Journal of Theoretical and Applied Information Technology
15th November 2020. Vol.98. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SCRUM METHODOLOGY AND IBM DESIGN THINKING COMBINED: AN EFFICIENT WAY FOR DEVELOP A SYSTEM (CASE STUDY)

NILO LEGOWO[1], ARDIAN ADITAMA[2]
[1]Information System Management, Master of Information System Management
BINUS Graduate Program, Bina Nusantara University, 11530, Jakarta, Indonesia
[2]Information System Management, Master of Information System Management
BINUS Graduate Program, Bina Nusantara University, 11530, Jakarta, Indonesia
E-mail: [1]nlegowo@binus.edu , [2]ardian.aditama001@binus.ac.id

ABSTRACT

Choosing the correct methodology can lead system development to a better result. The traditional waterfall methodology is commonly known with its limitations with its inflexible and non-user-centric approach. Thus, to overcome waterfall limitation, other methodology can be performed in system development. Scrum methodology is said to be a new breakthrough to overcome classic waterfall methodology limitations. In order to overcome waterfall limitations, Scrum methodology can be combined with IBM Design Thinking to provide the best result as both concept build on user-centric concept. This research will be discussing how to combine and the advantages of combination of Scrum and IBM Design Thinking in order to overcome the Waterfall's limitations. This research will cover system development with scrum methodologies that will be combined with IBM Design Thinking concept and tools in related Dashboard and Report System development. This paper will not cover the old waterfall detailed process and will only discussed it's limitations. Scrum and IBM Design Thinking both are user-centric frameworks that can be used by system developers to help them to have a better understanding of their user. This research shows that scrum that complimented with IBM Design Thinking delivers better results than Waterfall methodologies. Both Scrum and IBM Design Thinking overcame the Waterfall's methodology limitation which leads to better result for all team involved.

Keywords: *Development, Waterfall, Scrum, IBM Design Thinking, User-Centric, System Development*

1. INTRODUCTION

In application development, SDLC is a conceptual framework or structural process of a phase that is in the development of an application from feasibility studies to launch and maintenance in the field [1]. In further discussion, SDLC has been developed and evolved that many software methodologies are available now. The adoption of software development methodology in developing this application is always considered useful and critical that can help the development team to manage and develop their own products and/or services [2]. Thus, choosing the correct methodology plays important role in system development success is self. One methodology that is often used is the traditional waterfall methodology. It is known that the waterfall methodology encounters many challenges. In Standish Group Chaos Report, a study between 2011-2015, states that out of 10,000 projects, projects that use waterfalls have considerable difficulty in completing the project as a whole. This results in a low success ratio of applications that use waterfall as a methodology. In that study, it is known that successful waterfall projects account for 11% of the total projects and 29% of projects with the waterfall methodology fail and 60% of the projects are constrained. According to Standish Group, one

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of the success ratios of applications is that the application can be completed with a reasonable estimated time, remains in the price range, has features and functions that meet the needs.

The waterfall methodology tries to be predictive: to create a schedule at the start of a project and to adjust the schedule. Complex software can be built sequentially, according to the phase where all requirements are collected at the beginning, all designs are completed next, and finally the main design is implemented into quality production software. This approach argues that complex systems can occur built in one pass, without revisiting and reviewing design requirements or ideas in changing business or technological conditions. But what generally happens is "the problem with this project is that users keep changing their minds".

Many companies still use the waterfall methodology in system development. But there are obstacles and deficiencies found during the development carried out. In this case study, PT. Jasa Teknologi IBM (IBM-JTI) developed this system of dashboard and report for their internal, with a traditional waterfall methodology. The time set for the team to work on the project was around 3 months. After some time, the development of the system was stopped because it was too costly for the company and also other limitation and constraints were found and made the system development crawled.

Nowadays, Scrum methodology is the most well-known and most widely used form of agile methodology, which is about 56% compared to other agile methods [3]. This shows that in agile itself, Scrum methodologies has been commonly known and performed among all other agile methodology options and can be proved as companies nowadays are using scrum in their system development. Waterfall methodology limitation found in previous work such are inflexible in getting additional requirement from users, fixed timed target despite all additional requirement, lack of user acceptance in the end of development process, tend to have a gap between requirement and result since waterfall is not really user-centric methodology and also too time consuming to accommodate the new or changed user requirement that result in unexpected additional time needed even though system development time has been settled beforehand. Scrum is a user centric methodology that will help system developer to understand their user needs and helping them to map the features and function that they need in the system to be developed. The highest priority of scrum is to satisfy the customer through early and continuous delivery of valuable software [4]. This is also the prove that Scrum is a user centric methodology, by doing early and continues delivery of system, Scrum are able to accommodate better user acceptance and expected user requirement versus the real life user requirement built. Scrum is welcoming change requirements for the customer's competitive advantage. Unlike waterfall, Scrum will Deliver working software frequently, from a couple of weeks (varies based on the team themselves, usually a preference to the shorter timescale). Scrum processes promote sustainable and steady development. User, businesspeople and system developers must work together throughout the project. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. Here are scrum methodology benefits [5]: increasing client satisfaction (decrease of complaints); improve communication and increase in team cooperation; increase in project ROI, increase in team motivation; improve product quality; decrease costs; increase in team productivity; decrease time to conclude projects; decrease project risk (lower failure possibility).

Patton's User Story Mapping is one approach that can be used in Scrum to map user stories and features needed by the users. Then comes the important question, "Who are our users and what do we build for them?" and implements a constructive but tiring process of contracting them into "User mapped backlogs". This is a day-to-day process with a lot of thought and hard team work, but it is a valuable investment of time and effort that will be recommended to every Scrum Product Owner [6].

Since IBM-JTI is a subsidiary of IBM Indonesia, it also has a IBM Design Thinking

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

framework that can be used for software development. IBM expands the IBM Design Thinking framework provides an approach to writing requirements, managing teams, and tracking project progress, including End User feedback during all phases of project development. The purpose of IBM Design Thinking is to expand the principles of Design Thinking applied to develop software that captures user needs with the speed and scale needed for application development. IBM Design Thinking has several modifications, including three practices that are unique to the framework: User sponsorship, Playbacks and Hills [7]. These three factors play critical part in conduction IBM Design Thinking process, they have big impact when being conducted when IBM Design Thinking is being performed. They helped the team aligned, flexible, full of idea and yet still customer-focused. The application of IBM Design Thinking can be used to improve the results of Scrum. Scrum and design thinking methods both contain principles, frameworks, defined sets of roles and its activities that set to be user centric [8]. When applied in real life, these two approaches can help developer team to align and deliver effective solutions that drive growth and bring value to users. Agile development is a quick process that relies on close people and user collaboration to respond to changes rapidly. The goal is to adapt the user need as quickly as we can and iterate with improvements.

