Impact of Story Point Estimation on Product using Metrics in Scrum Development Process

Ali Raza Ahmed
Dept. of Software Engineering
Bahria University Islamabad,
Pakistan

Muhammad Tayyab
Dept. of Software Engineering,
Bahria University Islamabad,
Pakistan

Dr. Shahid Nazir Bhatti
Dept. of Software Engineering,
Bahria University Islamabad,
Pakistan

Dr. Abdullah J. Alzahrani
Department of CCSE,
University of Hail (UoH), KSA

Dr. Muhammad Imran Babar
Department of Computer Sciences,
APCOMS, Rawalpindi, Pakistan

***Abstract*—Agile Software Development techniques are worldwide accepted, regardless of the definition of agile we all must agree with the fact that agile is maturing day by day, suppliers of software systems are moving away from traditional waterfall techniques and other development practices in favor of agile methods. There are numerous types of methodologies, domains/ methods in agile for which are to be selected according to the current situation and demand of the current project. As a case scenario in the following research will discuss scrum as a development technique in which we will focus on the effort estimation(s) and their effects by discussing distinct metrics. Mainly estimation refers directly to cost, time and complexity during the life cycle of project. Metrics will help the teams to better understand the development progress and building releasing (releases) of software easier in a fluent and robust way. The following paper thus identifies aspects mainly ignored by the development team(s) during estimation.**

Keywords—product backlog; sprint backlog; backlog Item; front end designer; product Owner; agile software development; scrum master; product owner; sprint planning; velocity chart; Agile methodology; Effort Estimation; Story Points Estimation

I. Introduction

Scrum is a light weight process having a series of fixed length iterations known as sprints. In February 2001, about 17 developers published a Manifesto for Agile Methodology after discussion [16]. Actually they wanted a light weight method in which teams can work easily and comfortably.

According to Agile Alliance, the definition of agile based software development is that Agile based software development is a term for an arrangement of strategies and practices in view of the qualities and standards communicated in the Agile Manifesto.

Scrum simply gave our teams the self-determination that we needed to do our best work while helping our executives to get the business results they wants. Scrum has multiple basic roles and Team works with distinct artifacts PBL (Product Backlog) and SBL (Sprint Backlog). Scrum is assisted by the Scrum ceremonies that keep every team member aware of what is going on with the product, especially with the key reason of client awareness. Scrum is a transparent iterative technique that is openly disclosed to every member of the product, so here follows the purpose of metrics. Metrics don’t take effect until the meaningful estimation of every BL-Item (Product Backlog Item) or SB-Item (Sprint Backlog Item) is not done. Metrics plays a key role towards the Journey starting from product planning till product success.

The technical question arises here is that, before starting every project, how a product owner can tells the customer that, how much cost is required to complete the project and how much time it will take? To answer this question, a lot of research has been made and researchers presented many techniques and methods to calculate the effort, cost and duration required for a software system. “Effort and cost estimation” is the terms that is used by researchers to calculate the cost and duration required to complete the project. But software project in agile software development still fails. Failure of the projects has many reasons:

- Poor estimation (cost and effort)
- Neglecting certain risk factors that can be occurred during project life cycle

In traditional methodologies metrics are playing key role to keep everyone focused toward the goal, Our research will conclude that whether the estimations and measuring metrics being used in scrum following industries produce key benefits that keep the team on track or either they help the team to have a clear understanding of Sprint Scope, Goal, and Quality Delivered to Customers. In traditional approaches, Metrics depicts the team’s ability to understand and predict each and every member’s capabilities. Here our focus is on correct Estimation technique and the metrics used in development market better results or not.

In this paper our focus is on the effort estimation and its effects that are used for transforming the requirements, skills and equipment’s in terms of how much the project will cost and how much effort and time it will take to done. In simple we will discuss better way to estimate and will find how it will contribute to the product development and client satisfaction.

The rest of the paper has following section: Section II provides the Methods and Materials based on the existing research in this domain. Section III highlight the existing problems in this, Section IV presents Scrum Methodology, section V Software Metrics in Scrum, Section VI provides the Validation Results in this while section VII concludes the research and outlines future work.

