Journal of Theoretical and Applied Information Technology
31st March 2019. Vol.97. No 6
© 2005 – ongoing JATIT & LLS

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

DEVELOPMENT AND ANALYSIS OF QUALITY COMPREHENSIVE MODEL WITH SCRUM TREE ALGORITHM FOR AUTOMATIC USER STORY PROCESSING

[1]MRS. RUPALI A. MAHAJAN, [2]DR. S. K. YADAV, [3] DR. SURENDRA A. MAHAJAN

[1]PhD Research Scholar, Department of Computer Science and Engineering JJT University, Rajasthan, India
[2]Director -Research, Research –Guide, JJT University, Rajasthan, India

[3]Assistant Professor, Department Of Information Technology, PVG's College of Engineering and Technology, Pune, India

Email: [1]rupalimahajan@gmail.com, [2]drskyadav@hotmail.com, [3]sa_mahajan@yahoo.com

ABSTRACT

Scrum framework is needed to keep up openness of a project life cycle. If structure ceases to conserve the transparency subsequently it is vital to hunt down the reason behind same. This paper provides the new algorithm that helps you to examine the openness within tasks as per scrum user stories. The core assumption is that, user stories are back bone of work and this can be formulated at preliminary stage of project cycle and the actual performance of stories takes a long time to accomplish. If any of these tasks are time intensive, it is crucial to formulate sub-stories for major task and conduct whole Scrum cycle for same. To address this problem, proposed research focusing on decision goal modeling with scrum tree algorithm where project manager can get automatic goal scheduling effectively as per scrum stories. The evaluation of proposed system is done for real time project and also tested for Thurman dataset. Performance parameters like data centrality, team closeness centrality, influential team nodes, and decision proximity along with sprint delivery decision are evaluated and presented as a map. As per performance evaluation results, the proposed work is best suited for any existing agile scrum project.

Keywords: *Scrum, Agile, Kanban, Sprint, Sprint Retrospective*

1. INTRODUCTION

Scrum is amongst the most favored frameworks for employing agile. Numerous frameworks may be used to apply agile, like kanban as an example, however scrum has a distinctive quality as a result of responsibility to brief

Task. Small iterations furthermore boost the significance of beneficial estimation along with rapid feedback from assessments each repeating problems in waterfall projects.

Whenever a Scrum team have to figure out whether or not they can apply a user story

iterations of work. Having scrum, the product is made in several fixed-length iterations known as sprints that provides teams a structure for delivery of software with a standard cadence. Milestones i.e. the conclusion of a sprint–come usually, providing together a sense of perceptible improvement using every cycle which works on and stimulates every

immediately or needed a raise 1st, they will assess the degree of uncertainty. If that degree is over a specific tolerance, let’s call that the particular spike tolerance they’ve subliminally set to be an undesirable threat to consider in to a sprint, and then they choose a raise first. Even though we declare it really is challenging to gauge, let evaluate among teams/products, we

believe is this raise tolerance was pretty steady along the teams we had worked with. It turned out in ways that teams had to begin with a raise on around 10% of the user stories and this worked well.
Within this specific project, while using identical tolerance might have resulted in greater than 50% individuals sprint backlogs because spikes, which in turn we didn't wish mainly because that may delay our chance to produce benefit to our clients. Consequently, we consented to accept much more uncertainty within our sprints by improving our spike tolerance in order to maintain the volume of spikes within our sprints reasonably low and acquire an opportunity to deliver value more quickly.

Scrum over Agile

Lei, Howard, et al. (2017), Agile, Kanban as well as Scrum are phrases utilized in project administration. The Agile strategy engages incremental as well as iterative function defeats which are also known as sprints. Scrum, alternatively is the kind of agile technique that is certainly employed in software development. Kanban and also Scrum is a couple of highly effective Agile project management solutions in application development. The goal of Scrum and Kanban is accomplished by optimizing the development procedure by discovering the responsibilities, controlling time period better, along with setting-up groups.
Scrum is definitely an iterative and also incremental agile application development way for handling application projects as well as product or software development. Therefore Scrum is definitely a sort of Agile technique and that is employed broadly in application developments. Therefore, Scrum is really a particular taste of Agile, exclusively it is known as an agile project management platform. Within the SCRUM strategy, a Sprint will be the fundamental model of development. Every Sprint begins with an organizing meeting, in which the responsibilities for the sprint are usually determined and also approximately determination for that sprint objective is done. A Sprint concludes using an assessment or retrospective meeting in which the development is analyzed and lessons for that subsequent sprint are determined. Throughout each and every Sprint, the group produces completed parts of a Product.
Within the Agile approaches each and every iteration entails a team functioning by way of a entire application development cycle, which include organizing, specifications analysis, design and style, coding, unit testing, along with acceptance testing whenever a functioning product is exhibited to stakeholders. Consequently, if within a SCRUM Sprint an individual accomplish each of the application development stages (from requirement evaluation to acceptance testing), and we can easily claim SCRUM Sprints match AGILE Iterations.

In summary, in scrum methodology, initiatives are partitioned directly into sprints, that normally takes 1, 2 or 3 weeks. By the end of each one sprint, many stakeholders meet to measure the improvement and prepare its subsequent actions. Another benefit from scrum is that the project's route being modified depending on accomplished work, certainly not on supposition or forecasts. We consider there ought to be more intriguing truth of applying Scrum over Agile in application development as well as project supervision. Scrum is growing to be well-known because of its convenience, established productiveness and capability to accomplish as a wrapper for numerous design techniques endorsed by various agile techniques.

Scrum over Kanban

As per Padmanabhan et. al (2018), important aspects that influence the approach implementation are part of stakeholder, workforce administration, project framework and determination capability. Almost all of the noted resources and approaches primarily give attention to the approach formulation.
The 1st step with the agile transformation is determining exactly what agile structure is best suited in your case whilst your workforce. Agile is a collection of beliefs and concepts which provide as the North Star. Kanban as well as scrum tend to be frameworks. They're constructed around the agile principles and explain the method that you whilst your workforce team up, discover, and acquire things done.

As per research by Taibi, Davide, et al. (2017), whenever determining among scrum and kanban, realize that you're in right direction. Kanban and scrum are 2 of the extremely well-known agile frameworks and also have demonstrated their particular value at many organizations for almost 2 decades. Kanban and scrum aren't really the only alternatives. Fig. 1 shows the Kanban analysis tool cumulative flow diagram.

