CHITKARA

Journal on Today's Ideas - Tomorrow's Technologies
Journal homepage: https://jotitt.chitkara.edu.in/

Scrum: An Agile Software Development Process and Metrics

Rajani Dixit[1], Brij Bhushan[2]

[1]*School of Computers, IPS Academy, Indore, Madhya Pradesh, India.*
[2]*Mewar University, Mewar, Rajasthan, India.*

Email: [1]rajanisharma.ips@gmail.com, [2]brij@nic.in

ARTICLE INFORMATION

DOI: 10.15415/jotitt.2019.71005

ABSTRACT

In a traditional software development process such as the Waterfall Model, works best in a stable environment. But, it is not flexible when it comes to change. There is a gap in the interaction between the users and the development team which leads to incomplete and misunderstood specification. Because of this, the end product is sometimes a surprise to users and this gap accelerates incorrect development of the software product. Once requirements are frozen there is no scope of accepting changes. There is a need for a framework which holds the solution for all these situations. With this premise, the agile development methodology came into existence. Scrum, an agile approach supports continuous collaboration among the customer, team members, and other stakeholders. Its time-boxed approach and continuous feedback from the product owner ensures the development of working product with essential features at all the time. This paper explains the agile software development approach, its proclamation and different frameworks of agile approach. Further illustrate most widely used framework: Scrum. This research paper covers the implementation and application of Scrum. It focuses on why Scrum is preferred over the Waterfall Model with the help of some survey results and later a discussion on some Scrum Metrics which will be helpful and accounting for the best Scrum Practices in achieving goals set by the software development team, the product owner and the customers. The outcome of this study shows that Scrum Metrics is critical and highly valuable for successful product development. The quantitative insight that these metrics provide for the Scrum Team, Product Owner and Stakeholders is necessary for achieving strong project dynamics and optimal results.

Keywords: Agile, Process Model, Scrum, Scrum Metrics

1. Introduction

The Software Development Industry works with many methodologies to provide their best results. Earlier development methodology would be like the Waterfall Model, Sequential Model, etc. were most extensively used for Software Development. In the Waterfall Model, all the steps or phases taken place sequentially there is no return back to the previous step or phase. But the major drawback with this model is that when it is in the

 https://doi.org//10.15415/jotitt.2019.71005 | ISSN No.: 2321-3906 (Print) ISSN No.: 2321-7146 (Online) Registration No. : CHAENG/2013/51235
Periodicity: Bi-Annually

deliverable stage to the Client or Stakeholder then it is too old in the technology and client would see it the first time[1-3]. It may be it is too different from their expectations. This is Watermelon Situation where it seems to be green or all right from outside but actually, it is red or Problematic from inside [4].

Waterfall model is the first process model used for software development. It is divided into separate phases. There is no overlapping of phases. One phase acts as an input for the next phase. Any Phase will begin when the previous phase is completed. It is a linear sequential model like a waterfall. Its stages are Requirements, System Design, Implementation, Integration, and Testing, Deployment and Maintenance [5]. In this model, the output will be available towards its end phase. If the client was not satisfied it is a total wastage of time, Money and Effort. No changes may be possible at this stage. It is not suitable for the project in which requirements keeps on changing according to market Scenario [3].

It is a well-known fact that technology keeps on changing. With the changing requirements of technology, using the traditional model for software development process it is not apt. Past few years have witnessed the huge growth in app development. Within a period of Six months ,companies are launching new mobile sets, with some different techniques and features [6]. Thus if the Waterfall Model is followed, it will end up with an outdated product in the Market. An Approach or framework, which can be a solution to all these problems during the Software development process that is Agile Software Development [7, 8].

2. Agile approach of software development

In general term, Agile can be defined as the ability to move quickly and easily. This approach was discussed in February 2001 in Utah when a team of Software Developers met to find a solution for a situation that occurs every day in the field of Marketing, Management, External and Internal Customer and developers who don't want to make hard trade-off decisions, so they impose irrational demands through the imposition of corporate power structures [9].

They published the Manifesto for Agile Software Development for bringing up better ways of developing software by doing it and helping others to do it. They value Individual and Interaction over Processes and Tools, Working Software over Comprehensive documentation, Customer Collaboration over Contract Negotiation, Responding to Change over following a plan [9].

Manifesto [9] consists of 12 principles which are as follows:

1. Utmost Precedence is given to customer satisfaction, which is exhibited through primary and constant software delivery.
2. Requirement changes are accepted at the late stage of software development to address the customer's economic benefit.
3. The regular release of working software, which could range from a few weeks' time to a months' time, with the inclination towards short schedule.
4. Both the teams that are a corporate team and software development team works together during the entire project.
5. During the project, developed an enthusiastic ecosystem, which would help the individual in setting goals to achieve.
6. Communication within the development team is face to face conversation.
7. Progress of a project can be measured with the help of working software.
8. The Patrons, developers, and users are ought to continue constant speed open-endedly.