II. Methods and Materials

Research work has been done to improve the estimation of cost and effort but those work lakes the capability of Metrics. Rashmi Popli et al. [2] proposed the solution in which they determined total number of stories according to the customers and users requirements to find out total story points after that they calculated effort, cost and duration required for the project by determining the introductory speed of the project. They also proposed activity diagram for agile estimation.

Evita Coelho et al. [3] discussed estimation based on story points. According to authors story point estimation team should know the acceptance criteria on which the project will be approved. Team should also focus every new feature and estimate the features that can be selected in upcoming release and then team should identify the length of iteration. Author's states to estimate the story points, velocity of the team should also be calculated. They further discussed 3 techniques to measure the velocity of a team. These techniques are:

- Measuring the velocity of the team from historical data
- Run some iterations and then measure the velocity by using the result of those iterations
- If there is no historical data is present then forecasting can be used

After measuring velocity, team should prioritize user stories on the basis of some criteria after that, estimate delivery date, from the length of selected iteration.

Nils C. Haugen [1] studied empirically of using planning poker for the estimation of user stories. He focused only extreme programming and observed a team working on a project in extreme programming. Team played a planning poker game. He also took an unstructured group estimation process and then compared the results.

Kjetil Moløkken-Østvold [6] et al. combined their estimates with planning poker. They discussed different techniques and prepared research questions. Author proposed solution that will studied on a company and then they calculate estimation accuracy by using a formula on the basis of which they give results.

On the other side metrics measure the software development process and team's quality of work. Metrics allow you to better understand the progress of project and assist in better management of development process [16]. Metrics provide us guide lines about what we are doing and what we will do [24]. Downey and Sutherland [16] came up with a number of fundamental measurements which are significant and can be utilized for management. Metrics are categorized into three groups as predictability in terms of project, team productivity, and quality of product. These metrics provide several favors toward ASD(Agile Software Development) such as better tracking of project progress, monitoring product quality, prediction and project management [28].So, it is convenient to find out the suitable group of metrics, criteria's, and measurement scales, which are best to incorporate with ASD [30].

III. Problems in Existing Methods

All existing techniques ignore the developer skills in task estimation. Agile team members came up with different skill sets [16]. All have their own way of thinking and level of experience. If we estimate tasks on the basis of senior resources although that is against the team work and scrum but industry is working like that, reality is not every task will be done by senior members. Generally, if a senior member will take 1 day to complete a certain task then junior member can take 2 days for that task which will cause the sprint delay and ultimately it became difficult for team to meat project deadlines. Existing techniques did not talk about QA personals and PO but focus point is on developers. We visited some organizations, which were following agile; we asked questions from team members of different organizations for their estimation of story points. According to the answers we got, we noticed that this is the serious problem that in organizations, story points for testing are not added.

IV. Scrum Methodology

Scrum is an agile framework for completing the complex and innovative scope of work.This frame- work is extremely simple. Scrum always gives freedom to business so that one can customize the process according to the requirement of product and the market. It cuts off the complexity of management and let the team focus on the product delivery. Scrum has three basic roles: SM, PO and Team

- ***Product Owner:*** The PO is the person who has vision for the product to be developed and the authority. PO is responsible for continuous team work and priority. It is the responsibility of PO to full fill the PBL and SBL.
- ***Scrum Master:*** The SM is just an expediter for both team and product owner. The SM does not manage the team. The SM works to tackle with impediments that cause failure in achieving sprint goals.
- ***Team:*** The dev team is self-organized to complete their committed work. A Scrum development team consists of about seven fully dedicated members (officially 3-9) for projects in software; a typical team includes a mix of software engineers that generally are architects, programmers, analysts, QA experts, testers, FED and UI designers.

There are different Artifacts of scrum which are follows

- *Product Backlog* (The PBL is the cumulative list of desired deliverables for the product).
- *Sprint Backlog* (The SBL is the team's to-do list for the sprint prioritized by PO).
- *Product Increment* (At the end of the sprint new increment started).

There are also 4 types of Scrum Ceremonies which are follows

- Sprint Planning Meeting (Time Frame =4hours (2 weeks sprint) main goal is what and How).
- Sprint Review/Demonstration Meeting (Time Frame=1.5hours (2 weeks sprint) main goal is to demonstrate last sprint work to PO).
- Daily Scrum Meeting (Time Frame =15mins brief progress on sprint by each individual team player).
- Sprint Retrospective meeting (Time Frame=15-20mins team discussion what good and bad happened in the last sprint and do we need any improvements).