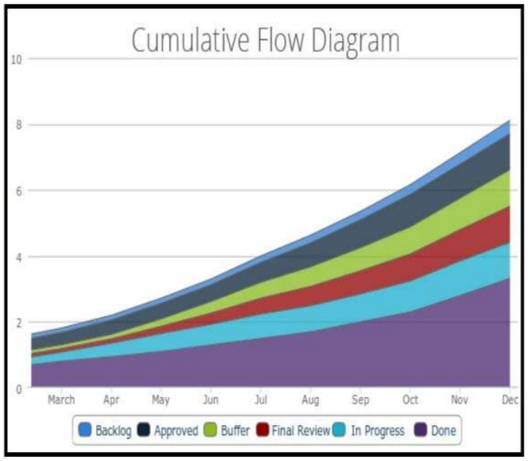

Figure1.Kanban Cumulative Flow Diagram Tool Output (Source: Kanbantool.com)

Kanban is dependent on continual improvement and shipping, dealing with a small amount of responsibilities fluidly and also simultaneously. Kanban groups utilize a graphic organizing tool the Kanban board which exhibits each and every task (user story) on a card in addition to moves cards by means of columns which symbolize intensifying levels of achievement. If the workforce carries a steady mode of work demands, Kanban can be befitting anyone.

2. LITERATURE REVIEW

As per Turetken et. al, (2017) even though agile application development techniques have acquired extensive endorsement in practice, the issues about the scalability as well as integration of agile routines in conventional large-scale program development assignments are relevant. Scaled Agile Framework (SAFe) has come forth as a strategy to deal with many of these problems. To deal with this particular need, author formulated a maturity design that delivers direction for application establishing organizations in determining a roadmap for implementing SAFe. The design could also be used to analyse the degree of SAFe usage.

As per Arthur et. al, (2017) although traditional Agile approaches are certainly not commonly appropriate to significant multi-year along with mission-critical programs, Agile hybrids are increasingly being designed (such as SAFe) to manipulate the efficiency enhancements of Agile whilst maintaining the mandatory practice rigor and coordination requirements of such assignments. Through the point of view of Self-sufficient Confirmation and Affirmation, even so, the adoption of such hybrid Agile frameworks is now challenging. This paper documents our analysis of
(a) appropriate literary works, (b) the customization as well as usage of Agile frameworks to allow the development of significant level, quest critical programs, and (c) the compatibility of typical IV&V approaches in hybrid Agile development frameworks.

Oliver et. al, (2018), introduced brief communication concerning Agile and Agile approaches like Scrum and Kanban, so you could utilize this structure within the operations of your respective subsequent application development task. At its easiest type, consider the structure in an effort to lessen inefficiency by means of quick prototyping, cross-functional workforce set up, along with evidently described non-overlapping assignments. The established predictions had been that by iterating about the same development routine a number of instances, a learning challenge commences to adopt consequence that generates effectiveness benefits.

As per study by Ghafoor et. al (2017), application Development corporations are implementing agile strategies in globally software development (GSD) along with local software development (LSD). Next to the comprehensive advantages that agile approaches present to GSD and LSD, additionally, it includes numerous essential concerns to each circumstance throughout the complete span of software development life cycle (SDLC). This paper provides the contribution in creating a systematic literature review (SLR) process for discovering the important problems in GSD and

LSD surroundings. The specified results of the SLR method might be a band of crucial risk factors/issues and their particular concerned procedures which will be a good choice for companies avoiding these kinds of difficulties while implementing agile guidelines in GSD and LSD. This paper also reveals the initial outcomes of the SLR method.

Annosi et. al (2018), aspires to discover Management Control Systems (MCS) caused by the enactment of agile progress strategies, depending on a recognized MCS taxonomy. An abdicative method was implemented, taking into consideration the lack of research analyzing the post-adoption outcomes of agile approaches. The paper shows exactly how Scrum, a prevalent agile technique, implicitly delivers numerous enforcing levers of management to a team’s self-regulatory learning procedures.

Mendez et. al (2018) depicted that defining the tasks of a Sprint and the assignment of tasks are activities that are done through team experience and based on the technique of Expert Judgment that are empirical. On the other hand, the mathematical optimization provides several tools to solve the problem of assigning tasks to resources; this is known task scheduling. This paper proposes to apply the exact techniques of scheduling to the task assignment problem in agile software development based on the Scrum model.

Study by Alhazmi et. al (2018), presents a Sprint Planning decision Support System (SPESS) which is a tool to assist the managers for Sprint planning. Among considering other Sprint planning factors, SPESS takes into consideration developer competency, developer seniority and task dependency. The results are that the assignments of the tasks of each Sprint to developers guarantee that each team member contributes to their fullest potential, and project planning is optimized for the shortest possible time.

As per Hron et. al (2018), Agile application development methods get accumulated prevalent endorsement and software throughout all market sectors. Scrum, among the most favored agile procedures, has become implemented in numerous companies. The particular changes and improvements which are created to match Scrum to real-world needs: whether it's to resolve methodological disadvantages, to match the strategy to particular contextual concern, as well as to include more importance towards the approach by enhancement or blend with different tools as well as techniques. To obtain an understanding of the recommended modifications and their particular benefits, this analyze offers a step-by-step overview of literature confirming on issues and reasons which bring about improvements with the Scrum procedure. According to 31 appropriate scientific studies we acquire 7 specific reasons for enhancing Scrum, along with 6 universal alternative ways of modify the technique.

Ahmed et. al (2018), Agile application development by means of Scrum is generally regarded as a strategy; nevertheless than demonstrating Scrum as strategy, contemplate of it as a structure for coping with an operation. This paper can set off while using the background, it's going to deal with the features and classification of agile application development and also features the significant different agile application strategies. Numerous agile approaches may also intricate in this paper. The key purpose of this paper is to determine risk on agile development and also enhance the standard of the application by making use of agile strategies.

Khmelevsky et. al (2017), Agile application development procedures, similar to Scrum, which enable groups to pay attention to delivering product or service and enhanced communication offers managed to get among the least complicated and finest application development methods. However, these kind of agile approaches are intended for collocated application development and are also therefore circuitously relevant to dispersed agile development. In this paper, author provided information through case studies as well as real world distributed Agile and Scrum projects carried out since 2011, in addition to the difficulties and gains the case assignments described and distinctive instruction discovered from them.

Dhir et. al (2019), In the agile software development, there are different factors behind the success and failure of projects. Paper represents the success, failure, and mitigation factors in agile development. A case study is presented depending on all of these factors after the completion of small projects. Each team grouped into 10 team members and developed the project with different approaches. Each group maintained the documentation from initial user stories and factors employed on the projects. Final outcomes are observed based on the analysis of efficiency,

accuracy, time management, risk analysis, and product quality of the project. Final outcomes are identified using the different approaches.