ISSN No.: 2321-3906 (Print) ISSN No.: 2321-7146 (Online) Registration No.: CHAENG/2013/51235
Periodicity: Bi-Annually

9. Agility increases with the constant concentration to quality designs and technical excellence.
10. It is important to do the simplest thing that does the job required.
11. Self-organizing teams choose the best architectures, requirements and designs rather than being directed by others outside the team.
12. A team must come together at regular interval to discuss the current status of work if required than modifies and adapt its performance appropriately.

Agile Software Development approach can be implemented in many frameworks. There are 7 different types of Agile framework namely Extreme Programming (XP), Feature Driven Development (FDD), Dynamics System Development Method (DSDM), Kanban, Crystal, Lean and Scrum [10, 11].

1. Extreme Programming (XP) is founded by Kent Beck. It is used in a scenario where quality, efficiency, customer focus, and feedback addressed at a priority level.
2. Feature- Driven Development (FDD) is founded by Jeff De Luca. It is a model -driven short iteration process that is built for user-focused features.
3. Dynamics System Development Method (DSDM) is founded by DSDM Consortium. It focuses on a structured approach to rapid development and direct teams to deliver on time and within budget.
4. Kanban is founded by David J Anderson. It focuses on three principles that visualize the workflow, limit the amount of work in progress and just-in-time development.
5. Crystal is founded by Alistair Cockburn. It realizes that every project involved somewhat customized set of policies, practices, and processes to fulfill the product's exclusive features.
6. Lean is founded by Mary and Tom Popendieck. It is highly flexible methodology without firm strategies and policies. It eliminates waste work that does not add customer value.
7. Scrum is founded by Jeff Sutherland and Ken Schwaber. It is an iterative and incremental model that follows a set of roles, responsibilities, and meetings that never change.

Sutherland and Schwaber [12] conceived the process Scrum in the early 1990s. The Term came from Rugby and referred to a team working toward a common goal. They codified Scrum in1995 and presented in the conference (OOPSLA'95) Austin, Texas [13] and published as a paper titled "SCRUM Software Development Process." Sutherland along with John Scumniotales and Jeff McKenna adapted this model for Software for the first time.

Scrum was thought to be the most popular framework for implementing Agile Approach [11]. It is a lightweight process which is iterative in nature used to manage complex software development. Fixed length iteration called sprints which are enduring for one to two weeks. It allows the tea to ship software or product on a regular pace.

Scrum follows a set of roles, responsibilities, and meetings that never change. Scrum organizes four ceremonies that provide structure to each sprint: Sprint Planning, Daily Stand-Up, Sprint Demo and Sprint Retrospective. During each sprint, the team will use visual artifacts like task boards or burndown charts to show progress and receive incremental feedback. There are three roles in Scrum. They are Product Owner, Scrum Master and Scrum Team [14].

Ken Schwaber defined "Scrum as a framework for developing complex products and systems. It is grounded in the empirical process and control theory. Scrum employs an iterative and incremental approach to optimize predictability and control risk".

ISSN No.: 2321-3906 (Print) ISSN No.: 2321-7146 (Online) Registration No.: CHAENG/2013/51235
Periodicity: Bi-Annually

In an iterative approach, first, draw the outline of sketch and improve it more and bring it to the middle level and then reaches to the perfect picture. In the Incremental approach, the picture is started by drawing it in a piece by piece over a period of time.

additions, cyclical release and upgraded pattern.

The Scrum follows both the iterative and incremental approaches. In first go it has one part of iteration and the increment and then perfects this piece and then add another

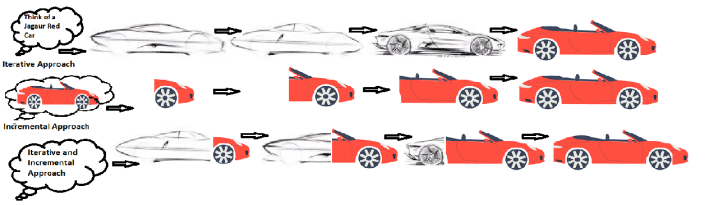

Figure 1: Iterative and incremental approach

Like in Fig. 1, it is displayed an Iterative approach where an idea of designing a red Jaguar car is thought and move towards building up the finished product. In an Incremental approach where A finished product is thought, it starts moving up by making a bit at a time and finally reaches to the Finished Product. Iterative and Incremental approach is a combination of both the approach, where a product designed with the gradual increase in its features

piece and perfect it that is how scrum works. Structure of a scrum is shown in Fig. 2.

The product vision starts with the stakeholders, the customers, the users, the senior manager, and the Management team. The other people involved are *Product Owner*, *Scrum Master*, and the *Scrum Team*. The Process of Scrum starts with the *Product Owner*. He gets the input from the end users, Customers, and the other stakeholders and comes up with the *Product Backlog*. *Product*