Common misconception is that scrum methods only emphasize speed and flexibility at the expense of the customer experience. But is necessary to know that satisfied customers is a primary objective. However, if scrum developer team are often become over-focused on technical incremental improvement itself, they sometimes can lose sight of importance of customer focus and user centric concept of scrum itself. Thus, IBM Design thinking can be helpful in this condition. IBM Design Thinking uses techniques from social sciences like ethnography and psychology to help the team understands people better, design thinking can help uncover needs and most importantly aligning the team. IBM Design thinking also able to motivate and practicing rapid prototyping methods that enable teams to test new ideas quickly in which it is align with scrum concept to deliver faster. Combination of Scrum and IBM Design Thinking are expected to answer the limitation of waterfall methodologies to capture and adapt in getting new additional user requirement, flexible time of development based on user requirement request, accommodating and finding user acceptance in every iterative or sprint and also minimize the gap(s) between user requirement and result that will accommodate user requirement as needed to be and also minimize the unexpected additional time needed during system development itself due to additional changes needed.

2. PROJECT BACKGROUND AND PREVIOUS WORK

This study is based on the background of the problem that company requires Dashboard and Reporting. The current Dashboard and Report are Ad Hoc and can only be accessed if requested. The following are the Dashboards and Reports needed and will be developed along with the business units and PIC users of each of the Dashboards and Reports:

1) Asset Management Details

Unit: IT-Asset; User: Akil / Sari. This dashboard contains data assets from company. Company assets are including laptops, cameras, camcorders, portable DVD players, speakers, hard drives and laptop bags. The assets are held by each JTI employee. This dashboard plotted to help Asset-IT team can find out who is using a particular asset, the age of the asset, the price of the asset in support of the Asset IT team to maintain the assets of Company.

2) Asset Summary

Unit: IT-Asset; User: Akil / Sari. This report is a derivative report from Asset Management above. Some employees who did not extend the contract or resign did not rule out the possibility that they were still in possession of several items belonging to Company. This report is projected to assist the Asset IT team in validating the items returned by employees who have resigned to the data in the system regarding asset ownership.

3) Employee Overtime Dashboard

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Unit: All IBM-JTI Departments; User: Board of Directors, Team Management. Many employees from various divisions and projects at JTI often work overtime. Of the many projects and divisions, the management team and the BoD want to have a picture of what projects have the most amount of overtime, personnel who most often overtime and the amount of overtime in certain periods, for example per week, per month or even during the project carried out. It will make easier for the Management and BoD teams to validate the overtime process proposed by employees and can help the management team and BoD in estimating the level of health of the project for employees. But on this system.

4) Employee Attendance Dashboard

Unit: TSS; User: Romi / Shanty/. The TSS team is a team that has high mobility which they must meet in answering the needs of employees, but that does not mean their presence in the office can be underestimated. The presence of employees from the TSS unit in the office must be noted because it will affect their incentives. Leave and permits must be recorded daily so that the team leader from TSS can validate the incentives and discipline of their employees.

5) Preventive Maintenance Dashboard

Unit: TSS; User: Romi / Shanty. Some hardware units that have been installed by JTI require routine maintenance in certain periods according to what is stated in the work contract. In order to facilitate the TSS team to track which hardware has completed maintenance and which hardware has not done maintenance.

6) Whereabouts Dashboard

Unit: TSS; User: Romi. The TSS unit has the obligation to maintain and check the machines with customers according to their needs. Often a TSS employee must move more than once a day to meet customer needs. The TSS Management Team must monitor the movement of TSS employees to be verified to the relevant customer. Tracking the whereabouts of TSS employees is important so that the TSS Team can collect the right bills to the customer and maintain the obligations of the TSS team in fulfilling their responsibilities in accordance with the contract.

Those are the dashboard and report that will be developed in this study. Previously, Dashboard and Reporting were developed using the waterfall methodology. In its development, the following is the phase in the waterfall methodology used

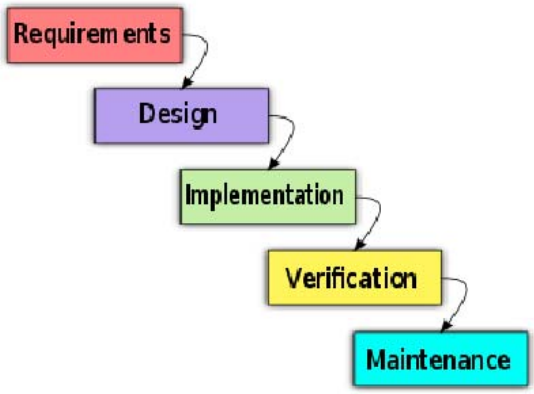

Figure 1 Waterfall Methodoloy Phases
Source: Saykol, Ediz [9]

Figure 1 shows waterfall phases commonly known. Definitions are as follows:

1. Planning and Requirements

In this phase, user requirements and requirements are collected. However, what happens in this phase, requirements can change according to user needs because the scope discussed is less clear and the user himself does not have an idea of the dashboard they need (users tend to request additional features after seeing the prototype). The absence of management scope and requirements becomes an issue in this phase. Even after development is done, often the team returns to this phase to discuss the requirements and scope. While the requirements change, the schedule that has been set cannot change, causing delays in processing reports and dashboards.

2. Design

At this stage the prototype dashboard and report began to be built. But often the user gets a different idea about the existing prototype so that this phase increases in Length.

3. Develop

The process of developing Dashboards and Reports is usually disrupted due to changes in

existing requirements. Modeling that has been made may be different or require adjustments to the needs.

4. Testing

When testing is done, users are often confused about the Dashboard and Reports that have been made, causing users to ask for adjustments to be made again according to their interpretation. This causes the project schedule to be used and causes delays to the schedule.

The project was stopped by the management because of many challenges and cost was wasted. Considering that it was not a big system to took such much cost. The following are constraints found during almost 3 months of this project using waterfall:

Table 1 Waterfall's Methodology Challenges Faced in Previous Development

Constraints	Description	Impact
Requirements change.	Requirements can change more than 1 time. This is also due to unclear scope so that the requirements are difficult not to move.The absence of requirements that remain permanent development time increases with changes that occur	Project delay (20% of total time)
The schedule is set even though the requirements are not fixed	The schedule is set for 3 months, because it was developed with a waterfall, some schedules have been set even with requirements that have not been fixed	The team found it difficult to reach the time target because the work was always there while the time did not increase.
Lack of Resources	Because the requirements are uncertain and always increasing, additional developers are needed to work on the project.	Resource costs increase
Resource re-assigned / deployed to another project	Many resources that are not always present in the internal project (according to company needs) cause the working team's resources to change frequently which have an effect on development time	In addition to increased costs, processing time also increases because there must be a handover from the old resource to the new one and the new resource must adapt for a long time before it can continue effectively
Resource cost	Because resources change, costs also change. As long as the project is not finished, the cost will always be there and increased.	Cost resources are a burden for companies because costs are better transferred to outside projects that are more profitable.

