

Modified Architecture and Innovative Algorithm for Story Prioritization in Scrum Framework to Aid Product Owner

Darshita Kalyani^{1*}, Devarshi Mehta², Priti Sajja³

^{1,2}GLS University, Ahmedabad, India

³Sardar Patel University, Vallabh Vidyanagar, India

DOI: <https://doi.org/10.26438/ijcse/v8i2.8186> | Available online at: www.ijcseonline.org

Accepted: 07/Feb/2020, Published: 28/Feb/2020

Abstract- Agile has emerged as the most popular approach to project management because of its iterative and flexible nature. It is described by a prominence on responding to changes. There are various prominent agile frameworks used by the industries and scrum is one of it. This paper focuses on the detailed overview of agile scrum framework. It elaborates the challenges / limitations in scrum implementation. The point of convergence of the paper is proposing the algorithm-based remediation for the story prioritization process of scrum framework. The proposed framework also confronting the problem of scope creep faced by scrum squad. The targeted audience is professionals who are involved in scrum project management methodology.

Keywords- Agile, Scrum, Scope-Creep, Stories, Product Owner, Backlog

I. INTRODUCTION

Agile project management approach is customer centric. This approach allows much more flexibility towards the change in requirements from stakeholder than traditional method. It has some qualities like iterative development and continuous feedback from stakeholders. The system is developed in rapid cycles. Adapting the changing need of the stakeholder is the key aspect of agile. These qualities make the agile model more popular in the industry where changes occur frequently. The goal is for both groups to participate in a series of activities that will foster shared understanding and provide an underpinning of effectiveness [1]. This methodology allows to accommodate new features in the product as per the feedback of the stakeholder, which makes them related. This model focuses frequent review by stakeholder which aids in accelerated delivery of high-quality software This aids the organization to stay in competing environment.

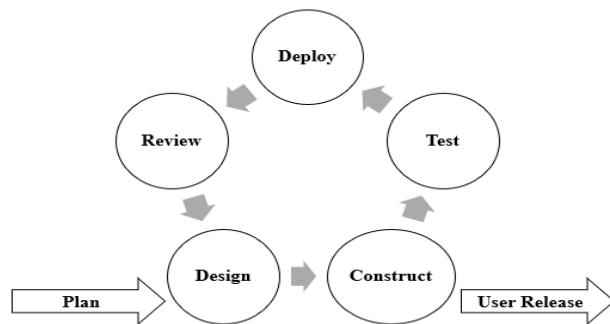


Figure 1: The Agile Concept

As shown in figure 1, any agile development method involves unceasing planning, design, development, testing

and deployment of the deliverables at the end of the iteration. This development process is done through various releasable iterations. At the end of each iteration the releasable feature is reviewed by the stakeholder. The agile project management methodology is implemented through various frameworks. The below section focuses on the detailed view of Scrum framework of agile.

II. SCRUM FRAMEWORK

Scrum framework works on the principle of observational inspect and adapt continuous feedback to deal with the changes in the product under development. Scrum framework is set of roles and responsibilities of the squad members, artifacts and meetings. Scrum framework conjectures that system progresses via series of iterations / versions known as sprints [3]. The scrum focuses on the squad members to take the product / system to a potentially shippable build when a particular sprint end. Actual product is reviewed at the end of each sprint which leads to detect bugs in the product at early stage and remove it in next sprint.