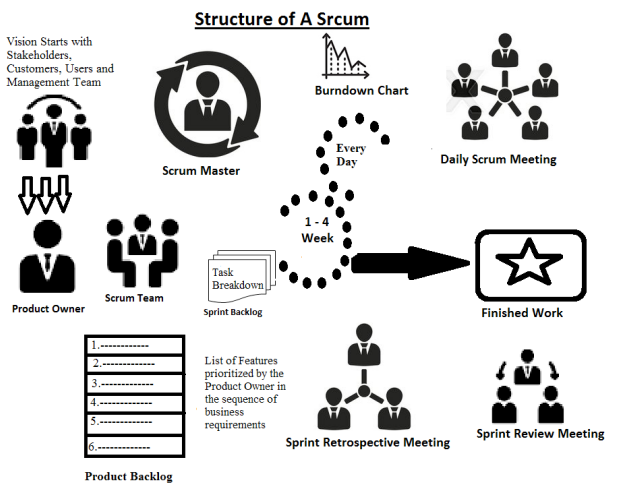

Figure 2: Scrum at a glance

Backlog or list of features prioritized by the *Product Owner* in the sequence of business requirements. So, these features which are also called as *User Stories* are prioritized at the starting of the project, this *Product Backlog* will always be live and the stakeholders keep on adding things to the backlog in the form of *User Stories*. The *Product Owner* is the only person who maintains the *Product Backlog*. *Product Owner* can make changes in the *Product Backlog* whichever change the user want to make.

Scrum Team typically consist of 7 ± 2 People. If the project is too big then break the entire team into multiple Scrum Teams and the teams have to be cross-functional, who have requirement people, designer, coder, testers? The team is self-organized and self-managed and there is no role of the project manager in the scrum project. The team organizes everything within themselves and the team makes the commitment how much they will produce within that sprint is the first level commitment they also commit every day in the daily scrum meeting, what they will do in the next day and what they have done in the previous day. So the commitment happens continuously for self-organizing, self-managed and committing to the product and the responsibilities.

Sprint is referred to the fixed period of time that the team commits to working in the course of development product. The sprint duration is typically 1 to 4 weeks. Once it is fixed for a project say 4 weeks, then it will not be changed in the next sprint until the final deliverable of this particular project. Thus it is the fixed duration of time the team commits to work and deliver the working software and working at the sustainable pace.

From the Product Backlog the Scrum Team pulls out a small portion of it on the highly prioritized items in a manageable chunk into the Sprint Backlog and this happens for every sprint. For example, sprint is of 4 weeks the once in 4 weeks this pull happens. In the Sprint Backlog, the task breaks down into the activity level and then schedule all those activities and then keep on doing those requirements, design, coding, and testing of all those features.

Scrum Meeting is primarily for the Scrum Team, Scrum Master and the Product Owner. The team has a short meeting to update each other. As this is a self-organizing team nobody distributes the work. The team should know what work team has been done everyday updates each other on the progress of the project as well as any impediments or block. They usually stand up so that the meeting is finished faster. The team members stand in a circle looking at each other. Everyone can listen to whatever anyone says. They have to quickly update what is completed and what are further plans to do next and what are the impediments. The Scrum Master notes the blocks and that is the primary responsibility of the Scrum Master. After the Scrum meeting is over, the Scrum Master's responsibility is to do away with all those blocks.

Scrum Master cannot allocate work like a traditional project manager. The Scrum Master role is to protect the team from any external or internal disturbance. He also teaches or guides the team to use the scrum. It is the responsibility of the Scrum Master to train the Product Owner and Scrum Team on the values, principles, and practices the agile. It is not only training if something goes wrong and people do not follow properly, it is the responsibility of the Scrum Master to make sure people follow the processes properly for not only getting trained but also for practicing it.

The everyday team uses a Burndown Chart (Fig. 3). This chart is visual to eve-

ISSN No.: 2321-3906 (Print) ISSN No.: 2321-7146 (Online) Registration No.: CHAENG/2013/51235
Periodicity: Bi-Annually

ryone. It is displayed on the working area of the Scrum Team. In the burndown, there is one planned line and another is an actual line. The Sprint Backlog list for the entire task it is determined primarily how much time is left to be complete this sprint. How much time is required to complete this project is more important. It will estimate whether the project is on time or behind the Schedule.

At the end of every sprint, two reviews are done. First one is the Sprint Review Meeting in which the Product Owner, the Scrum Team, Scrum Master and the Customer and Stakeholders of the product all of them come together and see a demo of the working software. Actual working software or product of functionality of this sprint are shown to the user. This is primarily a Product Review. The user of the product gives feedback. Usually, the Product Owner captures all the feedback and update the Product Backlog.

In the second meeting after the Product Review Meeting is the Retrospective Meetings. In Retrospective Meeting it is only for the Scrum Team, the Scrum Master as well as the Product Owner. These People meet at the end of each Sprint Review at the way of working primarily the process review what went wrong, process perspective, what was right and how to improve. The Sprint Review is the Product Review and Sprint Retrospection is the Process Review. In the Retrospective Meetings, the Entire Scrum process is reviewed and it re-evaluates what was right and what went wrong.

The aim of the team is to complete the 100% of what they have committed to ideally as an increment which should be potentially shipped for the product. The word potentially is very important, the final deliverable is the working software and if required it should be always to ship to the customer and it required customer should be in a position to use the working software. It should be potentially shippable to the client and it needs to be working software, should be given at the end of every sprint. The functionally designed, implemented and fully tested with no major bugs that has to be ensured for every Potentially Shippable Product Increment.

In the end, the team comes out with a product increment. These are not just documents but are all product increments and the working products and keep building are the products piece by piece then integrate them in line with the vision.

