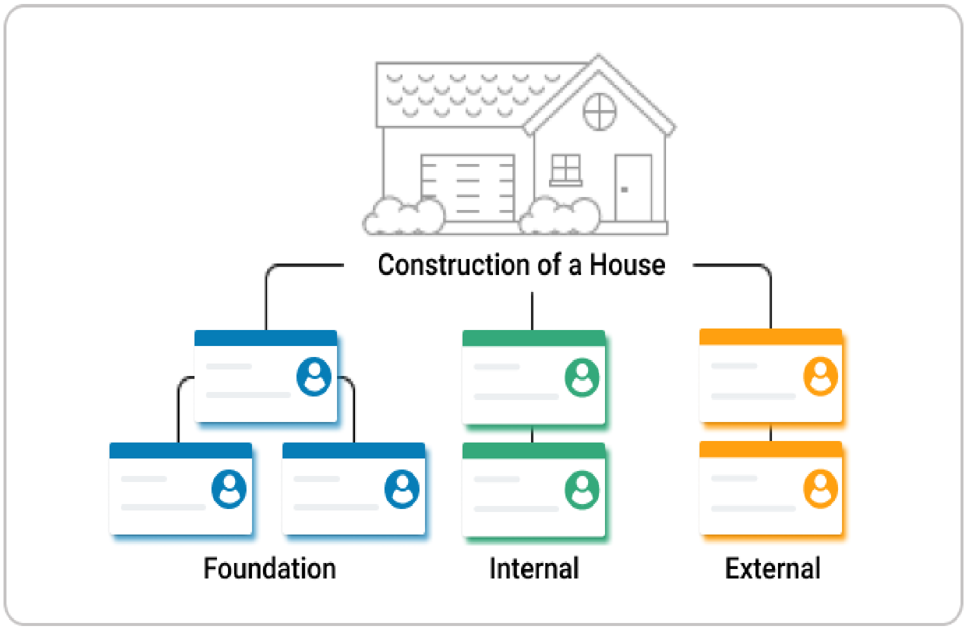
The situation was not sustainable. At that time, there was a change of management, and Aitor Eguren took the wheel of the company as General Manager. At first, Somabe agreed to follow Scrum to change this situation. For a certain period, this approach worked at the team level. However, a year or so later, Aitor started realizing that although they were organized in departments, they needed to manage the entire workflow to improve the project outcome. At the same time, it was necessary to break down the machine in assemblies to better manage their development and production and to reduce the large batches. Therefore, they choose to switch to Kanban and starter visualizing work with physical Kanban boards. Gaining a better overview, management at Somabes realized the need for more improvements in current work practices. They started analyzing digital Kanban tools to help them implement process changes, achieve client unity, and carry business agility and robustness to the next level.

Key Results:

We decided to use Kanban because the design and modification of our boards is so simple.

Developed a Work Breakdown Structure to Segment Smaller Complex Tasks, More Manageable Portions:

Kanbanize helped Somabe break down the work into smaller pieces by using different options for card links (parent, infant, relative to one or more other Kanban cards). Using these types of links, Somabe achieved a hierarchical structure of breakdown of work and introduced the Kanban concept of a portfolio.



Built an Advanced, Automated Workflow and Decision Taking for Aids Project Assignment, Encouraging Cross-Team Collaboration:

Starting with the management of the work through interconnected Kanban systems, Somabe created a more organized workflow and automated some of its routine acts with the assistance of the Company

Real-time gained, Company-wide visibility of all active projects and addictions:

Automating workflows, visualizing dependencies, and integrating Kanbanize into the company's projects with their ERP, CRM, BPM, and other engineering tools enabled Somabe to gain real-time visibility. This new level of transparency also made each department 's real ability available at a glance and allowed better management of possible workload and planning issues.

Created a Culture of Innovation, Learning, and Continuous Improvement:

“Kanban is the universal language of the entire company. Everyone understands and speaks it."
With its powerful analytics module, Kanban helped Somabe obtain an objective, data-based understanding of issues. This often demonstrated that problems were due to processes, not people, and encouraged a mentality of continuous improvement on all levels.

So, Why Kanaban? [1,3]

Kanban Helps Companies on a Portfolio level:

- Achieve a high level of transparency
- Organize and keep track of all initiatives and projects
- Acquire accurate, real-time status reports.
- Use data-driven forecasting

Why Kanaban rather Scrum? [1,2,3]

Scrum drives the completion of the story with what the team committed to completing in a fixed iteration of the length. Kanban concentrates on the flow of stories. Imagine stories being water, and being moved from the starting point to the ending point, they are completed.

In Scrum you take a bucket and push some water to the end stage. Maybe at the end you find you couldn't carry as much water as you thought without spilling it, So you're not filling the bucket quite as full next time. You can find that you can fill the bucket a little bit more without pouring out the water. Next time you 're adding only a little bit more to the tank, but all of the research is focused on moving water in the tank, assessing how it went and then doing it again — maybe with some changes to change.

You have got a gutter with Kanban to flow the water between two stages. You are watching all the different bends, turns and joints and looking for places where the flow is blocked and finding ways to smooth out those obstructions. You try things like increasing or decreasing water flow to determine the optimum rate for maximizing the system 's productivity. Any of this occurs without cycles loops in a constant manner.

My water comparison breaks down here a bit, but one of the main differences I see in Agile's new teams is that Scrum's iterations help promote the breaking down of stories to be at least small enough to complete within a single iteration. In Kanban, stories of improper size will appear in the metrics and they can always be split each time the problem of sizing becomes apparent, But for bigger stories it's a little harder to actually venture out into the production process. When the team has some practice breaking down stories into smaller tales, the issue goes off.

Another difference is that Scrum will lend itself to convincing a team member to spend extra hours fulfilling a commitment — losing track of the whole idea of "sustainable pace" Focusing Kanban on flow doesn't promote "I'm going to work on Saturday to get this done, just this once" moments quite as much.

4. Conclusion

In my research, an integrated methodology that integrating Scrum and Kanban methodologies to yield a comprehensive agile software methodology 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 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. It is emphasizing that 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, and at the end I have described the main differences in few steps. Also, I have discussed a case study to illustrate how we can use the Kanaban. At the end, I have given an Example or Case Study on Kanaban "Case Study: How an engineering company reshaped its culture with Kanban", and I explained the Key Results and why Kanaban? And why Kanaban rather than Scrum?

5. References

1. Sjøberg, D. I., Johnsen, A., & Solberg, J. (2012). Quantifying the effect of using kanban versus scrum: A case study. *IEEE software*, *29*(5), 47-53.
2. Johnsen, A. Quantifyingthe Effectof Using Kanban versus Scrum: A Case Study.
3. Alqudah, M., & Razali, R. (2017, November). A comparison of scrum and Kanban for identifying their selection factors. In *2017 6th International Conference on Electrical Engineering and Informatics (ICEEI)* (pp. 1-6). IEEE.
4. Brezočnik, L., & Majer, Č. (2016). Comparison of agile methods: Scrum, Kanban, and Scrumban. In *Proceedings of the 19th International Multiconference Information Society-IS* (p. 3).
5. https://openclassrooms.com/en/courses/4544621-learn-about-agile-project-management-and-scrum
6. https://www.mooc-list.com/tags/agile-development
7. https://www.mooc-list.com/course/running-valuable-design-sprints-coursera
8. https://www.planview.com/resources/guide/kanban-project-management-virtual-teams/kanban-board-examples/
9. https://kanbanize.com/kanban-resources/kanban-software/kanban-board-examples
10. https://www.planview.com/resources/articles/kanban-examples/