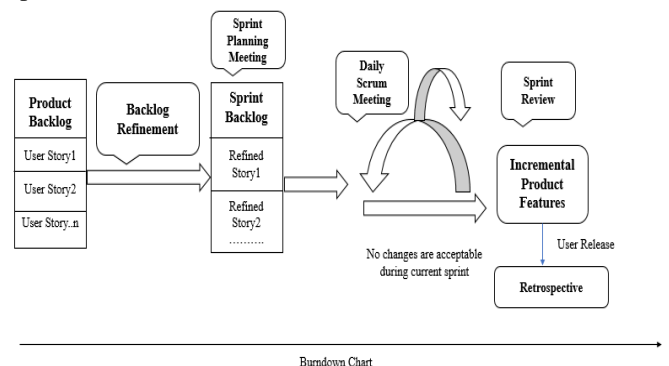


Figure 2 : Scrum Framework

The above figure depicts the concept of the scrum framework. In scrum, features are written from the perspective of the end-user. So, the features are known as user stories. Each story is implemented through the concept of sprint. Sprint is a fixed time line interval (typically between 1-4 weeks; with 2 weeks most common) during which team works on the stories. A meeting is conducted at the start of each sprint where the squad members commits on the stories / tasks they can implement during sprint [3].

Scrum works on the concept of epic, story and task. An epic is a logical grouping of work. Epic is considered as high-level description of what stakeholder wants. It is defined during initial product roadmap development stage. It creates the hierarchy of the product. It helps the team to break down their work. Work is broken down into shippable pieces. Each epic is divided into various user stories which includes several features / functions to be implemented. It is a short requirements / requests written with the context of an end user. And each story is further divided among various tasks which includes sub- functions. For example, if the user authentication is considered as an epic, then then several stories can be written for this epic like user login screen, forgot password workflow, lock accounts after certain attempts etc. Various tasks / subtasks can be created for each of these stories. Certain tasks of the story - User Login Screen – are design of the page, create required tables, design icons and images etc.

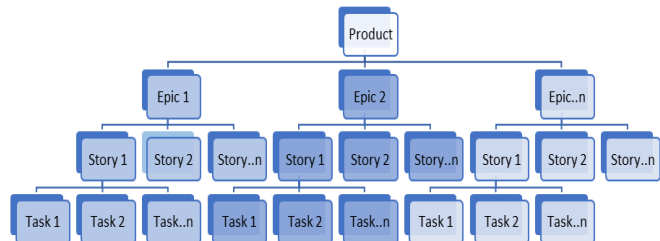


Figure 3: Epic, Story and Task Concept in Scrum

2.1 Scrum Steps

The scrum artifacts conveys the key information regarding flow of the user stories to the scrum team members. It guides the team regarding the activities to be implemented for development of user stories. The entire scrum framework is divided among four major steps which is depicted in the below figure.

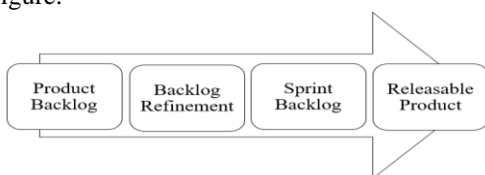


Figure 4: Scrum Steps

2.1.1 Product Backlog

Product Backlog contains the user stories created by product owner (User stories are a few sentences in simple language

that outline the desired outcome. They don't go into detail.) Any feature, any defect from the previous sprint is made available in product backlog. It is dynamic list which keeps on evolving as new requirements are incorporated. Stories in the product backlog are refined in backlog refinement stage.

2.1.2 Backlog Refinement

At the backlog refinement process, stories are refined i.e. stories are detailed in this process. E.g. in product backlog, the story may be created like: ‘Administrate user’ and in backlog refinement process, it is detailed like: ‘Register new user’, ‘Edit existing user’, ‘Find user’, ‘Delete user’. Stories are ranked as per the priority by product owner and scrum team in this process. The stories become actionable after backlog refinement process.

2.1.3 Sprint Backlog

Sprint Backlog is a container of work for delivering incremental product features to the stakeholders. The features that are in sprint backlog are coded and tested in particular sprint. Sprint Backlog contains the list of the refined stories with highest priority pulled from backlog refinement process and will be implemented in current sprint.