Yousaf et. al (2017), The key goal of the work is to suggest a new design for Scrum, by building consistency architectural techniques to accomplish a trusted application system. Furthermore, information gathered through the software sector, Scrum professionals have assisted us develop the suggested integration; it offers superior the entire validity with this integration, as the responses through industry experts has additionally been involved. The complete outcome pointed out soon after integration that, it also features a constructive effect on expense and time period of developing application. This analysis has opened up new opportunities to give the work within the agile area and dependability engineering techniques that may eventually flourish in developing high-quality, trustworthy applications.

Khalil et. al (2017), in this paper article author accumulated the information associated with the standard waterfall as well as new Agile method for accomplishment, inhibited and also unsuccessful rates through numerous information sites from 2010 to 2016 and determined the two of these techniques costs by means of percent to indicate exactly how Agile strategy is functioning excellent for establishing the application project which is the accomplishment, inhibited and unsuccessful rates of strategies are usually in agile.

Mishra et. al (2018), in “Agile” structure, the client performs in cooperation with all the venture workforce in prioritizing the requirements. The execution is performed by way of “Scrum” method, possessing numerous “sprints, ” and also each and every sprint features a “working software” as being a deliverable. This method has considerably lessened the “time” since the client can choose which popular features of the application they wish to be supplied on a top priority schedule. The release of sprints is just like multi-releases of a software package where application is screened carefully to discover the fundamental defects towards the end of each and every sprint and also leftover volume of fault of every sprint is considered onward for that subsequent sprint. Consequently, to design the problem prognosis trend and their particular pattern in every sprint, application stability progress modeling continues to be applied. In the present work, author employing application consistency progress types to determine the pattern within the sprints which could eventually determine the entire quality of the application. Mathematical model is described by the end of the paper for design affirmation.

3. PROPOSED RESEARCH METHODOLOGY

Scrum teams are the champs for lasting development routines. The most efficient scrum groups are co-located, and generally 5 to 15 associates. Associates have different type of expertise sets, and also cross-train one another therefore nobody turns into a bottleneck in the shipping of work. All associates from the workforce assist each other to guarantee an effective sprint achievement.

The key player

Scrum masters [18] include the champ for scrum into their workforce. They mentor the workforce, the product owner, as well as the organization about the scrum method to check out approaches to fine- tune their particular training of it. A powerful scrum master profoundly comprehends the effort getting accomplished through the workforce and may assist the workforce enhance their particular shipping flow. Because the facilitator-in-chief, they routine the essential assets for sprint setting up, sprint assessment, and also the sprint retrospective. Fig. 2 shows the standard agile project administration predicament.

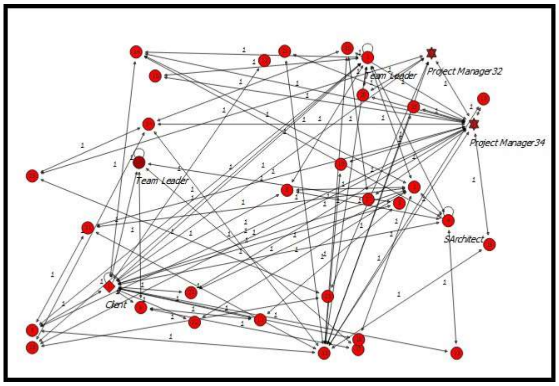

Figure2. Traditional Agile Software Project Management Scenario

fresh to scrum: modifying the sprint's extent right after it has previously initiated. Product owners can occasionally inquire, "Can't we have this one more super-important minor point into this sprint? " However maintaining extent firm, reinforces beneficial estimation and also product planning–not to note fends a cause of interruption to the development crew [19].

Scrum masters are generally wrongly diagnosed for task managers, whenever in reality, project managers do not genuinely have an area within the scrum system. A scrum workforce handles its very own success and self-organizes all-around their particular function. Agile groups employ yank designs in which the group draws a lot of work from the backlog and also commits to finishing it that sprint, which can be extremely successful in preserving good quality along with guaranteeing perfect effectiveness of the workforce within the long-term. Neither of them scrum masters nor project managers or product owners force function towards the workforce.

Key Processes

Productive application development is inhibited by the opportunity to handle complexness, technologies advancement, and prerequisites transform [20]. Consequently, sprint performs crucial role is usually complicated task delivery. A sprint (or iteration) is the significant unit of development in Scrum. The sprint is a occasion encased attempt; which is, limited to a certain length. The timeframe is predetermined upfront for each and every sprint and is commonly among 1 week and 30 days, having a couple weeks getting the most frequent.

Every sprint commences using a sprint Scrum masters furthermore turn to deal with road blocks and interruptions for the development workforce, insulating these individuals through external interferences whenever feasible.

Perhaps the scrum master's task is always to prevent an anti-pattern popular amongst groups planning part which seeks to establish a sprint backlog, discover the task for the sprint, and produce nearly prediction for the sprint aim. Every sprint finishes with a sprint assessment and sprint retrospective, which reviews development to demonstrate to stakeholders and determine lessons along with enhancements for that next sprints.

Scrum focuses on working product towards the end of the sprint which is definitely accomplished. When it comes to application, this most likely consists of how the application has become completely bundled, analyzed and recorded, and is likely shippable.

Throughout a sprint, graphic artifacts just like task boards as well as burn down charts [21], noticeable for the workforce and visitors as well, are highly effective motivators. They generate a nature of "we're achieving this!" Getting the probability to display new work at the sprint demonstration is similarly inspiring, and the reliable, incremental responses the workforce receives from stakeholders at each and every demonstration generates a strong approach to develop solutions. Fig. 3 shows the sample burn down chart.

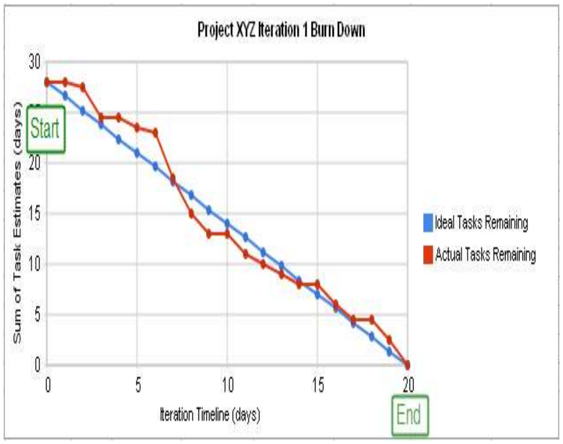

Figure3.Agile Scrum Burn Down Chart (Source: Agilenutshell.Com)

A time-boxed interval accustomed to investigate a perception or develop a straightforward prototype. Spikes can possibly be organized to occur between sprints or, for greater groups, a spike could possibly be acknowledged as among the list of sprint delivery goals. Spikes in many cases are launched prior to shipping of huge or complicated product backlog objects so as to safeguarded budget, broaden understanding, or make a evidence of strategy. The timeframe along with objective(s) of any spike is decided among product owner as well as development workforce prior to the commence. Contrary to sprint responsibilities, spikes may perhaps or may not provide perceptible, valuable features. As an example, the goal of a spike could possibly be to effectively accomplish a determination on a plan of action. The spike ends in the event the time period is up, definitely not when the goal has been delivered.