From the table 1 above, the waterfall method is incompatible with systems that have dynamic and changing requirements. The weaknesses of the waterfall methodology are the scope of the project must be clear, if there are changes it will be difficult to implement and if there is a problem in one phase then the problem can spread to other phases [10]. This leads us to the common result of this issue: keep changing user requirement, as waterfall is not really supporting this issue because all things need to be set beforehand, but in realty not every user are ready or aware enough of this state, this is when Scrum emerges as the solution.

The above constraints were the cause of the failure of the development of the Dashboard and Reporting after a few months running, the development was not resumed. Waterfall methodology is often related to more failure rate than scrum, the reasons are such as The requirements are not clear before beginning the project, users knows just what they want after seeing an early version of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the software. Requirements changes again and again during the development [11]. This issue will impact in time and overtime that can lead the team themselves burned out, easily distracted, unaligned and lost their focus. This is where IBM Design Thinking can help them to stay aligned and focused by their framework values such user sponsor, playback and hills and also IBM Design Thinking tools.

The other disadvantages of waterfall are: it needs all the requirements to be cleared specified at the very beginning which is hard for user in this study, waterfall does not allow changes to be made at least until the iteration is done, Longer wait until a working product is available. It doesn't accommodate risks and uncertainty. In waterfall, if there is next waterfall iteration, there may be no way to improve previous version. Time is wasted while waiting for each phase to be completed mean while changes can be request by the user at all time and all stages. [12]. Then, as the breakthrough of this problem, a set of methodology such as scrum, that is less focus on the overall control of the project and less bureaucratic, can help to resolve the issue. Scrum also preparing the team for changes and other problems that might come up during the project development. Thus, the company came up with new fresh idea to develop the system using different methodology, Scrum methodology.

3. PROPOSED FRAMEWORK

3.1. Scrum

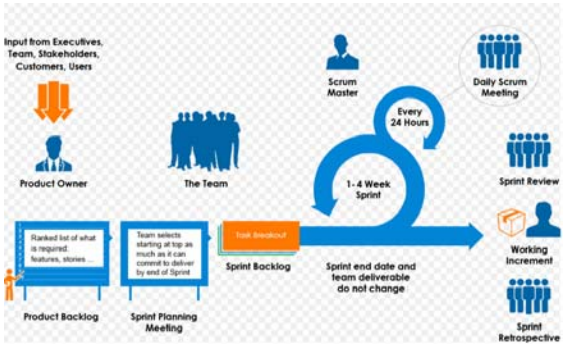

Figure 2 Scrum Process
(Source: Flaviu Fuiror [13])

Figure 2 above is the scrum process that commonly performed by all the development team across the world. Starting with Product Backlogs, Sprint Planning, Sprint Backlog, Daily Sprint and Sprint Reflection and the iteration will be restarted again and again. Table 2 below is the explanation of each scrum process:

Table 2 Scrum Process and its definition

Scrum Process	Definition
Product Backlog	Product backlog is a list of all the things that will be done in the project. Usually in the form of user stories and are user-centric. Product backlog criteria are [14]: Having added value for the customer, Prioritized as needed, The details of the product backlog depend on the priority position, There are estimates, ,Live- always developing, There are no items that can be ignored and not very technical.
Sprint Planning	Sprint planning is an event in Scrum where the Scrum Team determines the product backlog that will be worked on during the next sprint. Including discussing what they have to do in order to complete the backlog [15].
Sprint Backlog	The sprint backlog is the set of peroduct backlog selected to work on during the spint. The sprint backlog can be changed during the sprint and only the development team can change that.
Sprint	Sprint is an iteration in a certain period of time in the development of the system. In a sprint, the amount of work that must be done by the team has been arranged and is ready to be done. Usually 1 sprint ranges from 2-4 weeks.
Daily Scrum	In many projects, each sprint starts with a daily Scrum meeting. This meeting takes no more than 15 minutes and is conducted every day by the Scrum Master and Scrum Team [16]. At this meeting, what was discussed was o What has the team done or accomplished since the last Scrum? o What will be achieved until the next Scrum? o Are there obstacles that interfere with progress?
Scrum Review and Retrospectives	A retrospective sprint is an opportunity for the team to reflect on their sprint to be better in the next sprint. After the sprint retrospective, the process returns to sprint planning.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.2. IBM Design Thinking

IBM Design Thinking is a framework to solve users' problems (user-centric) based on design thinking and enhanced with IBM's research result as principal, value and process. IBM Design Thinking keeps our team lost focus on what matters for users while guiding them from ideas to outcomes faster and more efficiently [17]. IBM's Design Thinking is known for rapidly scaling practices to help team understand what user really need, IBM Design Thinking practices such as workshops that unite diverse perspectives and democratize decisions from the teams, user research paired with involvement of sponsor user that will ensures the teams really understand what users want and how they will use the solutions. IBM Design Thinking helps project teams stay in sync and align during outcome based on the IBM Design Thinking Keys: 'hills', 'playbacks' and exchange feedback.

3.2.1. IBM design thinking - the loop

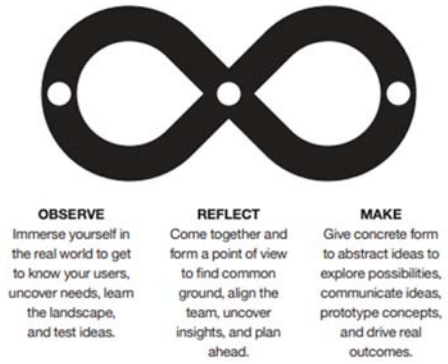

Figure 3 IBM Design Thinking The Loop Process (Source: IBM [18])

Unlike regular design thinking, that segment phases into a complex process, IBM Design Thinking built the model of a continuous and constant loop of activities as seen in Figure 3. Teams observe their users, reflect and synthesize what they've observed and quickly make a prototype (scratch) of a better user experience. Project team will always be moving through the loop of observe, reflect, and make. Cross-disciplinary teams engage in a series of collaborative activities and tools that bring better understand their user and envision the future experience. The low-fidelity deliverables and artefacts can be generated during these activities and enable the team to share ideas with users and stakeholders.