Scrum gives free hands to manage timeframes according to the business needs.

V. Software Metrics in Scrum

Generally software metrics quantify the characteristics of software in terms of LOC (source lines of code), complexity (code), Bugs per lines of code, coupling, and cohesion. But here when teams are working in sprints, our main focus is on productivity and defining metrics to measure that. The team has a list of items to work on in the PBL which are picked during the sprint planning meeting, their requirements are refined and when the team is satisfied with their understanding of the tasks, it commits them for the upcoming sprint cycle. By using each sprint item the team is able to develop metrics to determine that either they are on track to accomplish the goal or have they deviated from the measured path. It makes releasing software very easy and pre- sighted. Fig.1 illustrates the process of scrum generally followed in industries and highlights the existence of metrics and estimation in process.

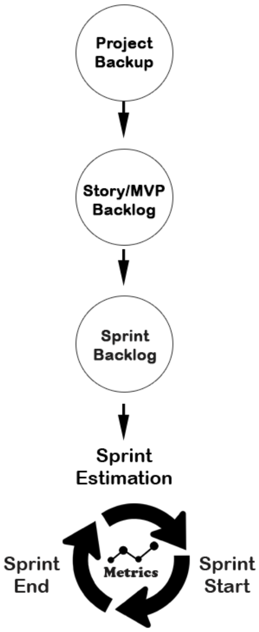

Fig. 1. According to scrum approach, sprint estimation and metrics

We have some metrics described below used in scrum widely to evaluate either they are full filling the needs or not. Every single metric mentioned below have their own quality of being suited to serve a purpose to achieve the sprint goal

- Sprint Burn Down
- Epic Burn Down
- Velocity Chart
- Control Chart

A. *Sprint Burn Down:*

Scrum team gets organized in development using this metrics. Before starting sprint in sprint planning meeting team estimates that how much work they can complete in a sprint (e.g. 2 weeks).This report tracks the work remaining to be completed for the sprint .The x-axis represents time duration of a sprint, and the y-axis refers to the amount of work left to complete in the form of story points (team measures complexity of each task). Goal is to complete all committed work by the sprint end.

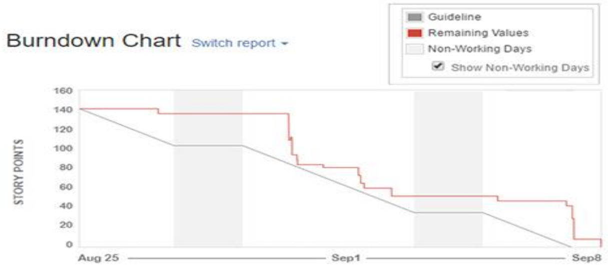

Fig. 2. Sprint burn down chart using JIRA following scrum approach

We have here an example of a sprint burn down chart generated via Jira, the sprint burn down chart depicts that scrum team commits to 140 story points, here gray line is the mean line in between story points and sprint end date known as a guide line for the team. Our main focus is the red line that is showing actual status of the sprint, every day the team works on the tasks and the graph displays a downward trend. It dips only when the task from the sprint move to done status according to the definition of done by the PO.

This is the point where the role of SM is, the red line has different meanings according to behavior

- If this line goes straight that means the team has a blocker they need assistance in their sprint as shown in the Fig. 2 i.e. 25th to 29th august.
- If graph line suddenly falls at the start of the sprint means team did wrong estimation in Sprint planning meeting.
- The ideal graph is shown in Fig.2. Is from 29th august till the end of the sprint.

The Sprint burn down chart is extremely helpful for PO and SM to keep eye on team's progress.

B. *Epic Burn Down:*

Epic Burn down chart tracks the progress of development over a larger bulk of work than the sprint burn down, and guide development for both scrum and Kanban teams. As seen in fig

2, we have the original estimate and the completed work, sprint by sprint for a specific epic .Epic is a group of tasks related to one module let's call it Report.so, Report is one of the product module that takes several sprints to complete.

With the injection of fluctuating requirements into a previously-defined project, it will help us in that situation and makes everyone clear/aware of the flow of work.