The parameter "Velocity" is the volume of story details accomplished in a sprint will be the core metric intended for scrum groups. It guides foreseeable future sprint responsibilities, or just how much work the scrum workforce commits to. In the event the workforce finishes typically 20 story points(Velocity = 20), it won't accept to a sprint backlog which contains 30 points. Groups attempt to not create extent modifications throughout a sprint. Scrum groups occasionally get opinions and discover that exactly what they're focusing on isn't as beneficial to the client as they considered. In such instances, the extent of the sprint ought to transform to indicate the significance of delivery benefit to the client to start with. Throughout the sprint retrospective, scrum groups must examine the way to control change in foreseeable future, as improvements place the potentially shippable increment at an increased risk. Project life cycle period is an essential metric intended for Kanban groups [22]. It is the common time frame which it normally takes for any activity to relocate from the beginning to end line. Enhancing cycle times implies the achievements of Kanban squads.

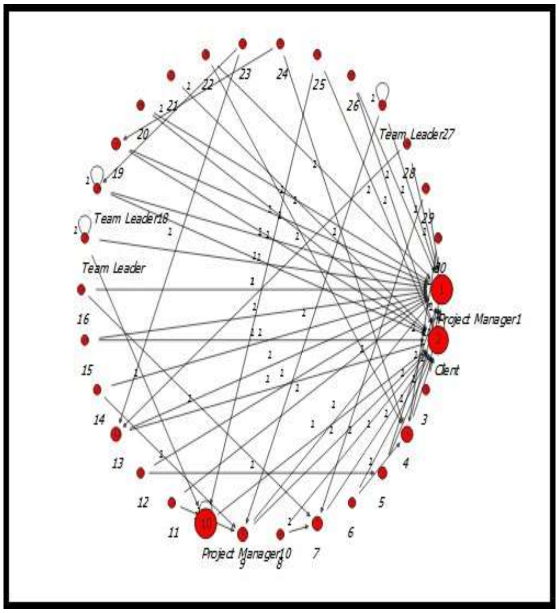

Figure 4. Kanban Project Management Scenario

Commonly, Kanban project management framework for stakeholders is as revealed in Fig. 4 above. Cumulative Flow Diagram (CFD) [23] is analytical software employed by Kanban squads to comprehend the volume of work products in each condition. CFD assists to determine particular bottlenecks which have to be reconciled for superior throughput.

Kanban is dependent on a consistent workflow framework which continues squads nimble and able to conform to altering priorities. Function products manifested by Kanban cards move from one phase of the workflow to another till there're designated as "Done". Popular workflow phases are "To Do", "In Progress". These kind of variables are compared in section 4 with proposed work. Additionally, Scrum functions much less properly within the subsequent conditions:

Teams in whose associates are geographically distributed or part-time: In Scrum, developers must have close proximity along with continuing conversation, essentially functioning collectively within the similar space quite often. The latest enhancements in technological innovation have lessen with the influence of such hindrances.

Teams in whose associates include very particular expertise: In Scrum, developers are able to focus on almost any activity or acquire work which a different developer has commenced. This is often handled by great Scrum leadership. Even though associates with extremely particular expertise can and do the job well, they must be motivated to acquire more information about and team up with different disciplines.

Products having numerous outside dependencies: In Scrum, splitting product development straight into brief sprints needs cautious setting up; outside dependencies, like sheduled delivery of application from some other groups, can cause delays along with the failure of specific sprints.
Products which are develop fully or older or with managed quality control: In Scrum, product increments need to be entirely designed and analyzed within a sprint; products that want considerable amounts of regression testing or security assessment for each and every release are much less suitable for brief sprints as compared to extended waterfall releases.

Proposed Methodology

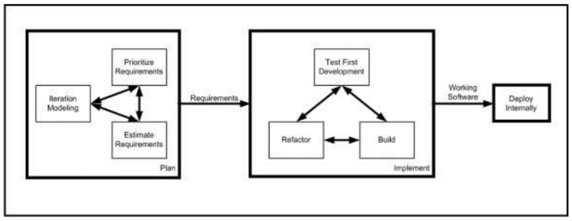

Figure 5: Existing Agile Model Driven Development [24]

According to literature evaluation, research gaps are mentioned as follows:
In recent times it is noticed a standard set of issues that organizations apparently experience. These kinds of typical difficulties are:

1. Presently there isn't a venture architecture effort.
2. Skewed focus.
3. Project squads don't understand the enterprise architecture is available.
4. Project squads don't adhere to the enterprise architecture.
5. Project squads don't assist the enterprise architects.
6. Obsolete architecture.
7. Directly targeted architecture models.
8. Dysfunctional "charge back" strategies.
9. A "do all of this additional work mainly because it's beneficial to the company" mind-set.

A typical thread driving these issues is a focus on techniques and tools around individuals and communications. In case the organization runs into many of these kinds of difficulties subsequently you might need to consider

To investigate the impact of utilizing numerous project management techniques in a group project, an experiment evaluating 3 distinctive approaches will be accomplished. Since we could analyze numerous groups carrying out work on a single task, a research considered a suitable methodology to evaluate and analyze the consequence of the assorted project administration tactics, and thus, an investigation is certainly an appropriate procedure for check out investigation issues. Proposed system is a structure for effective project performance. In this framework, Scrum Framework will be designed for decrease in SDLC delays and also concern problems. Refer Fig. 5 for existing agile model driven development architecture flow.

possessing an agile scrum procedure for organization structures with goal driven decision framework.

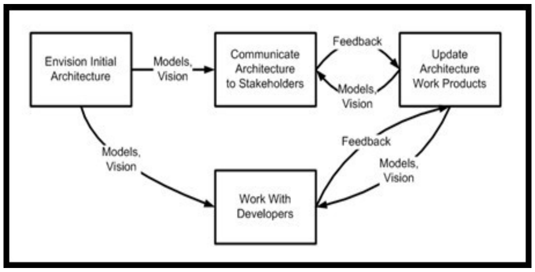
Envision Initial Architecture
Models, Vision
Communicate Architecture to Stakeholders
Feedback
Models, Vision
Update Architecture Work Products
Models, Vision
Feedback
Models, Vision
Work With Developers

Figure 6: Outline Of Proposed Method Architecture

Quality Extensive Model together with Scrum Tree Algorithm Development is proposed to deal with challenges discussed earlier. Fig. 6 (above) shows the method architecture. The figure shows two main blocks as "Plan" and "Implement".