3.2.2. IBM design thinking – the keys

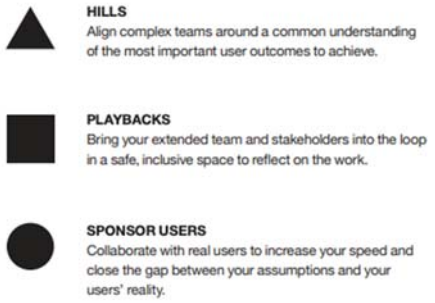

Figure 4 IBM Design Thinking-The Keys (Source: IBM [18])

Figure 4 is The Keys is a tactics or 'tricks' that are applied to help IBM Design Thinking practices able to navigate hard challenges and also team's diversity in demographic and perspectives. Hills, Playbacks, and Sponsor Users are the keys that help project teams align one to another and share a common user-centred purpose and maintain alignment across themselves within a complex project. Hills are statements of target or intention from the team that written as meaningful user outcomes. Playbacks bring all stakeholders into the loop to tell user-focused stories, information, prototype and exchange feedback. Sponsor Users are real-world users that contribute their expertise to the project, helping the project team stay in align with real users' real-world needs.

3.2.3. IBM design thinking - the principles

The principles of IBM Design Thinking are diverse empowered teams, a focus on user outcomes, restless reinvention. Diverse teams means the team is able to generate more ideas, increasing chance of a breakthrough in solving a problem. Empowering the team with the expertise and capabilities to turn those ideas into outcomes. A focus on user outcomes means that. Our users rely on our solutions and ideas to get solution of their problem done. It's Success measured by how well we fulfil our users' needs. Restless reinvention means everything is a prototype. When everything is considered as prototype that will go into another iteration, the teams are empowered to bring new ideas to even the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

oldest problems. The team will able to see problems and solutions from a new point of view.

3.3. Scrum And IBM Design Thinking

There are common understandings and ideas between Scrum and IBM Design Thinking which are: individuals and people interactions over processes and tools, prototypes over comprehensive artefacts, collaboration with users over paper contract negotiation, and emerging for change over sticking to the original plan. IBM Design Thinking delivers user experiences based on team's understanding, while scrum delivers software enablement within a short period. What these two things them most closely is the continuous cycle of experience maps and Playbacks.

Everyone grooms the product backlog. After a playback, team will collaborate on a that backlog. Throughout each release cycle, team meet to groom the backlog, updating the priority as needed and ensuring that backlog priorities is as user as shown in Figure 5 below.

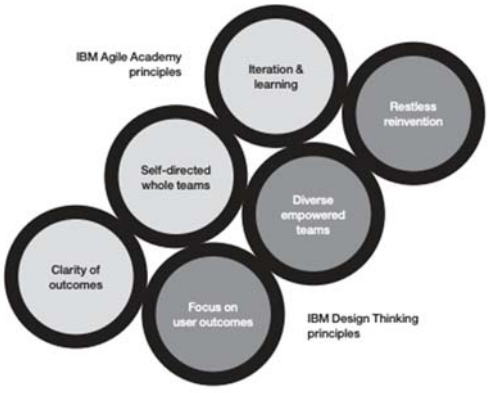

Figure 5 IBM Design Thinking The Loop Process (Source: IBM [18])

4. SCRUM AND IBM DESIGN THINKING FRAMEWORK

This study's research framework is based on the needs of IBM-JTI in developing the system. The following is the framework this study use:

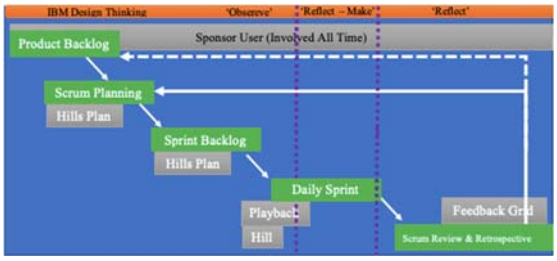

Figure 6 Framework Scrum and IBM Design Thinking

This research framework shown in Figure 6 is still based on scrum methodology's process. But, IBM Design Thinking's framework and it's deliverables were injected in scrum's processes. Below are Scrum methodologies process Following are the steps taken in developing this Dashboard and Report System along with its deliverables as shown in following table 3.

Table 3 Combined Scrum and IBM Design Thinking Framework and its Deliverables

Scrum Process	IBM Design Thinking - The Loop Phase	Deliverables (Scrum + IBM Design Thinking)
Product Backlog	The Loop – Observe	Need Statements matrix; User story; User sponsor Map / identification
Scrum Planning, Sprint Backlog	The Loop – Observe	Hills, user story with prioritation
Daily Sprint	The Loop – Observe, Reflect, Make	Hills Update (if any); Playback Plan and playback; Feedback, Backlog update (if any)
Sprint Retrospective	The Loop – Reflect	Feedback grid; User Story Updates (if any); Backlog updates (if any); Hills update (if any); Next Sprint Planning

As seen in Table 3, Product Backlog process, a list of all the things that will be done in the project are discussed and recorded. in the form of initial user stories and are user centric. Thus, IBM Design Thinking 'Observe' phase has been started also. In 'observe phase', the team will go deeper to user real life problem to get their needs and to feel the empathy. In this phase, the deliverables produced

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

such as: Initial user story, User sponsor identification/map and need statement matrix.

- Need statement matrix is tool in IBM Design Thinking to use with project teams when they feel that you're uncertain about the user's actual needs, desires or goals Need statement matrix helps the team to reorient or reframe the work around your user. To create Need statement matrix following are the easy steps: 1. Write statement like following: ***User needs** [a way to do something] that [addresses/helps their need]* **so that they** *[benefit directly].* 2. Focus on user's pain points to help get what the problems are. 3. Do not listing features, use following instead: "What does my user really seek? What does she really want?" 4. Group similar ideas and discuss.
- Stakeholder map helps the team to integrate team members, identify project stakeholders, their expectations, and also relationships. To build stakeholder map, 1. Diverge on identifying stakeholders "Stakeholders" can be project leads, team and the roles, executives, partners, customers, and also end users, write them in a stick note each. 2. For each stakeholder, add a quote expressing their thoughts, opinions, or expectations. 3. In parallel, cluster stakeholders and label the groups. 4. Draw and lines among groups representing their relationships such as influence, process, or dependencies or any relationship

In Scrum/Sprint planning, Scrum Team determines which product backlog that they will work the next sprint. This includes discussing what they have to do to complete the backlog. Then the team will have composition of Sprint Backlog, the set of product backlog selected to work on during the sprint. Sprint backlog can be updated during the sprint as needed. Thus on these two critical process, IBM Design Thinking can help to observe which user story is needed to be cleared first and when to clear it. The team used 'hills' to help them with these user stories and works they about to do.