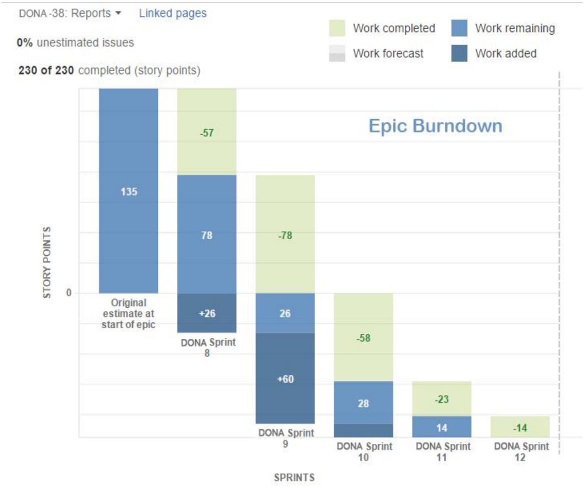

Fig. 3. Epic burn down chart using JIRA following scrum approach

Fig.3. graph shows an epic of reports that consist of 135 story points that are highlighted in the first bar among 6 sprint bars .Starting with sprint 8 of the project DONA here the sprint have 26 story points tasks newly added in and 57 got completed with 78 remaining out of 135 story points. This process is followed until all stories coming under the umbrella of Report epic got completed. As every metric have their own benefits and drawbacks this metrics doesn't show that how much story point commit by the team.

C. Velocity Chart:

Velocity Chart shows the average amount of work a scrum team completes during a sprint, measured in terms of story points and is very useful for predictions, it shows a graph between what is committed and what is delivered. It is the most accurate prediction for the team and PO about their capabilities.

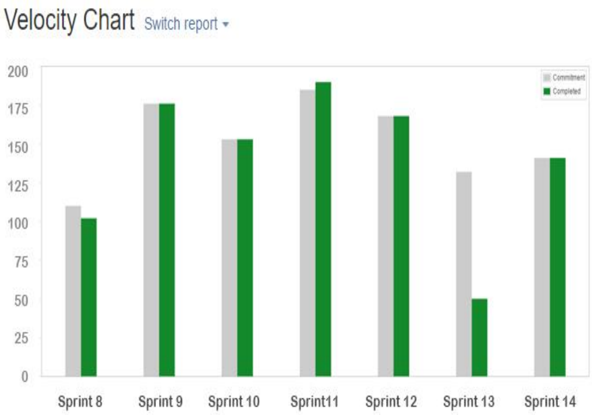

Fig. 4. Velocity chart using JIRA following scrum approach

Here are some stats about this graph, VC (Velocity Chart) is a forecasting bridge of the product, and at y-axis it shows story points along with x-axis that shows sprints. The graph Grey bar'-s shows what was the commitment of team about sprint and green is what team delivered after sprint end. The key usage of this graph is to forecast how much team is capable to the SM .So, next time when the team is over estimating or under estimating sprint SM alerts the team about their capabilities. On the other hand, it helps SM about resource planning.

C. Control Chart:

Control chart focuses on the cycle time for the individual issues. Cycle Time is the total time taken to work on an issue from start till the end [5] including any extra time needed to re-open and work on it again that the team which has shorter cycle time for the issues are more likely to make a smoother and reliable delivery [5]. So this is the primary metrics of each team.

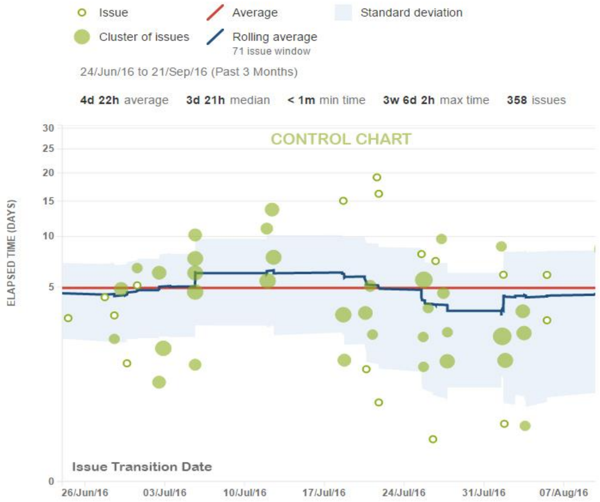

Fig. 5. Control chart using JIRA following scrum approach

Fig 5 depicts the Control chart graph, the filled in circles represent a cluster of issues and the hollow circle represent individual issues. Clicking on each issue opens up a pop up where the user can see the time the issues spent in each state (defined as per the organization's process).The control chart helps to identify the bottlenecks or any blockers the team is facing. The teams which are able to reduce the time each issue spends in a particular state are able to ensure a smoother delivery. This metrics helps SM and PO in several ways some are mentioned as:

- Analyze teams past performance.
- Measure the effect of change in process for the team.
- Provide PO, SM, and client with visibility of their team's performance.