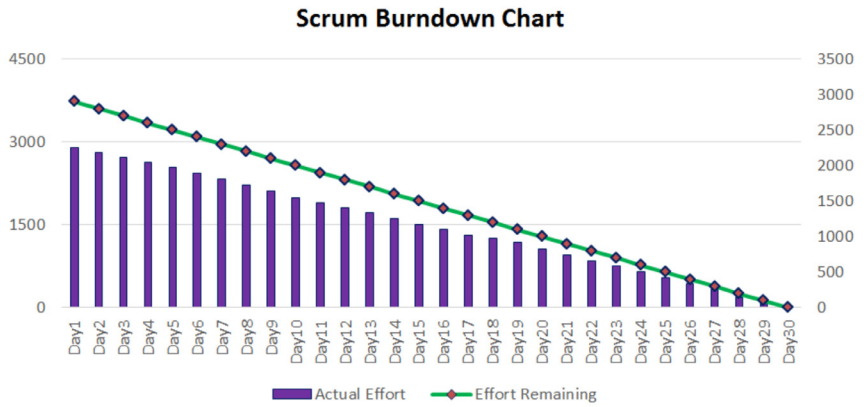

Figure 3: Scrum burndown chart

3. Survey results

The report by Sam Swapn Sinha, [15] CEO, Strategism Inc. on Forbes Technology Council analyses "Does Scrum Live Up to Its Hype? Steve Dunning [15] details how a 100-mpg car was developed from scratch in three months by using Agile in Manufacturing. It also shared that Fortune 100 companies in the United States use Scrum and Agile in Software Projects. 2015 State of Scrum Report shows 95% of the 4,452 people surveyed confirmed they will continue using Scrum in the future [15].

Agile projects are successful three times more often than non-agile projects, according to the 2011 CHAOS report [16] of the Standish Group. The report states "The agile process is the universal remedy for software development project failure. Software applications developed through the agile process have three times the success rate of the traditional waterfall method and a much lower percentage of time and cost overruns." [16] The Standish Group defines project success as on time, on budget, and with all planned features. They do not report how many projects are in their database but say that the results are from projects conducted during 2002- 2010. The following graph (Fig. 4) shows the specified result reported. Sutherland [12] mentioned that Traditional Waterfall Project Management is a predictive process control system which will lead to a large failure rate is 89% in traditional Project Management. It is a totally inappropriate process control mechanism in an environment where 65% of requirement change during development.

In the book "Scrum: The Art of Doing Twice the work in Half the Time" [12] explained how the FBI solved the 9/11 tracking problem by moving from Waterfall to Scrum. The details of the project to integrate all data needed to track terrorists after many years with hundred people and $400M, the US General Accounting Office closed the project because nothing worked out and no end was in sight. Then, The FBI hired an agile CIO and CTO. They put about 10% of the original staff doing Scrum and completed the project for less than $50M. This project is used to run all FBI operations. There are many real-life examples which show that Scrum is the solution for all those areas where the requirements keep on changing along with the market.

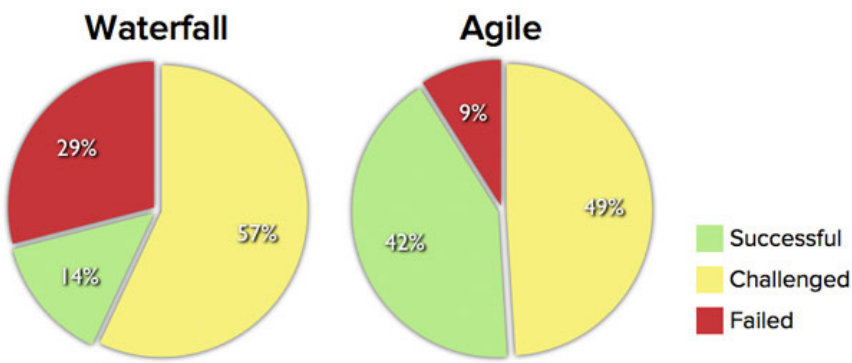

Figure 4: Percentage of successful and challenged projects in waterfall and agile process [16]

A growing body of literature has analyzed that traditional methods are not efficient for changing business needs. Long time between project start and go live causes a gap between initial solution blueprint and actual user requirements at the end of the project. A case study was done in a large telecommunications company (350 BI users) and the results of pilot research provided in the three large companies: Media, Digital and Insurance. Both studies prove that agile methods might be more effective in BI projects from an end-user perspective and give first results and added value in a much shorter time compared to a traditional approach [17].

Few reports on large scale agile transformations reveal that many large organizations are adopting agile software development as part of their continued push towards higher flexibility and shorter lead time [18]. A systematic study was carried out on how Ericsson introduced agile in a new R&D product development program developing a XaaS platform and a related set of services, while simultaneously scaling it up aggressively [18].

A recent review of the literature on agile software development in different industrial sectors is evident nowadays. Financial Institution needs to acts faster in response to quick changes in their business environment. This is relative because of the new generation of financial technology companies that have exhibited substantial transformation in time to market and accelerate software development. To stand with the pace of such fintech (Financial technology) companies, financial institutions are concentrating on implementing agile practices in order to advance their software development processes [19].