- Hills
  Hills helps turn users' needs into project goals and with a target completion time , help aligning the team align around a common understanding what is the intended outcomes to achieve and when. Hills are composed of **"Who"** (user or group of users), a **"What"** (action or enablement), and a **"Wow"** (a measurable, market differentiator).

In a sprint, the team will do the work and also perform daily scrum meeting to discuss all their work. Each sprint starts with a daily Scrum meeting. This meeting takes no more than 15 minutes and is conducted every day. In this phase, IBM Design Thinking phase 'Observe-Reflect-Make' will also being performed. Observation to user story point is still needed up to this point, followed by coming together in daily sprint meeting and form a point of view to find common ground and aligning the team, uncover insights and ideas, and plan ahead. Then the team will start to make prototype (or their work). In this process hills update and user story update can be occurred. In this process, Playback is being used to aligning the team and to get certain feedback from teams.

- Playback
  The teams reflect together to give and receive criticism. In playback all teams can be invited and shown the progress, ideas, thoughts and also prototype. In playback, the team encouraged to received any feedback to make the product better. So playback will help the team uncover new ideas, feedback and user sponsors perspectives.

The last part in scrum iteration is sprint review and retrospectives. In this process sprint has been closed and sprint will be reviewed by the team. To help reflection, IBM Design Thinking provides a tool named feedback grid:

- Feedback grid helps to gather any sort of feedback across the teams along with questions and ideas in order to help the team to improve and determine the next steps. To create feedback grid: 1. Draw four quadrants grids with following label: Things that 'worked',

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Things to 'change', 'New ideas' and 'Questions' team still have. 2. Each member of team can fill in each quadrant. Be specific when writing the statements and give constructive criticism. 3. Group similar ideas and discuss for patterns.

After the Scrum Review and Retrospective are done, the Scrum process will return to sprint planning. If there are requirements that change priorities, the team might return to the product backlog list to adjust the backlog. Then the iterative scrum process will spin again.

5. REVIEW AND DISCUSSION

After all things are set, all sprints were done according to framework and study above. Below are the review and discussions:

a. Sprint 0

This is also known as initiation or preparation sprint. In this sprint, there were project briefing, environment setup, initial interview and product backlogs and statement clarification and also initial user stories. In this phase 34 user stories were found. Scrum planning were held to compose sprint 1 works (with total 40 User story points) and also it's hills plan.

b. Sprint 1

In this sprint, 17 user stories with total 40 points has been set beforehand. Asset Management Detail Report and Summary Asset Dashboard were prioritized by the users. 12 additional user stories found. Following is the burnt down chart of sprint 1:

Table 4 sprint 1 burnt down chart

Day	Burned down Plan	Burned down Actual	Balance Plan	Balance Actual	Daily Completed
0			40	40	#N/A
1	3	2	37	38	2
2	3	2	34	36	2
3	4	4	30	32	4
4	4	5	26	27	5
5	3	4	23	23	4
6	5	5	18	18	5
7	6	4	12	14	4
8	4	6	8	8	6
9	6	6	2	2	6
10	2	2	0	0	

As shown in table 4, Sprint 1 has been completed by the development team in around day 9. The above features and user stories have been done and accepted by users. After gaining acceptance from users, the development team deployed features that had been built into the IBM-JTI production server.

c. Sprint 2

As mentioned above, there are 12 additional user stories for the Asset Management Detail report and the Asset Dashboard Summary. Therefore 12 user stories will be added to sprint 2. In total there were 13 user stories. In this Sprint 2, the authors conducted an experiment by not doing design thinking on the Employee Overtime dashboard. The following are burnt down chart during Sprint 2

Table 5 sprint 2 burnt down chart

Day	Burned down Plan	Burned down Actual	Balance Plan	Balance Actual	Daily Completed
0			40	40	#N/A
1	2	2	38	38	2
2	4	3	34	35	3
3	4	3	30	32	3
4	3	3	27	29	3
5	8	3	19	26	3
6	5	4	14	22	4
7	6	3	8	19	3
8	4	7	4	12	7
9	2	6	2	6	6
10	2	6	0	0	

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In the table 5 above, it can be seen that in the middle of the sprint (days 6, 7 and 8) there is still a lot of work that has not been completed. This can be seen from the actual value which is still at number 22 on the 6th day and only advances 3 to 19 the next day. It is possible that the absence of hills affected the work of the development team, which in turn could result in overload on the development team itself. In feedback sprint 2, the development team felt they lost guidance during sprint 2. This is because there is no IBM Design Thinking and hills planning that is usually done before the sprint begins. This also causes the output of the team's work not to the user's liking.

d. Sprint 3

Sprint 3 remains focused on the Overtime dashboard and report. 15 user stories were chosen. User Story in Sprint 3 is still dominated by many previous sprint results. Because there are some adjustments that must be done, some of the user story points above have been adjusted to the needs of the development team in this sprint.

Table 6 sprint 3 burnt down chart

Day	Burned down Plan	Burned down Actual	Balance Plan	Balance Actual	Daily Completed
0			40	40	#N/A
1	3	3	37	37	3
2	5	3	32	34	3
3	4	5	28	29	5
4	5	3	23	26	3
5	4	5	19	21	5
6	6	7	13	14	7
7	3	3	10	11	3
8	4	3	6	8	3
9	4	5	2	3	5
10	2	3	0	0	

As a result of Sprint 3 in table 6, dashboard overtime can be received by BoD and has been deployed to the production server. That way the development team can proceed to the dashboard and other reports in the next sprint.

In the Sprint 3 feedback grid, the team agreed that there were significant changes felt by the whole team when the IBM Design Thinking was done. The process and results of work have positive results both on the development team and the user directly. However, there are some issues such as the format and structure of the TSS team which are often different from other teams in working on all reports which may be a challenge for the development team themselves.

e. Sprint 4

In Sprint 4, the development team built 3 different types of dashboards and reports. Continuing the process of developing a dashboard and report system, the development team will create hills based on the user story planning table above.10 user stories were chosen. After the sprint runs for a few days, there are several obstacles to Preventive Maintenance. The raw format and raw data of Preventive Management produced still needs to be further processed before it can be used which results in additional work. This resulted in changes in work effort and changes in hills in Sprint 4. The user whereabouts report is dropped because the user story and the workload needed to do preventive management change. Whereabout report can be done in the next sprint (Sprint 5) and hills sprint 4 is updated. Table 7 below is burnt down chart of sprint 5.