VI. Validation

We took 2 development teams from 2 organizations. We can name teams as Team A and Team B. Both teams were following scrum. Normal working hours for a team member was 7 hours in a day. For both teams, sprint time period was 2 weeks which means each member of the teams has 10 working days.

We observed both the teams in backlog refinement and planning meeting. Teams estimated their tasks according to Fibonacci series. They started their sprint and worked on their tasks, we observed both teams during sprint. At the end of the sprint, we noticed that team members of both teams put extra efforts to complete sprint tasks. Team A was working for extra hours in last 2 days. Team B worked for extra hours on very last day for bug fixing. We tried to find out the reason behind extra efforts to complete their tasks. We took tasks estimates from both teams. Both teams were using Jira, a management tool. We gather data from sprint burn down chart using Jira. Fig. 6 and Fig. 7 respectively indicates Team A and Team B.

The tasks with their estimations for both teams are given in Table I.

TABLE I. TASKS VS ESTIMATIONS FOR BOTH TEAMS

Team A		**Team B**	
Task No.	Estimates	Task No.	Estimates
1	5	1	8
2	5	2	5
3	8	3	5
4	8	4	13
5	8	5	5
6	5	6	8
7	13	7	8
8	2	8	8
9	8	9	5
10	5	10	8
11	3	11	8
12	8	12	13
13	8		
14	5		
15	3		
16	2		

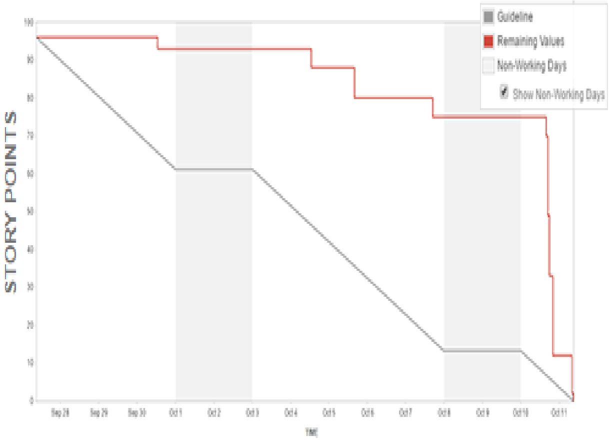

Fig. 6. Burn down chart for Team A

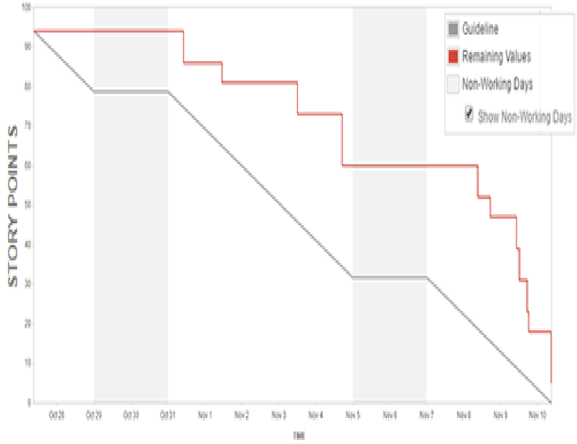

Fig. 7. Burn down chart for Team B

In above figures, grey colored line indicates the ideal state of sprint and red line shows the progress of team during the sprint. Red line progresses when a task moved to done status. In both figures progress state (red line) of sprint, is so much different from ideal line.

