CHAPTER 6

Improved Scrum Method through Staging Priority and Cyclomatic Complexity to enhance Software Process and Quality

6.1 Introduction

The technique of developing the software, utilizing a procedural model is called software engineering. Waterfall model, Prototyping model, Spiral model, V-model, Agile model, etc. are a few mostly followed traditional models. Each technique has its own advantages and disadvantages. The transition to agile methodology from traditional methods is often seen today. The Scrum, Extreme programming, Kanban, Test Driven Development are various flavours present in agile methodology. The scrum framework is mainly concentrated in this report. Less clarity and complexity are the characteristics that make Scrum process very hard (Akif and Majeed 2012). The outcome of this project is a unique framework which resolves the major problems that exist in Scrum. Software development plays a very major role in the industrial world (Lei et al. 2017).

Previously, organizations used to stick to traditional software models, but now there is a big shift to agile methodologies. The importance and popularity of the methods and principles of Agile Software development have increased a lot (AL-Taani and Razali 2013).

The trend is growing with an assortment of unique technologies and several approaches and is managed according to the conditions of the economic system, expectations, emerging trends, marker needs and etc. Enhancing the features of a system which is having more users is the main target of all product development (Serrador and Pinto 2015). Software development organizations often fail to produce and deliver the system on time and within the budget allocated (Yu and Petter 2014). A fresh approach which is much more matured has to be established which has greater convenience in order to sustain the competitive pressures among the various software organizations. The multiple opportunities corresponding to the organization play a really important role in approaching an issue. Some operations do not fit the current business scenarios (Brhel et al. 2015). The traditional development approach is one such theory which comes after a detailed procedure. An assurance of a high importance is to render the system func-
tions in minimal time with the high character and thus, a development methodology is crafted. When the prerequisites of a customer alone are satisfied, a system, in general, is supposed to be a quality one.

Agile process development is the primary and the foremost thing which comes into an individuals mind when he or she wants to play with any software. An alternate tactic to make a software Agile is fundamentally a time-boxed approach to building software in an incremental manner starting from the initiation of the project (Devedzic et al. 2011). Delivering all features at one go at the near end is not a sound thought. The changing requirements and essentials can be managed entirely by the agile process. Sprints are specially designed incremental work pieces which are released by the agile process. Customer requirement and change proposals are the central elements in the prioritization of the characteristics of this process.

The partners and the clients see agile methodology as a significant model in the current trends. Scrum, as we know, is the most important methods which are related to agile model and it is also widely used by the customers. Just like any other process, Scrum comes with a bunch of advantages and disadvantages. Substantial time consumption and new techniques are the two issues which scrum can be sorted into. Many problems are listed out based on the literature and background review. When agile methodology goes wrong, it implies that there exist a complexity and difficulty in the software engineering procedure.

Agile and Scrum processes are the key concepts of software engineering practice. Following agile practice has its own advantages in both the company perspective and the client perspective. Having feared for adopting new techniques, in the year 2001, many companies which were started up by then did not even think about agile methodology (Gandomani and Nafchi 2015).

The main objective of introducing a new framework is, for enabling easier access and also for satisfying various problems. This model should also assist in breaking up various challenges existing in agile methodologies, primarily in Scrum and thereby help the firms and/or the society to create quality and high-end software within a time frame in order to increase the credibility of the product to enhance the business value of the node.

6.2 Related work

Scrum, Kanban, Feature Driven Development, Test Driven Development, Extreme Programming, etc. are the various methodologies present in Agile. In this presentation, we focus on Scrum as it is greatly used in Agile.

The framework of agile software consists of Scrum as one of the popular processes. Complex system and products are the outcomes of Scrum (Paasivaara and Lassenius 2014). Ken Schwaber stated that “Scrum employs an interactive and incremental ap-
proach to optimize predictability and control risk”. The three main people who are principally involved are (Heikkilä et al. 2015)

1. Owner of the Product.
2. Scrum Master.
3. Project Team or Scrum Team.

The owner is involved at the state at which the process of Scrum begins. The end-users, customer, and stakeholders give inputs to the proprietor of the product and the input consists of priorities of the owner regarding product backlog and the list of features in an edict of the requirements adopted in the business and they are kept active and are owned by the product owners. The sole soul who can edit and revise the backlog is the product owner. For each and every Sprint, there exists a spring backlog, which is drawn by the team from the product backlog.

The period of time that the team promises to work and fulfill the needs of the project is called Sprint. It is the responsibility of the owner of the product and squad to decide Sprint (Alzoubi et al. 2016). Throughout this process of the sprint, the end date and the work that has to be delivered by the team is kept constant. Termination and initiation of a new one can only pass off at the discretion of the product owner if something sounds awry 5-9 people are basically a portion of the team Scrum. The characters of the team range from self-establishment and self-management and primarily cross-functional behaviour. As Sprint comes to end the team would have decided the work that had to be delivered through that period of time. The rescue is done based on a design created by the squad. Experience level is not a hurdle for any member to take part and all are equally participating in decision making. A burn down chart is held and updated each day, which indicates the updates and the time or the total hours given to finish the projects. The progress and blocks are updated during the regular meetings conducted each day. The meeting lasts for about 15 transactions where everyone mainly focuses on reporting on three central periods (Bougroun et al. 2014).