Table 7 sprint 3 burnt down chart

Day	Burned down Plan	Burned down Actual	Balance Plan	Balance Actual	Daily Completed
0			40	40	#N/A
1	4	3	36	37	3
2	4	4	32	33	4
3	3	3	29	30	3
4	5	3	24	27	3
5	4	4	20	23	4
6	4	6	16	17	6
7	5	3	11	14	3
8	7	7	4	7	7
9	2	3	2	4	3
10	2	4	0	0	4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A fairly rapid identification from the development team regarding the possibility of work with a greater weight than the previous plan has helped the team develop a better and more realistic plan. By moving the whereabout report, the team can focus on the 2 available reports. On the burndown chart above, the team's work load is also balanced from the beginning to the end of the sprint.

f. Sprint 5

In Sprint 5, the development team can focus on whereabout report that was displaced in the previous sprint. In addition, the team will still be working on adding the user story the user requested from the previous sprint.

Table 8 sprint 5 burnt down chart

	Burned down		Balance		Daily
Day	Plan	Actual	Plan	Actual	Completed
0			16	16	#N/A
1	2	2	14	14	2
2	2	1	12	13	1
3	2	3	10	10	3
4	3	2	7	8	2
5	3	4	4	4	4
6	2	2	2	2	2
7	2	2	0	0	2

As table 8 show, because the load from Sprint 5 is not too heavy, from being burned down above, the sprint has been declared complete on Day 7. The results are 3 days faster than normal sprints.
With the closing of Sprint 5, the development of the dashboard and report system for internal IBM-JTI was also revealed.

As stated above, combination of Scrum and IBM Design Thinking are expected to answer the limitation of waterfall methodologies. Below are the review of Scrum and IBM Design Thinking performance in answering following concerns of waterfall methodologies: accommodating new additional user requirement, flexible length time of development based on user requirement request (not fixed as waterfall methodology), facilitating better user acceptance in every iterative or sprint and also minimize the gap(s) between user requirement and result that will accommodate user requirement as needed to be and also minimize the unexpected additional time needed in development due to user requirement changes.

This project takes 5 sprints or a total of 47 days (two and a half months) and is faster than the initial estimate of about 3 months. During this project, the development team has worked on a total of 55 User Stories, with details: Number of initial User Stories = 34 User Stories; Added 21 User Stories with 21 User Stories with details of severity.

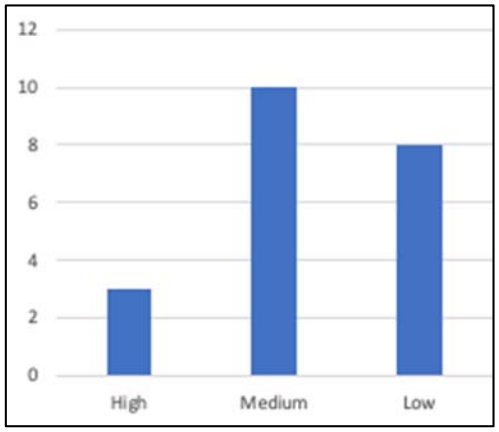

Figure 7 Additional User Strory Per Severity

From the figure 7 above, here is the addition of user stories per severity: 3 high severity user stories: 10 medium severity user stories: 8 low severity user stories

The addition of user stories classified as High and Medium is 13 user stories. If this user story is not implemented, it will affect the user's acceptance of this system. This also indicates that there are still important additions during the project that must be accommodated by the development team. Chances are the user is unaware or unaware of what they need at the start of the dashboard and report system. There are 38% additional user stories added during the running sprint:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

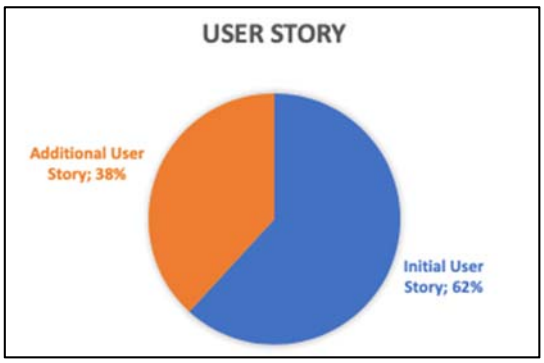

Figure 8 Percentage of Initial vs Additional User Story

Figure 8 shows an increase of 38%, the picture above supports that Scum is indeed more adaptive in absorbing changes and additional requirements that occur during the development of a system. As many as 38% of additional user stories have been absorbed and worked on during the development of this system. Adding or revising user stories such as in sprints 4 and 5 will be adjusted to the existing sprint backlog so that the development team does not feel burdened with more and more targets and the target completion time has not changed. Adjusting the backlog on this scrum helps the development team measure their workability with the available time. Here are the addition of user stories per sprint:

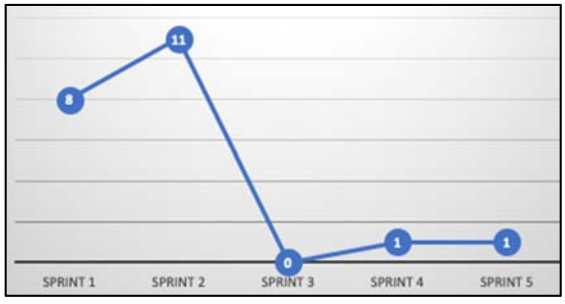

Figure 9 Addional User Story per Sprint

From data in figure 9, the most addition to user stories lies in Sprint 2. This means that the use of IBM Design thinking affects the addition of user stories at the stage of development of this project. Total 11 (52% of the total additions), the addition of user stories is obtained in sprint 2. Following are the status per sprint: Sprint 1,3,4 and 5 are declared as 'GREEN' only Sprint 2 declared as 'RED'. Sprints are declared GREEN if there are no unrequited user stories brought to the next sprint. The sprint is declared FAIL (also called "red") if there is an unfinished user story brought from the last sprint.

80% of Sprints are declared PASS and only 2 sprints fail. In sprint 2, the majority of user stories are brought to sprint 3 due to misunderstandings that occur during the development, as a result the user story must be carried over to sprint 3 and will be worked on including the original sprint 3 user story itself. The following is a burndown chart for each sprint:

1) Sprint 1

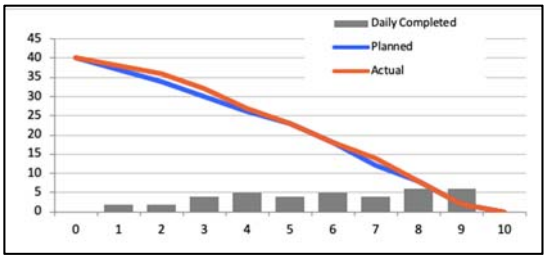

Figure 10 Sprint 1 Burndown Chart

In the first sprint, the plan and actual beginning of the sprint are still some distance away from meeting the point as shown in Figure 10 above. But on days 4 and 5 the plan and actual start to meet and go hand in hand until the end of the sprint. Burndown charts like this are ideal burned down charts for sprint system activities.