2.1.4 Releasable Product

The output of each sprint is considered as potential releasable product. It is the sum of all backlog refinement items which are pulled in sprint backlog and executed in current sprint. At the end of each sprint, a feature is released to the user (User Release). It must be in usable condition. Stakeholders review the deliverable and gives the feedback to the team. Any change suggested by the stakeholder is transformed into user stories and feed into product backlog again.

2.2 Scrum Squad

Scrum squad works towards achieving goals of the product under development. Team members works on different profiles towards releasing the features of the product during sprint. As depicted in below figure, the scrum squad consist of Product Owner, Scrum master and Cross-Functional Team.

Product Owner	Scrum Master	Cross-Functional Team
<ul style="list-style-type: none"> Has overall vision of the product, manages product backlog Creates epics, stories of the product and manage the changes by transforming it into stories 	<ul style="list-style-type: none"> Removes any impediments that obstruct a team's pursuit of its sprint goals 	<ul style="list-style-type: none"> Implements the stories of the product Performs analysis, design, coding and testing activities in sprint

Figure 5: Scrum Squad

2.2.1 Product Owner

Product owner receives input from the stakeholders. He has the global vision of the product and helps in setting the direction of the product. Product owner makes sure that right feature makes it into the product backlog. He creates epics and stories of the epics. Product Owner feeds the stories into product backlog. He is responsible for story sign off. The story is not considered as complete unless product owner signs it. Product Owners refined the stories in backlog refinement process along with scrum team. Product Owner sets the priority of the stories in backlog refinement process with scrum team [8].

When any major change requested from the customer, product owner needs to decide that to which epic the change belongs to. He also takes decision to create an entire new epic, if required, to accommodate big change from the stakeholders. Product owner creates stories for the requested change.

2.2.2 Scrum Master

Scrum master is responsible for smooth working of the sprint. He is responsible for ensuring the scrum framework is followed by squad members. He does this by ensuring that the scrum squad follows the practices / principles of the scrum framework. Scrum Master interacts with product owner to understand the vision for the product being built and he makes sure that the product domain is understood by each member of squad. He makes sure that every member of the team has tools that need to get their job done. He does not allow any change to get accommodate in the mid of the sprint. The Scrum Master removes any impediments that obstruct a team's pursuit of its sprint goals. He tries to ensure healthy relationship between product owner and scrum squad. He is responsible for solving any conflicts that arises among squad members. Scrum Master also tracks the progress of each sprint. The Scrum Master conducts daily stand-ups to get updates on the progress of the sprint [8].

2.2.3 The Cross-Functional Team

Cross-Functional team has different domain expertise. Cross-functional means that the squad has all the required traits to turn backlog refinement stories into a done increment. The stories are pulled into sprint backlog by scrum team according to priority set at backlog refinement process. The scrum team works on implementation of the story i.e. scrum team performs analysis, development (front-end and back-end developer), design, QA (quality analysis) activity, UAT (User Acceptance Testing) etc. The scrum team along with the product owner sets the priority of stories. When any change from the stakeholder is requested, the product owner along with the team sets the priority of the stories created for that change. The scrum team determines that how many stories can be pulled to sprint backlog for the current sprint [9,10].

2.3 Scrum Flow

Stories / Change in terms of stories are feed into product backlog. From product backlog, the stories are pulled into backlog refinement and stories are refined by product owner and scrum team. In this process, product owner and scrum team set the priority of the stories. The stories are ranked according to its complexity. Most often Fibonacci series - 1,1,2,3,5,8,11...n - is used to assign complexity point. Higher the value in the series, higher the complexity of the story. Highest priority stories are implemented in sprint. At the end of sprint, potential shippable features are released.