In recent years, the rapid development of banks is more and more dependent on the function of a bank software system. The demand for software development is constantly changing, which makes some banks software development team unable to adapt to frequent demand changes. In order to adapt to the needs of frequent changes, more and more software development teams use agile software development methods [20].

In Production System Engineering (PSE), many projects conceptually follow the plan of traditional waterfall processes with sequential process steps and limited security activities, while engineers actually work in parallel and distributed groups following a Round-Trip-Engineering (RTE) process. Unfortunately, the applied RTE process in PSE is a coarse-grained that is often data are exchanged via E-Mail and integrated seldom and inefficiently as the RTE process is not well supported by methods and tools that facilitate efficient and secure data exchange. Thus, there is a need for frequent synchronization in a secure way to enable engineers building on a stable and baseline of engineering data. The system is built on Scrum, as an established agile engineering process, and security best practices to support flexible and secure RTE processes. Further results show that the augmented RTE process can provide string benefits from agile practices for the collaboration of engineers in PSE environments [21].

4. Scrum metrics

The main objective of scrum metrics lies in Predictable Software Delivery and the maximum value to the Customer. Goals of Scrum can be divided into three Category based on Key Performance Indexes.

(i) To measure the deliverable of the Scrum Tea and understand how much value is being delivered.

(ii) To measure the effectiveness of the

ISSN No.: 2321-3906 (Print) ISSN No.: 2321-7146 (Online) Registration No.: CHAENG/2013/51235
Periodicity: Bi-Annually

Scrum Team; its contribution to the business in terms of ROI, time to market, etc.

(iii) To measure the Scrum Team itself in order to gauge its health and catch problems like team turnover, attrition, and dissatisfied developers.

4.1 Scrum metrics – Measuring deliverable

The following metrics help in measuring the work done by the Scrum Team and value delivered to Customer.

4.1.1 Sprint goal success

A Sprint Goal is the objective that is set to achieve after completion of each sprint. Sprint Goals are discussed between the Product Owner and the Scrum Team. Sprint goal must be specified and measurable. A sprint goal process is shown in Fig. 5. For example, delivering an X Feature, Check if the architecture enables the desired performance (addressing a risk), and Test if a user is willing to register before using the product features (testing an assumption). Selecting a Sprint Goal can be possible if the team will answer the following three questions:

a. Why do we carry out sprint?
b. How do we reach its goal?
c. How do we know that goal has been met?

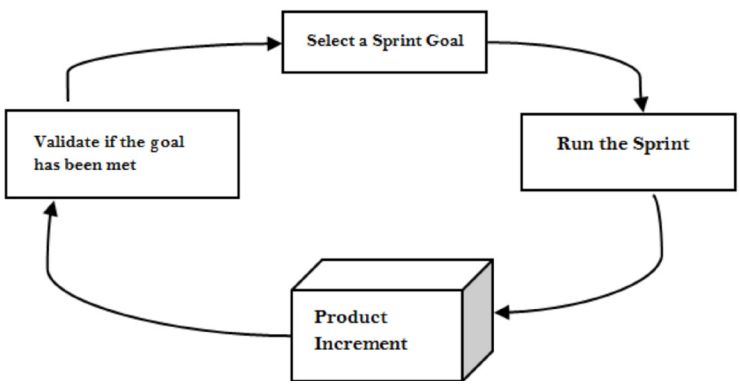

Figure 5: Sprint goal process

$$\text{Sprint Goal Success Percentage} = \frac{\textit{No. of sprint goal achieve}}{\textit{Total no. of sprints}} \times 100$$

The greater percentage value indicates that frequent business objectives are met.

4.1.2 Escaped defects and defect density

It is the total no. of bugs faced by the users. A Scrum team try to avoid escaped defects by full test cases. But a trend of Escaped defects indicates a good product quality. It is calculated as shown in equation 1.

$$\text{Escaped Defects Percentage} = \frac{\textit{Total no. of bugs faced by the users}}{\textit{Total no. of fully tested units}} \times 100 \quad (1)$$

Defect Density is measured by no. of defects per software size. For example - Per Lines of code (LOC). It is calculated as shown in equation 2.

$$\text{Defect Density} = \frac{\textit{Total no. of defects}}{\textit{Total no. of lines of code}} \quad (2)$$

It is more important for fast-moving projects to check if the growth in defects is "normal" given the growth of the underlying codebase.

ISSN No.: 2321-3906 (Print) ISSN No.: 2321-7146 (Online) Registration No.: CHAENG/2013/51235
Periodicity: Bi-Annually

4.1.3 Team velocity

It is measured by calculating no. of units of software developed by a team in a sprint. It can be used for planning and estimating no. of sprints. As the name suggests, it is a metric for Scrum Teams to leverage for its internal purpose for continuous improvements.

For example, as shown in Fig. 6, a team planned to complete 20 story points in their first sprint. They completed 15 story points and rolled 5 story points over the next sprint, in the second sprint they planned to complete 10 story points. They completed 15 story points (including 5 story point of the first sprint). In third sprint they planned to complete 25 story points, but completed 20 story points, in the fourth sprint they planned to complete 30 story points and completed 32 story points (including 2 story point of the third sprint). In the fifth sprint, they planned to complete 25 story points and they completed 25 story points. So, 21.4 is their average velocity.