2) Sprint 2

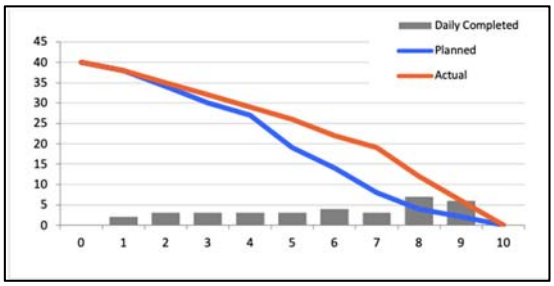

Figure 11 Sprint 2 Burndown Chart

In this sprint, IBM Design Thinking was not carried out. From the beginning of the sprint, the actual and planned charts have not yet met a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

common ground as shown in figure 11. Plan and actual charts only meet at the end of the sprint (days 9 and 10). This indicates that the development team was confused or lost direction or guidance in doing the work in Sprint 2. The gap that occurs is a bad example of a burndown chart on a sprint. If there is no meeting point at the end of the sprint then the sprint will automatically be declared red or not finished.

3) Sprint 3

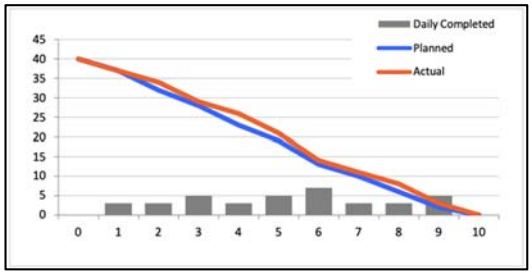

Figure 12 Sprint 3 Burndown Chart

With the re-implementation of IBM Design Thinking, the development team returned to work in a directed direction and the results can be seen from the graph above that the gap between actual and plan is not too far away during the sprint until it finally meets at the end of the sprint. There is no gap that is too far away as in the previous Sprint 2 as shown in figure 12.

4) Sprint 4

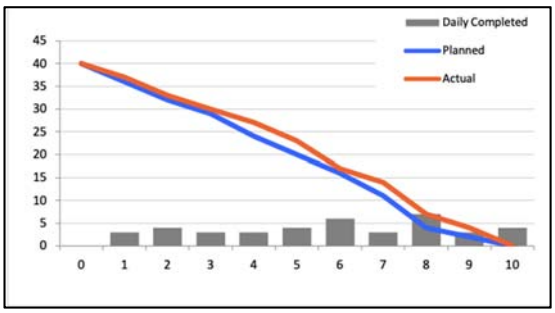

Figure 13 Sprint 4 Burndown Chart

In this sprint, identifying possible issues found by the development team plays an important determinant. With the discovery of this, figure 13 shows that the user story and the work of the development team can be adjusted to the existing conditions. The result is the achievement of the user story in the sprint without the issue being interrupted.

5) Sprint 5

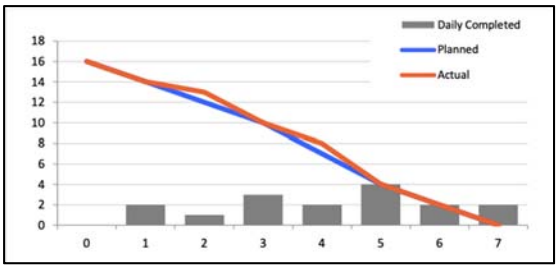

Figure 14 Sprint 5 Burndown Chart

This sprint has a lower load than other sprints and is completed faster as shown in figure 14. Although there are additions requested in the middle of the sprint, additions can still be done and completed properly. Sprint 5 closes the development of this system.

From the comparison of the sprint burndown chart above, in sprint 2 because the IBM Design Thinking was not carried out, it looks like a big gap compared to other burndown charts. This gap is caused by a mismatch of the execution carried out and the plan that has been made. In sprint 2, plans made do not refer to "Hills" as in other sprints. Therefore, the plan made is not in line with the weight of the user sprint story itself and also results in actual execution that is problematic.

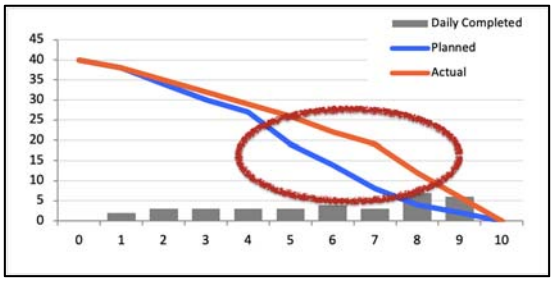

Figure 15 Gap Plan vs Actual Activities in Sprint 2

Although at the end of the sprint 2 burndown chart it appears that the actual plan and execution finally reached the target, but in reality, the results of sprint 2 are far from the user's expectations as shown in figure 15. From these results it can be concluded

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

that IBM Design Thinking can help users and the development team be more aligned in developing this system. If the user and the development team are not in line, then events like sprint 2 can be repeated, the sprint is declared finished by the development team but does not meet the wants and needs of the user.

6. CONCLUSION AND RECOMMENDATION

Combination of Scrum and IBM Design Thinking are expected to answer the limitation of waterfall methodologies in accommodating new additional user requirement, flexible length time of development based on user requirement request (not fixed as waterfall methodology), facilitating better user acceptance in every iterative or sprint and also minimize the gap(s) between user requirement and result that will accommodate user requirement as needed to be and also minimize the unexpected additional time needed in development due to user requirement changes. This research found that scrum and IBM Design Thinking are complementing each other to provide the best result for user or customer. IBM Design Thinking framework and it's tools helps team to deliver the best outcome to user. Seems to author that IBM Design Thinking helps and compliment Scrum practices in each of it's phases.