- The plan-block works on scrum objectives depending on requirements.
- The practice decision will likely be prioritized according to input goals/requirements.
- Further greater priority needs sent for evaluation to requirement block.
- Simultaneously, in iteration modeling block, project manager confirms requirement achievement or any kind of rework (if needed).
- Finally, scrum tree execution ends here to produce requirement/goal decision plan and sent to implementation block.
- As usual, after productive implementation working product is deployed internally.
- Our scope is till requirement preauthorization and decision plan modeling as per goals.

The intention for name “scrum-tree” for proposed algorithm is:

- To process decision tree structure for agile/scrum activities.
- To avoid the time consuming story-writing process in scrum.
- Instead of making huge scrum board, scrum tree may help each stakeholder with specific decision control.
- To get specific decision model based on goal modeling.

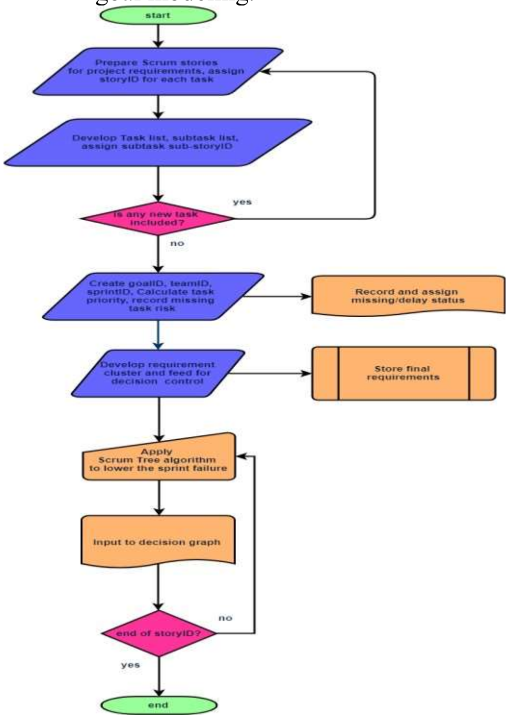
start
Prepare Scrum stories for project requirements, assign storyID for each task
Develop Task list, subtask list, assign subtask sub-storyID
Is any new task included?
yes
no
Create goalID, teamID, sprintID, Calculate task priority, record missing task risk
Record and assign missing/delay status
Develop requirement cluster and feed for decision control
Store final requirements
Apply Scrum Tree algorithm to lower the sprint failure
Input to decision graph
end of storyID?
no
yes
end

Figure 7: Flowchart Of Proposed Research

Proposed research (refer Fig. 7 above) is useful as instead of making huge scrum board, scrum tree may help each stakeholder with specific decision control so that time is saved and it will forward exact and correct requirement / goal decision plan to implementation block. After successful implementation working product is deployed internally. Here scope is in requirement preauthorization and decision plan modeling as per goals. Even if story description historical data is preserved using machine learning techniques, there is no need of any identical user story (scrum story) decoding. Following Algorithm-1 depicts the flow of computations by means of pseudo code.

Algorithm 1: Scrum Tree’ Algorithm

```
1. label_story_storyId = int storyID;
label_substory_storyId= int substoryID;
label_newstory = int NS; label_oldStory=int olds; label_substory_goalId= int goalID;
label_substory_taskId= int taskID;
label_substory_sprintId= int sprintID;
label_substory_teamId= int teamID; Initiate;
storyID=(0….N); substoryID=(0….M); goalID=(0….G); tasked= (0….Ta); sprinted= (0….Sp);
teamID= (0….Tm);
2. CurrentStory=Initial=i;
If storyID!=0; storyID++;
if substoryID=(0….M);
```

```
substoryID++; end if
storyID==N && substoryID==M; return
storyID, substoryID; storyID= substoryID=olds;
3. do
4. for each Ni ∈ M do
5. CurrentStory [i]=Update_newstory(NS);
6. until != (CurrentStory ==oldS);
7. return CurrentStory;
8. $G\oplus Ta = G+Ta - G\cdot Ta$
9. $Sp\otimes Tm = Sp\cdot Tm$
10. $G\ int=0,\ Sp = +\infty$
11. $G\oplus Sp = G+Sp$
12. $Tm\otimes Sp = (G\cdot Tm)/(Tm+Sp)$
13. for each Ta ∈ Sp target(Tm)== G do
14. Return all;
```

In this algorithm story, sub stories (if any), team, sprint, goals are assigned with numbers. The user stories are converted to scrum stories (manually) which further sequentially assigned with unique identification code. All these labels are initiated as “0”. Further, Goals are assigned to teams. Each task is divided into sufficient sprints as per requirements. Now, the coordination of Task, sprints and stories are cumulatively calculated automatically to set priorities as a goal.

This can be used by project managers to put on graphical representations. As the algorithm is developed using java language, the deployable “jar” file is built. Further, this “jar” file can be deployed to project management tool library to draw automatic decision tree graphs. The scrum tree algorithm output for product based software project is shown in Fig. 8 below.

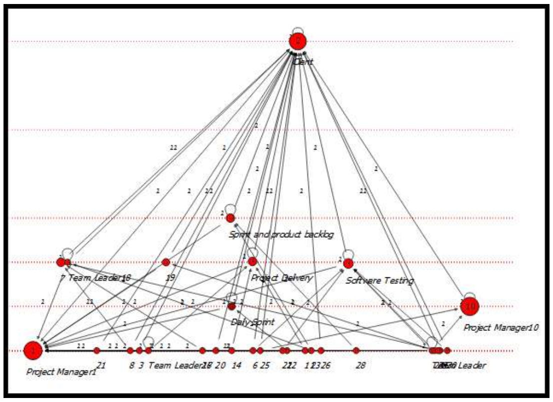

Figure 8. Scrum Tree Algorithm Execution Output For Product Based Project

The numbers of red color nodes are stakeholders of project with different roles and decision power. The correlation is shown by single headed arrows with decision index. The decision index is high if scrum story is dependent of other scrum story (i.e. client requirement). In such case, the interrelated decision tree is formed by scrum tree algorithm automatically to indicate common “sprint” need. Every product backlog is continuously checked for open sprint/closed sprint or retrospective. But, it project is process/support based or it is a maintenance project then scrum tree automatically defines hierarchy of responsibilities and decision tree is drawn. Such process based scrum tree execution output is shown in following Fig. 9.