This velocity is used to make predictions. By knowing the velocity, team members can recognize an estimate of how long the project will take to complete.

It should not be used for any other purpose; otherwise, the benefits of Scrum will be lost by the team and the organizations.

4.1.4 The sprint burndown chart

The Burndown Chart represents the progress within a Sprint. It depicts the no of hours remaining to complete the stores planned for the current sprints, for each day during the Sprint. It shows whether the team is on schedule to complete the sprint objective or not.

To create this graph, calculate how much work remains by adding the sprint backlog evaluates every day of the sprint. The amount of work remaining for a sprint is the sum of the work remaining for the whole sprint backlog. Then monitoring these sums day by day and use them to create a graph that shows the work remaining over time.

In order to create a graph which is shown in figure 7, the duration is taken: - 5 Days, Sprint Backlog: - 8 Tasks and velocity: - 80 available hours. That is 80 hours over 5 days equating to 16 hours a day. To create the project burn-down chart, the data needs to be collected as a daily running total starting with 80 hours than 64 hours left at the end of day 1, 48 hours left at the end of day 2, etc. which is shown in Table 1.

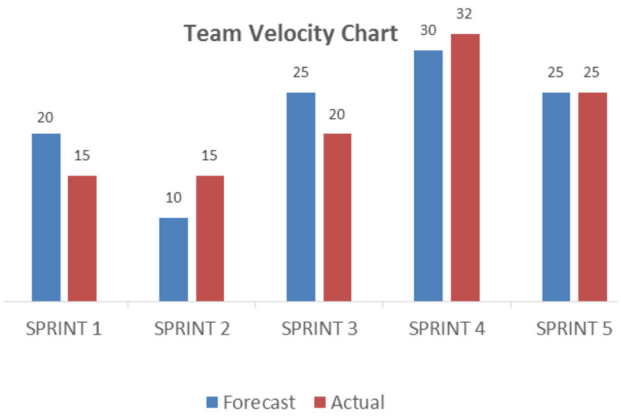

Figure 6: Team velocity chart

ISSN No.: 2321-3906 (Print) ISSN No.: 2321-7146 (Online) Registration No.: CHAENG/2013/51235
Periodicity: Bi-Annually

Table 1: Burndown – Estimate effort

Day	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5
Effort Remaining	80	64	48	32	16	0

The daily progress (Table 2) is then collected in the table against each task. The value collected for each day is the estimated effort to complete the task instead of actual effort.

4.2 Scrum metrics – Measuring effectiveness

The following metrics measure the effectiveness of Scrum Teams in terms of meeting business goals.

Table 2: Burndown – Daily progress

Task	Hours	Day 1	Day 2	Day 3	Day 4	Day 5
Task 1	10	3	2	0	1	4
Task 2	10	3	2	0	1	4
Task 3	10	3	2	0	1	4
Task 4	10	3	2	0	1	4
Task 5	10	3	2	0	1	4
Task 6	10	3	2	0	1	4
Task 7	10	3	2	0	1	4
Task 8	10	3	2	0	1	4

The total remaining effort needs to be collected at the end of each day. This is the total (sum) of all the estimated time remaining at the end of each day as shown in Table 3.

4.2.1 Time to market

Depending upon the Project, it is the time a project takes to start providing value to the Customer, which can be calculated by taking the length of the no. of sprints before a Scrum Team releases to production or the

Table 3: Burndown- Final dataset

	Day 1	Day 2	Day 3	Day 4	Day 5
Actual effort	56	40	40	32	0
Effort Remaining	64	48	32	16	0

When the data is ready, then project burndown chart can be created using a line chart option of Excel (Fig.7).

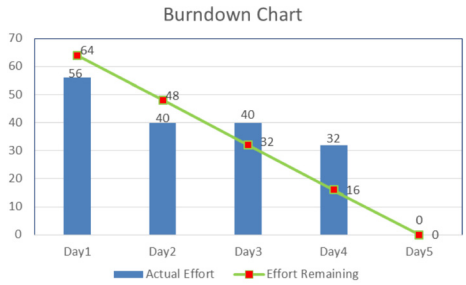

Figure 7: Sprint burndown chart

ISSN No.: 2321-3906 (Print) ISSN No.: 2321-7146 (Online) Registration No.: CHAENG/2013/51235
Periodicity: Bi-Annually

time it takes to start generating revenues which can be calculated by taking the length of the no. of sprints before its release plus depending on the organization's alpha and beta testing strategy.

It can be evaluated with the help of the Schedule Performance Index as shown in equation 3 which is a ration of total original authorized duration versus total final project duration. The ability to accurately forecast schedule helps to meet Time to Market and shows the accuracy of Schedule estimating.

$$\text{Schedule Performance Index} = \frac{\textit{Total Original Authorized Duration}}{\textit{Total Final Project Duration}} \quad (3)$$

4.2.2 Return on investment (ROI)

Return on Investment for a scrum project calculates the Net Benefits generated from a product versus the cost of the sprints required to develop it. Then multiply it by 100 will determine the percentage return for every invested as shown in equation 4.

$$\text{Return on Investment (ROI)} = \frac{\textit{Net benefit generated from a product}}{\textit{Cost of the sprints required to develop that prodcut}} \times 100 \quad (4)$$

To measure net benefits, placing a currency vale on each unit of data and other sources are a variety of measures like a contribution to profit saving of costs, increase in quantity of output and quality of improvements. Cost includes the costs to design and develop or maintain the product, cost of product management initiative, cost of resources, cost of travel and expenses, the cost to train and other overhead costs, etc.