We used these burn down charts to compare estimations with the original effort done by team members on the tasks. We tried to identify why the tasks took so much time to complete. Meanwhile it comes up in noticed that tasks were completed by developers in time and complexity of tasks for developers was similar to the estimations given in Table I. After discussion with team we identified that developers completed their tasks but task's status cannot be changed to done because of testing by quality engineer and product owner, which consumes a lot of time. After proper testing Bugs were also identified and developers also fixed the bugs identified by product owner and quality assurance engineer. Here we identified that estimation done by both teams for all task lacks the effort of quality assurance engineer and product owner but the teams ignored these efforts while estimating the tasks.

For 2nd sprint, we asked both the teams to consider effort put by quality assurance engineer and product owner to test the task. Now developer decided their story points for every task and then scrum master ask from QA engineer and product owner to tell the team about their efforts to test the task. For each task, story points have also been added for QA engineer and PO. After sprint planning meeting teams started their sprints and we started monitoring their performance. At the end of sprints by both teams, we compared tasks of Table I with Table II and Table III and also compared estimations of the tasks with same complexity and we found changes for the estimations. Change in estimations are show in Table II and Table III. We again checked burn down charts and found that there was major positive change in burn down charts of both teams. Fig. 8 and Fig. 9 shows the new burn down charts for both teams respectively.

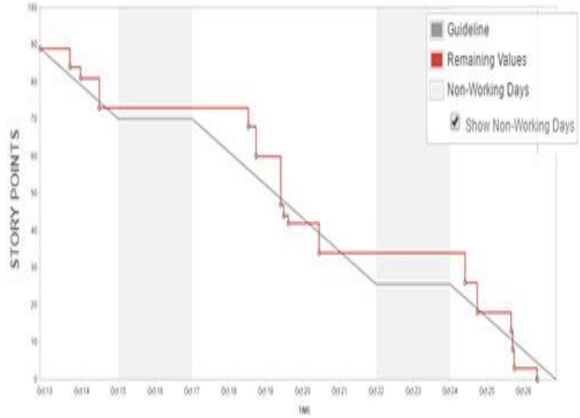

Fig. 8. Team A burn down chart

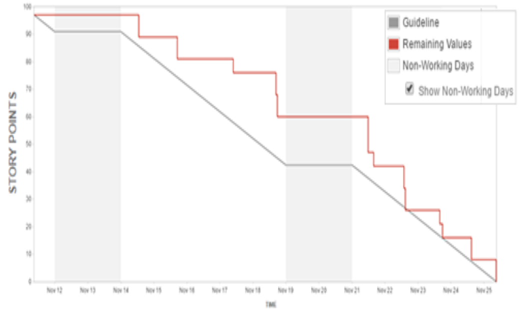

Fig. 9. Team B burn down chart

As we can see in Fig. 8 and Fig. 9, there are numerous changes observed in burn down charts as depicted in the mentioned figures, which shows the team performance throughout the sprint. Teams' progress line was close to ideal line. At the end of the sprint, both teams were relaxed and completed their tasks easily. Tasks moved accordingly and burndown chart is also improved.

TABLE II. TEAM (A) TASK VS ESTIMATIONS

Team A		
Task No	Equivalent Task in Table 1.1	Estimation
1	8, 16	3
2	11, 15	5
3	8, 16	3
4	1, 2, 6, 10, 14	8
5	11, 15	5
6		2
7	3, 4, 5, 9, 12, 13	13
8	8, 16	3
9	1, 2, 6, 10, 14	8
10	1, 2, 6, 10, 14	8
11	11, 15	5
12	11, 15	5
13	11, 15	5
14	1, 2, 6, 10, 14	8
15	1, 2, 6, 10, 14	8

TABLE III. TEAM (B) TASK VS ESTIMATIONS

Team B		
Task No	Equivalent Task in Table 1.1	Estimation
1		5
2	2, 3, 5, 9	8
3	2, 3, 5, 9	8
4		5
5	2, 3, 5, 9	8
6	2, 3, 5, 9	8
7	1, 6, 7, 8, 10, 11	13
8	2, 3, 5, 9	8
9	2, 3, 5, 9	8
10		5
11		5
12	2, 3, 5, 9	8
13	2, 3, 5, 9	8