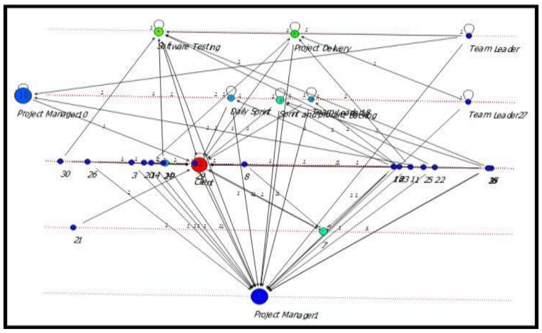

Figure 9. Scrum Tree Algorithm Execution Output For Process Based Project

Hence, as shown in figure 9, the bigger blue colored nodes are highly responsible stakeholders like project managers, software architects etc. This implies that, such stakeholders must register product backlogs, must review all the sprints and contribute to retrospective phase. The small blue dots are team member/team leaders who receive assigned module level sprints and reports to biggerblue dots for getting review. The green color dots are most responsible stakeholders who can correlate the decision, project changes and reports to project manager. The scrum tree algorithm automatically shares/passes the decision to respective stockholder.

As proposed research is focused to reduce manual decision making and erroneous decisions, scrum tree is best suited solution for agile projects where team size is bigger and complex to manage.

We compared the proposed work with Thurman dataset and scrum tree dataset and tested for real time project scenario. Next section discusses the performance analysis for proposed work.

4. RESULT AND ANALYSIS

Scrum Tree Process Flow

The proposed system is tested for performance with SocNetFs Tool with existing "Thurman" dataset and newly developed "scrumtree" dataset with 4000 user stories. As a scrum tree algorithm testing pre-processing, the actual project requirements are converted in to scrum stories. The process flow for proposed work testing is depicted in the following Fig. 10.

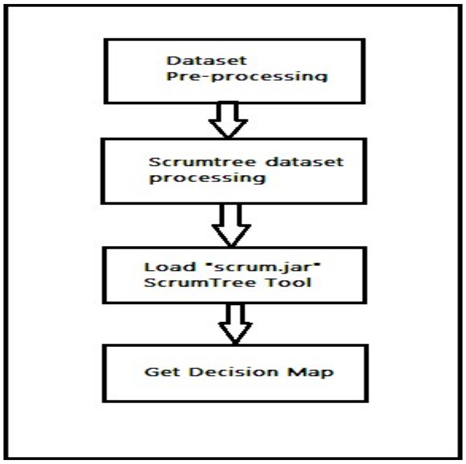

Figure 10. Representation Of Scrum Tree Process Flow

"Scrumtree" dataset is developed using scrum stories where each scrum story is assigned with storyId. This storyId is used as a unique number to process scrum tree decision modeling. The "scrumtree" dataset is developed in the format as shown in table 1 below.

Further using scrum.jar, "scrumtree" dataset is loaded to testing tool. As discussed previously, the data is processed by scrum tree algorithm and decision map/graph will be displayed. Proposed work is tested for real time scenario for scrumtree dataset and output is shown in Fig. 11.

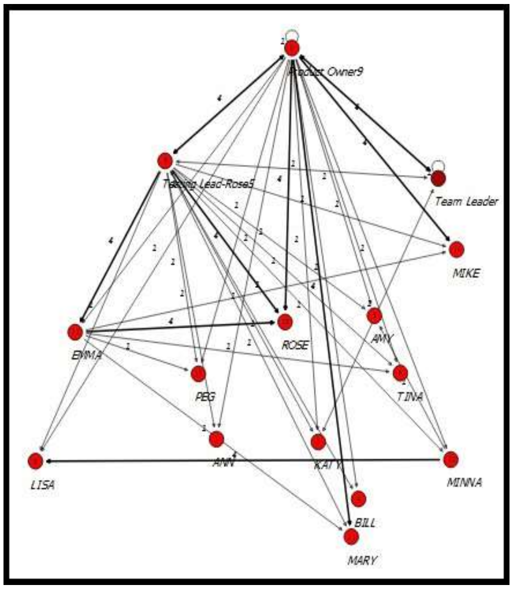

Figure 11. Real Time Testing Of Scrum Tree Algorithm On Thurman Dataset

Real time testing conducted for small team of 15 stakeholders, the team is comprised of 12 software developers, 2 team leaders and one product owner (in this case project manager).

The scrum tree algorithm provided appropriate decision/goal tree to be executed by each member of team.

As any new algorithm needs to be checked for its performance, the detail performance analysis is shown in next section 4.2.

Performance Analysis

As proposed system is developed for automatic decision control based on priority/importance of scrum user stories, scrum tree assigns product backlog, appropriate count of sprint based on retrospective status. Further, to evaluate proposed work for various parameters like data centrality, team closeness centrality, influential team nodes, and decision proximity along with sprint delivery decision the scrum tree algorithm embedded to library of Socnet for dataset.

Data centrality parameter shows the correlation of sprint tasks for developers. In Fig. 12, agile team coordination is mapped and correlations can be observed by two headed arrows. This implies, proposed work is appropriate for Agile development.

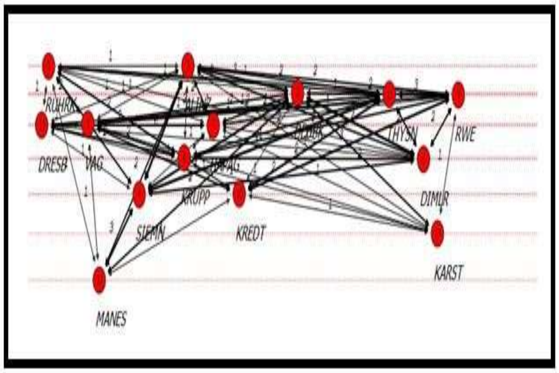

Figure 12. Data Centrality

Further, to analyze team task coordination "team closeness" in terms of decision sharing, we evaluated radial graph. Fig. 13 shows that, out of 12 developers; 9 developers are assigned with core development tasks and 3 are assigned with collaborative tasks with 2 team leaders and all are reporting to one product owner. Hence, team closeness is more in radial structure.

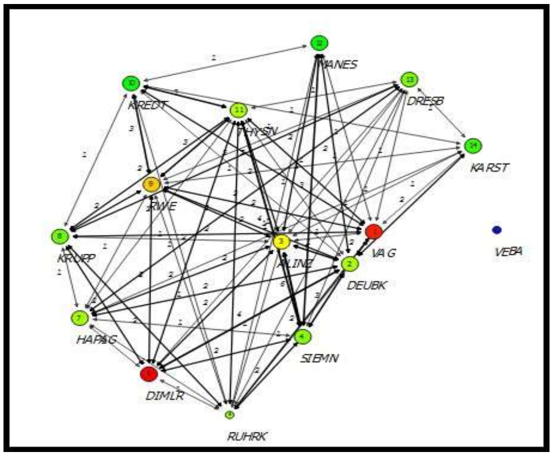

Figure 13. Agile Team Closeness Coordination

As team is a heart of any Agile project, it is important to provide right scrum user story to right team member, so scrum tree executes and provides influential weight to appropriate node (i.e. developer) who is with less task or previous task is finished or node who worked on similar task before. Such influential task decision is shown in Fig. 14 below. Here red color dots showing that two nodes are already with sufficient tasks and no new task can be redirected due to heavy influence. A yellow and orange color dot shows the influential nodes due to sharing of mutual tasks. Whereas green color dots can be assigned with new tasks as they can collaborate with core team and complete the tasks.