From the study above, here are the conclusions that the writer can draw 1.Developing system using the Scrum methodology and IBM Design Thinking has extraordinary results in absorbing additional user requirement or changes in the development of the system itself(which is often the case). As seen on above result, many additional or user requirement changes emerged on performing sprint, this surely impact time and effort set in sprint planning. But, as stated on literature above, this is one of the strength of scrum itself, to adapt to changes needed in order to accommodate user needs, then after a few adjustment is made then all things were back on track. Since the length time of project itself has not been set beforehand like waterfall methodology, Scrum methodology and IBM Design Thinking framework are able to adapt to its changes and capture any logical and reasonable user needed and also set a better understanding about length time needed to develop the user requirement to user themselves. Communicating the time needed and other impact due to additional requirement or changes is very important value in this combination of Scrum and IBM Design Thinking framework so the users are able to understand the effort needed due to their request and the delivery due date. 2. Implementation of the use of the Scrum methodology and IBM Design Thinking helps the system development team and its users stay in line and remain open to each other. IBM Design Thinking basic concept of empathy and sympathy played the basic way to understand the users better. With empathy and sympathy, the development team can feel, understand and analyse user needs and also assisted by IBM Design Thinking tools they were able to reflect what user needs and align themselves one to another. This is the result of user centric concept in capturing user needs and (later on) user acceptance in every iterative of sprint. It can be seen that from all phases above that less testing phase were conducted as the result of better user understanding from requirement development so that the test phase itself is not repeatedly conducted as the result of the system build has been answering the need of user requirement. 3. IBM Design Thinking is very effective in helping Scrum in understanding and meeting user needs. With the IBM Design Thinking, the development team can produce things that are more effective in answering the needs of the user. On the other hand, when IBM Design Thinking is not done, the development team loses direction and empathy for users who suffer from the waste of effort and time that has been done during system development. This also minimize the gap between user requirement and result that accommodated user requirement. 4. The last thing that author find unique of this research is sometimes scrum only is not enough to set the team focus on both development and user centric practice. As can be seen on sprint 2 when IBM Design Thinking did not performed, the team lose their focus on user centric practice and tend to finish the technical part as they wanted not per user needed. This indicates that IBM Design Thinking and it's tools play quite a good role to compliment Scrum to fulfil the user needs, team alignment and team focus. To help the team set their eye on user, IBM Design Thinking plays an important role through it's concept and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

tools. 5.The author recommends using Scrum and IBM Design thinking in project development in addition to developing dashboards and reports or similar systems. The use of Scrum and IBM Design Thinking can be extended to more complex and more complex applications to see other results from the combined methodology.

However, this study is only limited in developing the system of dashboard and report system. It is relatively smaller system if it compared with bigger system with more features and business process on the system itself. Perhaps, other results can be emerged if the system being developed contains more business processes and automation. With more varies user stories and far more cases, maybe the use of scrum can be utilized more fully. Other limitation is IBM Design Thinking tools used above are being used as needed from the team, there are still many IBM Design thinking tools that can be used as needed and no specific justification that can be used as firm reference to when or why using certain tools. The use of these tools might produce different experiences for users and the development team. Scrum development combined with IBM design thinking is recommended to be further developed as so many areas that can be elaborated with, so that the results can be more diverse and effective in breaking down the combination of Scrum and design thinking itself.

This study shows that Scrum and IBM Design Thinking are able to overcome waterfall limitation with quite outstanding result and made the system development more user-centric and user friendly and also good result in aligning all the teams involved. This study shows that user-centric frameworks can work together in harmony and complementing each other.

BIBLIOGRAPHY

[1] N. B. Ruparelia, "Software Development Lifecycle Models," 2010.

[2] C. W. L. K. Y. T. Whee Yen Wong, " *Importance of a Software Developn Methodology in IT Project Management: Innovative Six Sigma Approach - A Case St of a Malaysian SME Organization,* 2013.

[3] C. Kulbacki, "Scrumstar: What's more popular than scrum," 23 Jun 2018. [Online]. Available: https://scrumstar.com/articles/the-most-popular-agile-methodologies. [Accessed 2020 Apr 4].

[4] Kent Beck Mike Beedle Arie van Bennekum, et al, "https://agilemanifesto.org/," 2011. [Online]. Available: https://agilemanifesto.org/principles.html. [Accessed 14 July 2020].

[5] Bernardo Vasconcelos de Carvalhoa , Carlos Henrique Pereira Mellob, "Scrum agile product development method - literature review, analysis and classification," vol. 9, p. 11, 2011.

[6] T. Yoshida, "Medium Try Design Thinking + Scrum," 19 Sept 2018 . [Online]. Available: https://medium.com/@takeshi.yoshida/design-thinking-plus-scrum-d671a1a8e67a. [Accessed 2020 Apr 4].

[7] Lucena, Percival; Braz, Alan; Chicoria, Adilson; Tizzei, Leonardo , "IBM Design Thinking Software Development Framework," pp. 98-109, 2017.

[8] I. I. f. B. Team, "https://www.ibm.com/downloads/cas/7KL6JLMJ," IBM Corporation , New Orchard Road, Armonk, NY 10504, 2018.

[9] E. Saykol, "An Economic Analysis of Software Development Process based on Cost Models," 2012.

[10] M. Mahalakshmi, M. Sundararajan, " Traditional SDLC Vs Scrum Methodology – A Comparative Study," 2013.

[11] Bruno Antunes, Diogo Santos, Eurico Lopes, Filipe Fidalgo, Paulo Alves,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

"Blisstrail: an agile project business case study," *Procedia Computer Science,* no. 64, p. 529 – 536 , 2015.

[12] Vaishnavi Kannan, Smita Jhajharia, DR.Seema Verma , "Agile vs waterfall: A Comparative Analysis," *International Journal of Science, Engineering and Technology Research (IJSETR),* vol. 3, no. 10, pp. 2680-2686, 2014.

[13] F. Fuiror, "Key elements for the success of the most popular Agile methods," p. 29, 2019.

[14] Y. Obergfell, "www.scrum-institute.org," [Online]. Available: www.scrum-institute.org https://www.scrum-institute.org/The_Scrum_Product_Backlog.php. [Accessed 17 Jul 2020].

[15] agilealliance, "agilealliance," [Online]. Available: https://www.agilealliance.org/glossary/sprint-planning/#q=~(infinite~false~filters~(postType~(~'page~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'sprint*20planning))~searchTerm~'~s. [Accessed 17 July 2020].

[16] H. F. Cervone, "Understanding agile project management methods using Scrum," *Understanding agile project management methods using Scrum,* vol. 27, no. 1, pp. 18-22, 2011.

[17] A. F. T. E. I. Study, "The Total Economic Impact™ Of IBM's Design Thinking Practice How IBM Drives Client Value And Measurable Outcomes With Its Design Thinking Framework," Forrester, 2018.

[18] IBM, "IBM Design Thinking Field Guide," IBM Corporation, 2017.