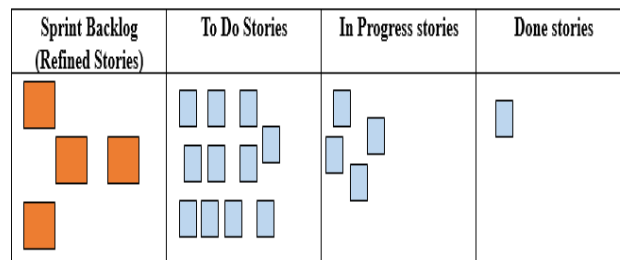


Figure 6 : Sprint Roadmap

As shown in figure 7, the tasks that are to be implemented in sprint are placed in left-most column. The tasks move from left to right on dashboard upon their completion. The sprint is considered as successful if all the tasks are in 'Done' column.

The team holds the daily stand-up meeting to discuss the current and future progress of sprint. Retrospective is conducted at the end of each sprint to discuss the experience during previous sprint.

Scrum team uses the visual tool called burndown chart to view the information of progress of the sprint. The burndown is generated squad wise. It is used to judge how did a squad do it in the previous sprints – it is past looking approach. velocity is one important element of burndown chart. It is used to visualize whether the team is in the right direction with required speed to complete the target [10].

III. SCRUM LIMITATIONS

In scrum, sequence of events to achieve the product is not fixed. This make product expand to several sprints. Scrum also faces the problem of requirements change or the understanding of requirements change. Due to this, the scrum squad often faces the problem of scope creep. The problem of scope creep arises due to the change in requirements of stakeholders. In scrum, the situation of the scope creep arises when the stakeholders want to swap new work for work already completed by scrum squad in sprint.

After study of existing framework, three possibilities of requirement change have been identified which has been shown through the following figure.

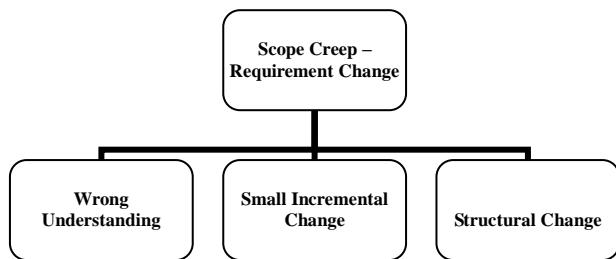


Figure 7: Scope Creep issue in Scrum Framework

Wrong Understanding:

In this issue, the requirement given by stakeholder is right but the product owner has not understood it in correct way. So, when the demo is presented before stakeholder, the change in the incremental feature is demanded from stakeholder.

Small Incremental Change:

In this issue, the requirement is not mentioned by the stakeholder. Product owner needs to deal with this kind of changes. For example. The stakeholder may demand to put an extra button on page, may demand to create an extra page etc. This change might not lead to throwaway of existing / past work.

Structural change:

This can be considered as major change in releasable product. It may also lead to throwaway of existing / past work. Product owner need to decide about the epic to which the change will belong. The change might affect existing epic, multiple epics or the product owner needs to create entire new epic to confront with big structural change. Product owner creates stories for the change i.e. function

wise breakdown is done for the change. Product owner needs to revised the entire roadmap of product.

When any structural change is requested from stakeholder, product owner needs to figure out which existing stories are affected. He needs to create various stories for change and reprioritized the stories contained in the product backlog. This paper focuses on the algorithm which calculates rank for stories to be implemented in sprint. The structural change is divided among multiple stories. Each story is evaluated on the basis of certain parameters and then highest ranked story is implemented first.

Please Assign Weights [Between 0 to 1] to each Parameter for Each Kind of Change in Scrum.

IV. MODIFIED SCRUM FRAMEWORK

As per the literature survey and discussion with the professionals working in industry, currently the priorities of the stories are assigned by product owner and team members manually. The following sections focuses on modified scrum framework with an additional component- algorithm for story prioritization. The score of the story is calculated through algorithm using statistical formula which has been covered in following section.

4.1 Proposed Architecture to Modify Scrum Framework

The following diagram shows the modified scrum framework with the introduction of additional components – ‘Innovative Story Prioritization Algorithm’. The Additional component is highlighted with orange colour.