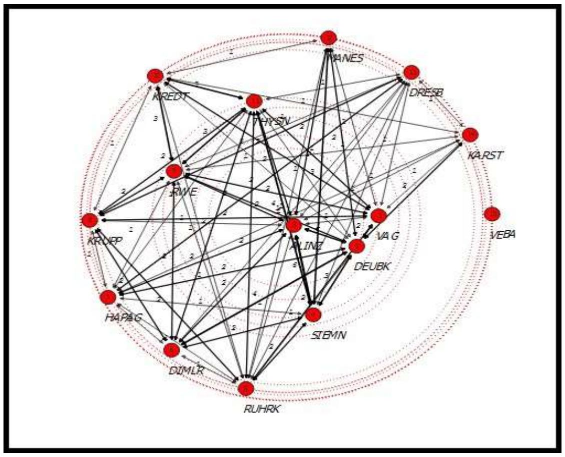

Figure 14. Scrum Tree Influential Team Nodes

This automatic mapping for influential team nodes due to scrum tree algorithm provides easy way for project managers to manage daily sprints and product backlog.

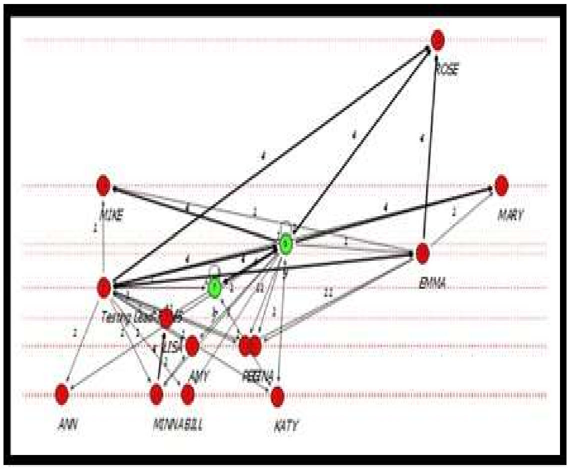

Figure 15. Scrum Tree Decision Proximity

Further, scrum tree also checks for decision proximity parameter where, already generated goals/decisions are closely checked for scheduled of task completion. Here, monitoring of task is assigned to green color dots (project manager/product owner). Where, project manager/ product owner can view the schedule of task, progress of task, sprint outcome, retrospective events of team. The task is shown in Fig. 15 as an edge number. For example, developer ROSE has edge value=4*3=12 which means his tasks are near to delivery. Where developer MARY has edge value 4 and 2 which means one task is about to complete with value 4 and other task is half completed with value=2.

As a last agile scrum stage for project is “Last Sprint Delivery Decision” which means, project is successfully reached to product delivery as an outcome of all daily sprints, product backlogging and it is in stage of retrospective. This is very important phase for scrum tree algorithm as it checks all developers are free from major task values (edge values) and ready to face any maintenance/modifications as per product owner expectations. This is represented by Fig. 16 shown below.

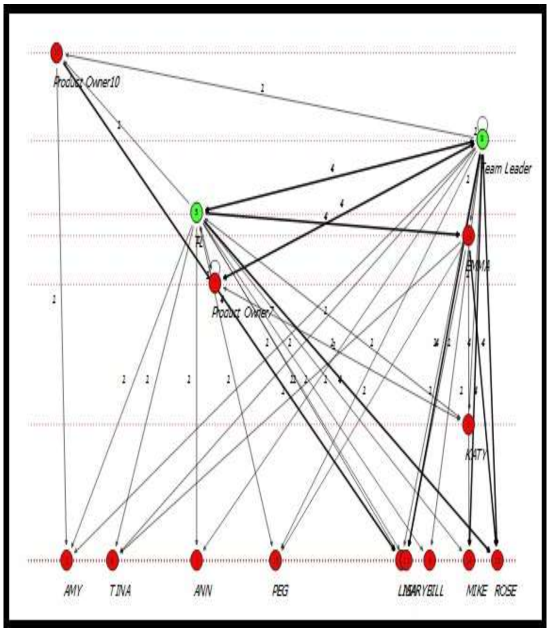

Figure 16. Scrum Tree Last Sprint Delivery Decision

Here, green color dots are teal leaders who can counter check the task completion based on edge value. The bottom level dotted red line shows the all developers in agile team, while rest of three red dots representing collaborative task contributors (developers) who interacting among them. The top level red dotted line specifies the product owner who monitoring bottom red line developers and one team leader. So, scrum tree algorithm provided a hierarchical way of representation.

Based on the table 2 data, performance comparison of Scrum Tree algorithm tool (STAT) with existing tools is shown in figure 17 below. This shows the STAT is useful from early stages of software life cycle for any agile project development and management.

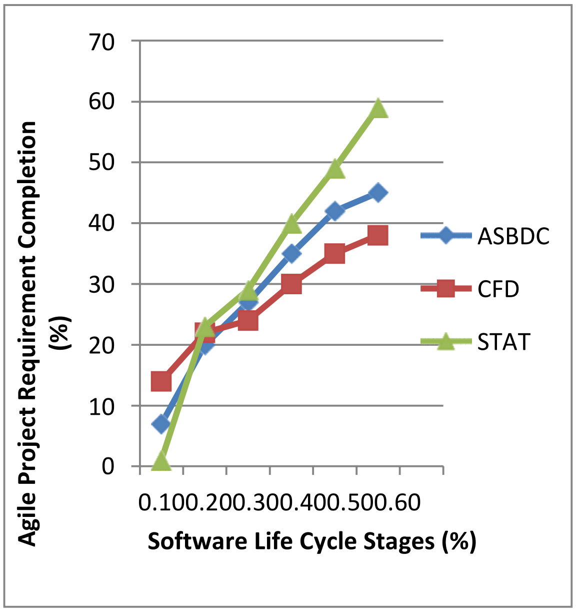

Figure 17: Performance Comparison Of Scrum Tree Algorithm Tool (STAT) With Existing Tools.

Based on previous discussion about Agile scrum burn down chart, Kanban Cumulative Flow Diagram tool and proposed Scrum Tree algorithm tool; we compared performance in terms of various parameter support and functionalities. Above table

2 shows the comparison results and shows that proposed work is more beneficial for any size of agile software project.