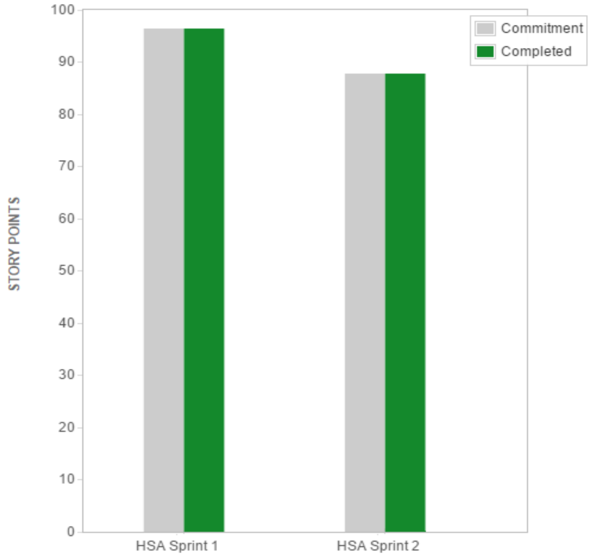

Fig. 10. Team A Velocity Chart

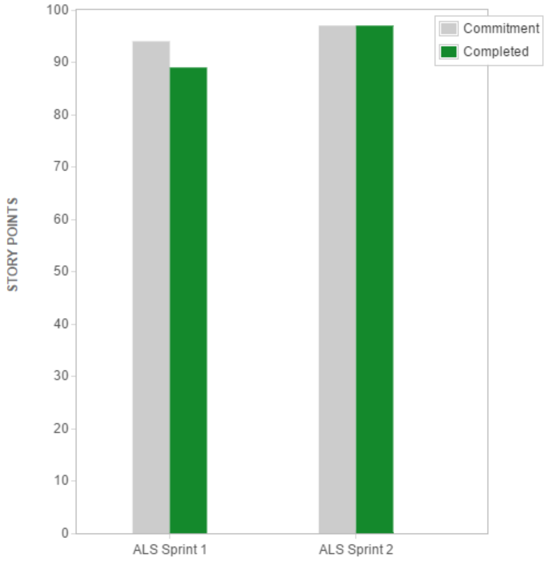

Fig. 11. Team B velocity chart

Fig. 10 and Fig. 11 are showing the velocity char for both sprints of both teams for both projects respectively. According to Table II and Table III, we can see story points has been increased and all team members will have equal opportunity to put effort on the tasks. We can concluded from these results that story points should be estimated according to developers, quality assurance engineer and product owner.

VII. Conclusion

Software systems have to ensure consistent and bug free execution at a rapid pace every time they are used especially in Agile Development. Improving software quality and performance has become a priority for almost every organization that relies on the software development. Thus the quality (estimation) issue related to the software's industry becomes more important, apparent and more technical also considering the user's requirements in this aspect.

The following work demonstrates an approach in the support of ASD especially in Scrum and comprehensively illustrate the different metrics to realize the team's work mistakes from different dimensions in the development industries. On the other side estimation of tasks, every team should also consider story points for QA engineer and Product Owner, as they also put effort on the tasks. This will be very helpful for better estimation. After every estimation, team should authenticate weather they have added points for all steps taken by the team to complete tasks. According to this paper every metrics has its own dimension that needs to be applied in the correct process. Many of Metrics calculations are strongly coupled with the story points to items in current version/Epic or sprint. Metrics play truly a key role in team development and help track the progress of the teams and any until product delivery. There are still a lot of metrics left that are related to scrum and estimations that needed to researched to introduce them into the current development scenario to boost up the agile team and process. Further, the following paper highlights the diagrammatic representation(s) of each metric with the help of Jira by Atlassian who are working in support of agile software development.

REFERENCES

[1] Nils C. Haugen, "An Empirical Study of Using Planning Poker for User Story Estimation,", in Proceedings of AGILE 2006 Conference (AGILE'06), 2006 IEEE.

[2] Rashmi Popli and Naresh Chauhan, "Cost and Effort Estimation in Agile Software Development", in 2014 International Conference on Reliability, Optimization and Information Technology ICROIT 2014, India, Feb 6-8 2014.

[3] Evita Coelho and Anirban Basu, "Effort Estimation in Agile Software Development using Story Points," in International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, US.

[4] Binish Tanveer, Liliana Guzmán and Ulf Martin Engel, "Understanding and improving effort estimation in agile software development- an industrial case study" in 2016 IEEE/ACM International Conference on Software and System Processes.