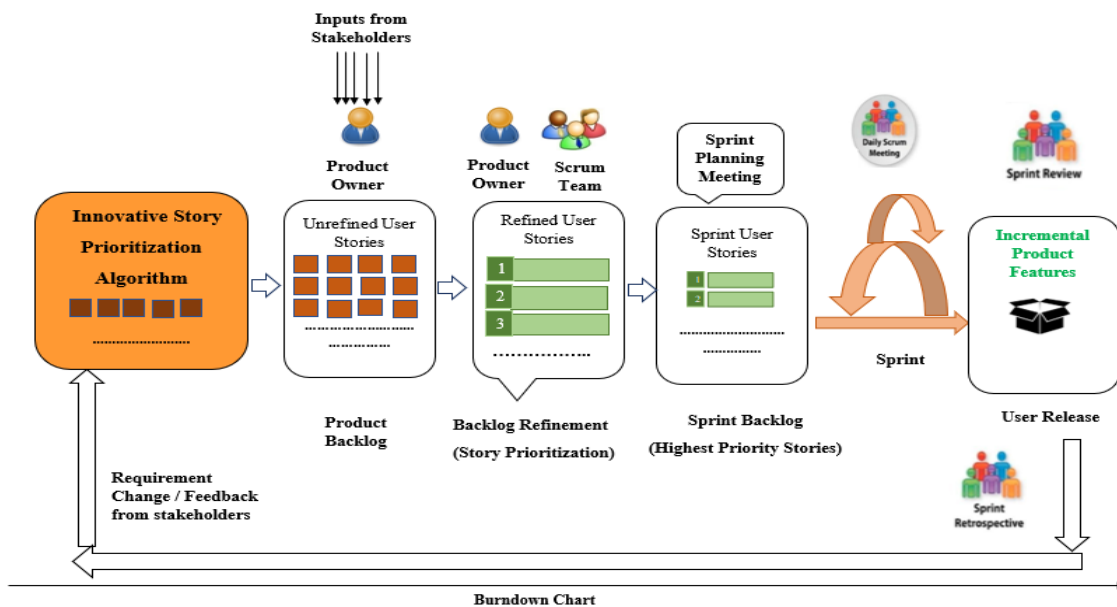


Figure 8 : Modified Scrum Framework

As per figure 8, modified scrum framework works in the following manner:

When any change from the stakeholder is requested, product owner creates stories for that change and put it under existing epic / epics or create an entirely new epic in the case of major structural change. The proposed algorithm calculates the score of each story through statistical formula. Certain parameters (as mentioned in below algorithm) are used to compute score of each story. The different parameters used in the calculation of the score of the stories are decided after consulting experts from industry whom have sound working experience in scrum. These parameters are assigned weights as suggested by the industry experts. Each story is assigned complexity points. Each member of the team assigns complexity point to each story in the range of 1-10. Then the average is computed for each story complexity point. Using weights assigned to parameters and complexity points of each story, score for each story is calculated. Highest priority stories are pulled in sprint backlog and executed through sprints.

The below mentioned algorithm computes the score of each story created for change requested from stakeholders. The algorithm calculates the score based on weights assigned to different parameters and story complexity points (sets by team member). Highest score stories are pulled in backlog that would be executed in sprint.

4.2 Proposed Algorithm for Calculating Score of the Story for Setting Priority

Input: Stories for the change, Weights to parameters, Story complexity points

Output: Score of each story

Step 1: Start

Step 2: Assign weights to each parameter [Value of weight \rightarrow [0,1]]

Operational_Feasibility \leftarrow Weight_Operational_Feasibility

Technical_Feasibility \leftarrow Weight_Technical_Feasibility

Economical_Feasibility \leftarrow Weight_Economical_Feasibility

Time_Feasibility \leftarrow Weight_Time_Feasibility

Legal_Feasibility \leftarrow Weight_Legal_Feasibility

Usability \leftarrow Weight_Usability

Step 3: For each change in Product n requested from stakeholder

Step 4: Create n stories for change in product

Step 5: Calculate complexity point for each story [Value of complexity point [1,10]]