In Scrum, ROI starts generating very fast as compared to the traditional development methods, as working software can be delivered to the Customer very early. With each sprint added features increase the growth in revenue.

4.2.3 Capital redeployment

It is measured to check whether this Scrum Project is worthwhile or not. In case if it is not worthwhile, then the team should be deployed to other more profitable projects. It can be calculated as.

The revenue value of the remaining items in the projects backlog marked as **V**, The actual cost of the sprint needed to complete these items marked as **AC**, The opportunity cost of alternative product work the team could do, marked as **OC**

When**, V< AC + OC** then, the project should end and the team redeployed to other projects.

4.2.4 Customer satisfaction

It means customer expectations are met. This requires the conformance to requirement and Fitness for use. It can be calculated as shown in equation 5.

$$\text{Customer Satisfaction Score} = \frac{\textit{Total Survey Point Score}}{\textit{Total Questions}} \times 100 \quad (5)$$

The Customer Satisfaction Index is an index comprising hard measures of Customer buying/use behavior and soft measures of customer opinions or feelings.

The index is weighted based on how important each value is in determining customer overall customer satisfaction and buying/use behavior. Includes measures such as repeat and lost a customer (30%), Revenue from existing customers (15%), Market share (15%), Customer Satisfaction Survey results (20%), Complaints/Returns (10%) and Project-specific surveys (10%).

4.3 Scrum metrics – Monitoring the scrum team

These metrics help the Scrum Team in monitoring its activity and identify problems before they impact development.

4.3.1 Daily scrum and sprint retrospective

Daily Scrum, improve the communication

ISSN No.: 2321-3906 (Print) ISSN No.: 2321-7146 (Online) Registration No.: CHAENG/2013/51235
Periodicity: Bi-Annually

between the team, identify impediments so as to find an early solution for it, highlight and promote quick decision making and improve the team's level of knowledge. The sprint retrospective is an opportunity for the Scrum Team to introspect improve within the Scrum process so as to make the next sprint outcome more effective. So, these two events, if carried out regularly with the well-documented conclusion, can provide an important qualitative measurement of team progress and process health.

4.3.2 Team satisfaction

It is an important metric to survey as if done periodically will notify how satisfied team is with their work, can provide warning signal about culture issues, team conflicts issues, team conflicts or process issues. It can be measured through a survey which contains questions based upon their working culture, team coordination and process integrations and environment, etc. It can have a scale from 1 to 100. It is calculated as shown in equation 6.

$$\text{Team Satisfaction} = \frac{\textit{Total Survey Point Score}}{\textit{Total Questions}} \times 100 \qquad (6)$$

4.3.2 Team member turnover

Team Member turnover means replacement of team members in a Scrum team. Low turnover percentage in Scrum team indicated a healthy environment and increase in the overall company turnover, while a high percentage indicates the opposite of it. It can be calculated as shown in equation 7.

$$\text{Team Member Turnover Percentage} = \frac{\textit{No. of team members replaced}}{\textit{Total no. of team members}} \times 100 \qquad (7)$$

4.3.3 Team productivity

Team Productivity, it is also one of the most important metrics to evaluate. It indicates whether the cost involved to have people is worth or not. The direct method to measure is to use revenue per employee as the key metric and then divide revenue per employee by the average fully burdened salary per employee yield a ratio. This ratio is the average-per-employee "Product Ratio" for the organization as a whole. It is calculated as shown in equation 8.

$$\text{Productive Ratio} = \frac{\textit{Revenue per employee}}{\textit{Average salary per employee}} \qquad (8)$$

4.4 Scrum reporting- Metric which reports to stakeholders

A stakeholder is interested in knowing the progress of scrum project and wanted to know whether it is on track. Following metrics help them to communicate this and explain deviations from the expected project path.

1. Sprint and release burndown:- It gives a view of the progress at a glance
2. Sprint Velocity: - A historical review of how much value have been delivering.
3. Scope Change: - The number of stories added to the project during the release, which is often a cause of delays.
4. Team Capacity: - No. of developers in a team is on a full time, Work capacity been affected by actions or sick leaves. Developers pulled off to side projects.
5. Escaped Defects: - It provides a picture of software performing in the production.

5. Conclusion

This paper has explained that Scrum has the power to transform project management across every industry or business. The evidence from the survey results study implies that by using Scrum. The team become more agile and succeeded in a way to react more quickly and respond more accurately to the inevitable change that comes their way.