[5] Ziauddin, Shahid Kamal Tipu and Shahrukh Zia, "An Effort Estimation Model for Agile Software Development" in Advances in Computer Science and its Applications (ACSA) 314 Vol. 2, No. 1, 2012.

[6] Kjetil Moløkken-Østvold and Nils Christian Haugen, "Combining Estimates with Planning Poker – An Empirical Study," in Proceedings of the 2007 Australian Software Engineering Conference (ASWEC'07), 2007 IEEE.

[7] Rashmi Popli and Dr. Naresh Chauhan, "Research Challenges of Agile Estimation" in IJITKMI Volume 7 • Number 1 • December 2013

[8] Muhammad Usman, Emilia Mendes Francila Weidt and Ricardo Britto, "Effort Estimation in Agile Software Development: A Systematic Literature Review" in PROMISE '14, September 17 2014, Torino, Italy, ACM 978-1-4503-2898-2/14/09

[9] Torgeir Dingsøyr and Nils Brede Moe, "Research Challenges in Large-Scale Agile Software Development" in ACM SIGSOFT Software Engineering Notes, September 2013 Volume 38 Number 5

[10] Aneesa Rida Asghar, Shahid Nazir Bhatti, Atika Tabassum, "Role of Requirements Elicitation & Prioritization to Optimize Quality in Scrum Agile Development" International Journal of Advanced Computer Science and Applications(IJACSA), 7(12), 2016.

[11] Ingo Richter, Florian Raith and Michael Weber, "Problems in Agile Global Software Engineering Projects especially within Traditionally Organised Corporations" in C3S2E '16, July 20 - 22, 2016, Porto, Portugal, 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.

[12] Dan Radigan, "Five Agile Metric you won't be hate", Atlassian, 2015.

[13] Janeth López-Martínez, Reyes Juárez-Ramírez, Carlos Huertas, Samantha Jiménez and Cesar Guerra-García, "Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review" in 2016 4th International Conference in Software Engineering Research and Innovation, 978-1-5090-1074-5/16 $31.00 © 2016 IEEE

[14] Dan Turk, Robert France and Bernhard Rumpe, "Limitations of Agile Software Processes" in Third International Conference on Extreme Programming and Flexible, 2002

[15] Reiner R. Dumke, Andreas Schmietendorf, Martin Kunz, Konstantina Gorgieva,"Software Metrics for Agile Softwar Development Software Engineering Group", University of Magdeburg, Germany, 2008.

[16] Cockburn, "Agile Software Development," in Pearson Education, Asia Low Price Edition, 2007

[17] Aneesa Rida Asghar, Shahid Nazir Bhatti, Atika Tabassum, "The Impact of Analytical Assessment of Requirements Prioritization Models: An Empirical Study" International Journal of Advanced Computer Science and Applications(IJACSA), 8(2), 2017.

[18] Gurtej Pal Singh, "Scrum ceremonies and Artifacts", Agile and related methodologies, 2013.

[19] Robert Boyd, Practical Training, "9 Scrum metric to keep your team track", 21st Nov., 2014

[20] Zainab Sultan, Rabiya Abbas, Shahid Nazir Bhatti and S. Asim Ali Shah, "Analytical Review on Test Cases Prioritization Techniques: An Empirical Study" International Journal of Advanced Computer Science and Applications(IJACSA), 8(2), 2017.

[21] Quality metrics, "A guide to measuring software quality", SearchSoftware Quality, 2016.

[22] Scott Downey, "Scrum Metrics for Hyper productive Teams: How They Fly like Fighter Aircraft", 2013

[23] Shahid Nazir Bhatti, Deducing the complexity to quality of a system using UML. ACM SIGSOFT Software Engineering Notes 34(3): 1-7 (2009)

[24] CTS,"Essential QA Metrics for Determining Solution Quality", 2015

[25] Zephyr, "A Metrics the Value of Testing Metrics within Software Development", 2014

[26] Martin Kunz, Reiner R. Dumke and Niko Zenker, "Software Metrics for Agile Software Development", 2008

[27] K.V. Jeeva Padmini, H. M. N. Dilum Bandara, and Indika Perera, "Use of Software Metrics in Agile Software Development Process", 2015

[28] M. Kunz, R. R. Dumke, and N. Zenker, "Software metrics for Agile software development," in. Proc. 19th IEEE Australian Conference, 2008, pp. 673-678.