Step 6: Complexity point assignment to each story by

every team member of team

Step 7: Calculate average of all complexity point assigned to story by team member

Step 8: (

Step 9: (Value of complexity point by team member 1 + Value of complexity point by team member 2 + + Value of complexity point by team member n) / Total_no_of_Team_members

Step 10:)

Step 11: Assign complexity point to each story obtained as result of calculation of average function

Step 12: Calculate Complexity Points _Parameters Weights matrix

Step 13: (

Step 14: Aggregate final weights for each story

Step 15:)

Step 16: Sort the stories in descending order

Step 17: Pull the highest priority stories according to score in sprint backlog

Step 18: End

This will aid the product owner to calculate score of each story using statistical formula instead of assigned it manually based on merely a wisdom. So, it would add a value-addition into setting priority of the stories and accordingly pull the items in sprint backlog.

V. CONCLUSION

As per the discussion, the priority to the stories are set by product owner and team manually in the scrum. This paper focuses on the algorithm which uses statistical formula for computing score of each story for the change requested by the stakeholders. According to the score calculated through algorithm, the stories are pulled into sprint for implementation. This priority is set through systematic calculation rather than the mere wisdom of the product owner and team. This proposed algorithm uses parameters through which the change is evaluated for its importance on different grounds and ranked accordingly for the implantation. This may help the product owner to revised the roadmap by resetting the stories in backlog refinement process.

REFERENCES

- [1] The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game.
- [2] Eliza S. F. Cardozo , J. Benito F. Araújo Neto , Alexandre Barza , A. César C. França , Fabio Q. B. da Silva, SCRUM and Productivity in Software Projects: A Systematic Literature Review, **2010**.
- [3] Daniel Haslinger & Dr. Andreas Wintersteiger, White Paper: Implementing Scrum & Agile Development, **2017**.
- [4] Julian M. Bass, Scrum Master Activities: Process Tailoring in Large Enterprise Projects IEEE, **2014**.
- [5] Dan Rawsthorne & Douglas Shimp , White Paper: Scrum 3.0 – 3BACK, LLC, **2017**.
- [6] Santosh Kumar Yadav, Naveen Kumar Jangid, S Rajeshwari, A Survey on Agile Software Development Methodologies, IJSRCSEIT, **2017**.
- [7] K M. Jyoti, Mr. Tinku Singh, Parul Saharavat, Analysis the Strength of Agile Methodologies in Software Development, IJSRCSEIT, **2018**.
- [8] Eliza S. F. Cardozo, J. Benito F. Araújo Neto, Alexandre Barza, A. César C. França, Fabio Q. B. da Silva, SCRUM and Productivity in Software Projects: A Systematic Literature Review
- [9] Kevin Vlaanderen, Slinger Jansen, Sjaak Brinkkemper, The agile requirements refinery: Applying SCRUM principles to software product management -
- [10] Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review – IEEE, **2016**.
- [11] Shruti Sharma , Nitasha Hasteer, A comprehensive study on state of Scrum development , **2016**.
- [12] DEEPA VIJAY, GOPINATH GANAPATHY, Empirical Case Study of Agile Scrum Process, **2013**.
- [13] Astha Singhal, Divya Gupta, Scrum: An Agile Method - International Journal of Engineering Technology, Management and Applied Sciences, **2014**.
- [14] Youry Khmelevsky, Xitong Li, Stuart Madnick, Software development using agile and scrum in distributed teams-IEEE, **2017**
- [15] Monika Agarwal, Prof. Rana Majumdar- Tracking Scrum projects Tools, Metrics and Myths About Agile, International Journal of Emerging Technology and Advanced Engineering, **2012**.
- [16] Nagy Ramadan Darwish, Salwa Megahed, Requirements Engineering in Scrum Framework, **2016**.