This paper has highlighted the importance of Scrum Metrics. Scrum Metrics

ISSN No.: 2321-3906 (Print) ISSN No.: 2321-7146 (Online) Registration No.: CHAENG/2013/51235
Periodicity: Bi-Annually

can have a specific purpose and importance in an organization and teams. It is not merely a number, but it can provide a trend which has to observe to make it impactful. With the appropriate use of Scrum metrics, organizations can link each measure to a well-articulated goal that team the understands.

Scrum metrics provides a powerful tool which can be used to make improvements and help businesses to focus their human and other resources. Organizations if using Scrum metrics, can understand the value in watching the trends, monitoring in smaller durations in order to understand individual, management and organization influences and helps in making the decision to accelerate and decelerate these influences.

Taken together Scrum and Scrum Metrics, the present findings might suggest Scrum could also be applied in any field even across life in general. It is recommended try to plan Scrum, and use Scrum Metrics in order to stay focused, collaborating, and communicating, to accomplish what truly needs to be done – successfully.

References

[1] M. Kramer, "Best practices in systems development lifecycle: An analyses based on the waterfall model", *Review of Business & Finance Studies*, vol. 9, no. 1, pp. 77-84, 2018.

[2] J. Yu, "Research process on software development model", *IOP Conference Series: Materials Science and Engineering*, 394, 2018.

[3] M. Kumara and E. Rashid, "An efficient software development life cycle model for developing software project", *International Journal of Education and Management Engineering*, vol. 8, no. 6, pp. 59-68, 2018.

[4] R. Srikrishna, "*The current state of outsourcing relationship: the Watermelon Effect*", 2015. Accessed: February 8 2019 [Online]. Available: https://www.raconteur.net/business-innovatin/current-state-of-outsourcing-relations-the-watermelon-effect.

[5] C. Weisert, "*Waterfall methodology: there's no such things*", 2003. Accessed: February 8 2019 [Online]. Available: https://www.idnews.com/waterfall.html.

[6] C. Ziegler, "*Android: A visual history*", 2011. Accessed: February 8 2019 [Online]. Available: http://www.theverge.com/2011/12/7/2585779/andriod-10th-anniversary-google-history-pie-oreo-noghat-cupcake.

[7] A. Ahmed, S. Ahmad, N. Ehsan *et al.*, "Agile software development: Impact on productivity and quality", *IEEE IC-MIT*, 2010.

[8] J. A. Livermore, "Factors that impact implementing an agile software development methodology", *IEEE SoutheastCon*, pp. 85-85, 2007.

[9] Beck *et. al*, "*Lifecycle starts here: Manifesto for agile software development*", 2001. Accessed: February 8 2019 [Online] Available: http://agilemanifesto.org.

[10] A. Qumer and B. Henderson-Seller, "An evaluation of the degree of agility in six methods and its applicability for method engineering", *Elsevier Information and Software Technology*, vol. 50, no. 4, pp. 20-29, 2008.

[11] D. West, M. Gilpin, T. Grant and A. Anderson, "Water-Scrum-Fall is the reality of agile for most organizations todays", *Technical Report Forrester Research Inc.*, 2011.

[12] J. Sutherland, "*Why waterfall doesn't Scale*, 2017. Accessed: February 8 2019 [Online]. Available: https://www.scru-

minc.com/waterfall-doesn't-scale/.

[13] K. Schwaber, "Scrum development process", *Proceedings of the 10th Annual ACM Conference on Object-Oriented Programming Systems, Languages and Applications OOPSLA*, Texas, USA, pp.117-134, 1995.

[14] K. Schwaber and J. Sutherland, "*The definitive guide to Scrum: The rules of the game*, 2013. Accessed: February 8 2019 [Online]. Available: http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-us.pdf.

[15] S. S. Sinha, "*Does scum live up to its hype?*", 2017. Accessed: February 8 2019 [Online]. Available: https://www.fobes.com/sites/forbestechcouncil/people/samswapnsinha.

[16] M. Cohn, "*Agile succeeds three times more often than waterfall*", 2012. Accessed: February 8 2019 [Online] Available: https://www.mountaingoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall.

[17] J. Kisielnicki and M. A. Misiak, "Effectiveness of agile compared to waterfall implementation methods in IT projects: Analysis based on business intelligence projects", *Foundation of Management*, vol. 9, 2017.

[18] M. Passivaara and B. Behm, C. Lassenius *et al*, "Large-scale agile transformation at Ericsson: a case study", *Empir Software Eng*, 23:2550, 2018.

[19] E. Kilu, F. Milani, E. Scott and D. Pfahl, "Agile software process improvement by learning from financial and fintech companies: lhv bank case study", *Software Quality: The Complexity and Challenges of Software Engineering and Software Quality in the Cloud.* SWQD, Lecture Notes in Business Information Processing, vol 338. Springer, Cham, 2019.

[20] E. Qi, H. Bi, X. Bi, "Study on agile software development based on scrum method", *Proceeding of the 24th International Conference on Industrial Engineering and Engineering Management*, Springer, Singapore, 2018.

[21] D. Winkler, F. Rinker and P. Kieseberg, "Towards a flexible and secure round-tri-engineering process for production systems engineering with agile practices", *Software Quality: The Complexity and Challenges of Software Engineering and Software Quality in the Cloud.* SWQD, Lecture Notes in Business Information Processing., vol 338. Springer, Cham, 2019.