5. CONCLUSION

In this paper, agile scrum burn down chart, Kanban Cumulative Flow Diagram tool are studied to understand various functionalities and research gaps. As a new development of Quality Comprehensive Model, Scrum Tree Algorithm is developed by pre-processing scrum user stories. Such user stories and sub-stories (in case of complex user stories) are used as an input to system. Various parameters as data centrality, team closeness centrality, influential team nodes, decision proximity, sprint delivery decision, automatic task scheduling for project stakeholders are tested and it is noted that proposed work is with more parameter supports than existing tools. As a future enhancement, this tool can be deployed to cloud as an open source decision goal modeling scrum tool.

REFERENCES:

[1] Lei, Howard, et al. "A statistical analysis of the effects of Scrum and Kanban on software development projects." Robotics and Computer- Integrated Manufacturing 43 (2017): 59-67.

[2] Padmanabhan, Vasu. "Functional Strategy Implementation-Experimental Study on Agile KANBAN." Sumedha Journal of Management 7.2 (2018): 6-17.

[3] Taibi, Davide, et al. "Comparing requirements decomposition within the scrum, scrum with kanban, XP, and banana development processes." International Conference on Agile Software Development. Springer, Cham, 2017.

[4] Turetken, Oktay, Igor Stojanov, and Jos JM Trienekens. "Assessing the adoption level of scaled agile development: a maturity model for scaled agile framework." Journal of Software: Evolution and Process 29.6 (2017): e1796.

[5] Arthur, James D and James B. Dabney. "Applying standard independent verification and validation (IV&V) techniques within an Agile framework: Is there a compatibility issue?" Systems Conference (SysCon), 2017 Annual IEEE International. IEEE, 2017.

[6] Oliver, Peter, and Concise Reads. "Agile Software Development: Agile, Scrum, and Kanban for Project Management." (2018).

[7] Ghafoor, Fawad, Ibrar Ali Shah, and Nasir Rashid. "Issues in adopting agile methodologies in global and local software development: A systematic literature review protocol with preliminary results." International Journal of Computer Applications 160.7 (2017).

[8] Annosi, Maria Carmela, Antonella Martini, and Mats Magnusson. "Investigating the impact of agile control mechanisms on learning in scrum teams." Learning and Innovation in Hybrid Organizations. Palgrave Macmillan, Cham, 2018. 213-229.

[9] Mendez, Ellen L., and Diego P. Pinto-Roa. "Optimization model for assigning tasks in Scrum Agile Development (work in progress)." Proceeding Series of the Brazilian Society of Computational and Applied Mathematics 6.1 (2018).

[10] Alhazmi, Alhejab Shawqi. A Decision Support System for Sprint Planning in Scrum Practice. Diss. Florida Atlantic University, 2018.

[11] Hron, Michal, and Nikolaus Obwegeser. "Scrum in practice: an overview of Scrum adaptations." (2018).

[12] Ahmed, Muhammad, et al. "Estimation of Risks in Scrum Using Agile Software Development." International Conference on Applied Human Factors and Ergonomics. Springer, Cham, 2018.

[13] Khmelevsky, Youry, Xitong Li, and Stuart Madnick. "Software development using agile and scrum in distributed teams." Systems Conference (SysCon), 2017 Annual IEEE International. IEEE, 2017.

[14] Dhir, Saru, Deepak Kumar, and V. B. Singh. "Success and Failure Factors that Impact on Project Implementation using Agile Software Development Methodology." Software Engineering. Springer, Singapore, 2019. 647-654.

[15] Yousaf, Rizwan, Muhammad Hamid, and Saima Munawar. "Extended Scrum Process Model Using Software Reliability Engineering Concerns." journal of information communication technologies and robotics applications (jictra). ISSN# 2226-3683 8 (2017): 1-10.

[16] Khalil, Md Asif, and Bonthu Kotaiah. "Implementation of agile methodology based on SCRUM tool." 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS). IEEE, 2017.

[17] Mishra, Prabhanjan, et al. "Modeling Fault Detection Phenomenon in Multiple Sprints for Agile Software Environment." Quality, IT and Business Operations. Springer, Singapore, 2018. 251-263.

[18] Chandrasekara, Chaminda, and Sanjaya Yapa. "Effective Team Management with VSTS and TFS: A Guide for Scrum Masters." (2018).

[19] Bolloju, Narasimha, Rahul Chawla, and Rajat Ranjan. "Pros and Cons of Rotating Scrum Master Role: A Qualitative Study." Proceedings of the 11th Innovations in Software Engineering Conference. ACM, 2018.

[20] Sutherland, Jeff, Carsten Ruseng Jakobsen, and Kent Johnson. "Scrum and CMMI level 5: The magic potion for code warriors." Hawaii International Conference on System Sciences, Proceedings of the 41st Annual. IEEE, 2008.

[21] Falcone, D., et al. "An integrated model for an advanced production process-Agile Re-engineering Project Management." IFAC-PapersOnLine 51.11 (2018): 1630-1635.

[22] Alaidaros, Hamzah, Mazni Omar, and Rohaida Romli. "A Theoretical Framework for Improving Software Project Monitoring Task of Agile Kanban Method." International Conference of Reliable Information and Communication Technology. Springer, Cham, 2018.

[23] Ahmad, Muhammad Ovais, et al. "Kanban in software engineering: A systematic mapping study." Journal of Systems and Software 137 (2018): 96-113.

[24] Margaria, Tiziana. "Generative model driven design for agile system design and evolution: a tale of two worlds." International Workshop on Formal Methods for Industrial Critical Systems. Springer, Cham, 2018.

Table 1. Scrumtree Dataset Parameter Format

StoryId	**Requirement Id**	**Product Backlog Created? (yes/no)**	**Project Phase**				**Release (yes/no)**
			Sprint (1….n)	Sprint Backlog (yes/no)	Retrospectiv e	Next Sprint Needed (yes/no)	

Table2.Comparison Between Existing Kanban Cumulative Flow Diagram Tool And Proposed Scrum Tree Algorithm Tool

Parameters	**Agile Scrum Burn Down Chart (ASBDC)**	**Cumulative Flow Diagram (CFD)**	**Scrum Tree algorithm tool (STAT)**
allows the team to visualize their own effort	No	Yes	Yes
allows the team to visualize project progress	Yes	Yes	Yes
Shows distribution along the different process stages	Yes	Yes	Yes
Shows work left to do	Yes	Yes	
Uses user stories as a direct input to tool	No	Yes	Yes
data centrality	No	Yes	Yes
team closeness centrality	No	No	Yes
influential team nodes	No	No	Yes
decision proximity	Yes	No	Yes
sprint delivery decision	No	No	Yes
Automatic task scheduling for project stakeholders	No	No	Yes