1. Done since yesterday.
2. Done by tomorrow night.
3. Blocks and obstacles.

The notice of blocks and obstacles is done by the Scrum master who also assists the team to break up all the troubles (Jan Vlietland et al., 2014). A former project manager or a team member can be holding the position of a Scrum Master. Protecting and the serving the team is done by this person itself. This individual assists the whole team.
to eradicate any obstacles and also assists the team to stay immune from whatever kind of noises and shows the team the right way. The squad will invariably possess a high danger of bankruptcy without a proper Scrum master. A Scrum Master aims at finishing any task ideally to an extent of 100% of what they committed an increment, which is essentially shippable for the welfare of the labour. Testing and implementation of every shippable increment must be performed without any major flaws.

Scrum consists of two critiques that are declared before the Sprint. In the sprint review meeting, the owner of the product, scrum master, etc. meet and verify the working of the software and assess the feedback to be applied. The retrospective meeting is essentially only for the team, product owner, and the Scrum master. They also pile up by the conclusion of each and every sprint to have a check on the oeuvre and its potency (Ihme 2012).

Sprint review is termed as a “Product review” and Retrospective is termed as a “Process review”.

Scrum has got numerous generic and technical subjects connected to it (Batarseh and Gonzalez 2015). Generic issues which were already spoken about have not found a proper solution yet (Shrivastava and Rathod 2015).

6.3 Objective

- To discover an approach which can prioritize requirements by considering customer priorities.
- To provide more agility to the software product or process by reduction of time for release.

6.4 Issues in scrum

The list mentioned below includes the technical problems related with Scrum which is regarded from the literature review and background study.

6.4.1 Assigning PBI

The primary task of assigning PBI is the conversion of stories of the user into the items of the Product Backlog. In the Software Requirement Specific document, when the PBIs are assigned there always exists a lot of disarrays. On that point is a lot of matters where the PBIs are given, preferably from the functionalities of software that needs the node for the Scrum team, PBIs assignment and in few situations, PBIs are given established for the sprint duration (Hoda et al. 2013). The above examples are not proper ways to assign PBI, due to the fact that a functionality of the software can eventually be a PBI but all of them cannot be functionalities. PBI cannot be assigned to the scrum
team since it does not possess impending on the project, whereas assigning PBI can only be done by the Owner and Scrum Master (Rola et al. 2016). An early point of Scrum is the assigning PBI and if that is wrongly done, then every part of the project gives out and inaccurate delivery of increments occurs.

6.4.2 PBI Prioritization

Product Based Backlogs (PBI)s are the converted entities of user stories in the scrum. The client and the society will get benefited if and only if PBIs are prioritized. Due to the varying nature of the agile environment, prioritization of user-news reports is very hard. Some basic characteristics have to be viewed to prioritize PBIs (Daneva et al. 2013)(Popli et al. 2014). PBIs, give the freedom to choose own priorities to client and company. PBI is prioritized based on the criticality of the PBI, to run on a particular PBI a necessary type of scrum team has to be available, in which one will have greater software reuse and priorities of customers based on a need and business value of PBI. The above different priorities may not be the same and in that situation, it is required to prefer a prioritization method that primarily gives importance and speaks about the characteristics possessed by both the priorities of the client and society. Scrum is not equipped with this technique and this is a major problem.

6.4.3 Regression Testing

Testing software after it has been modified to verify that the current functionality gets affected or not is called Regression Testing. As software is developed and organized Testing is one of the most expensive processes that occur. Retesting is an important concept wherein a prominent component of growth and maintenance cost are occurred due to the regression testing. 80% of the overall monetary value of testing is used for regression testing and 50% of the full cost is used for software maintenance (Pino et al. 2010, Logue and McDaid 2008). There are tons of challenges for prominent regression testing in practice due to the rapid change in software and computing environments. Regression testing can also be performed after alterations are made to the current software, as and when a new software is released, software is stored and compiled each time, as in agile environments or prior to releasing patches(Elallaoui et al. 2016). Improvisation of the confidence that leads to changes which behave as programmed and the fact that they do not move the previous functions of the software are the primary goals of regression testing. The software being developed always adapts to changes according to the working of Scrum and in this instance, it is important to create the modifications and then perform a regression test to fit the software (Inayat et al. 2015, Vlietland et al. 2016). The last rescue of the PBI is shown by the implementation of regression testing with every integration of PBI in a fashion which it does not generate bugs during inte-
6.4.4 Integration of PBI

In the Scrum process carried in agile methodology, once PBIs development is finished, an increment will be presented to the customer. Each and every PBI is then mixed with the Scrum team after all the PBIs are completed and delivered. A major problem which comes up here is that the testers will have to perform regression testing for all integrations that exist which in turn brings up the testing cost and the time. Regression testing prominently uses 80% of the monetary value of testing. Integrating PBIs with reduced regression testing has no proper model or method as of immediately (Blom 2010).

6.5 Proposed methodology

In parliamentary law to master the issues faced in Scrum a new framework which can likewise be shouted as an extension of Scrum is developed for Software Engineering and Management. This RScrum (Refined Scrum) solves the major events and challenges of the previous agile methodologies and helps the fellowship to bring forth high-quality software in a minimal time and also improves the business value of the node Fig 6.1.

6.6 RScrum Architecture

![Fig. 6.1 Proposed RScrum Architecture](image-url)
6.7 Architecture specifications

6.7.1 Getting UserStories

The very first measure of the scrum process model is to get user stories from the customer, which is same as requirement analysis in the Waterfall model (Tessem 2014). The user levels are then converted into Product Backlog Items (PBI). Each PBI represents the demands which are needed for the execution of that particular PBI.

6.7.2 Prioritization of the PBI

![Fig. 6.2 Prioritization of PBI](image)

The next step is to push all the PBIs into a stack. User-stories prioritization is an environment that is difficult to task due to its volatile nature. Ignoring the critical user stories will produce many problems like the client not satisfied, bad quality of the product. According to the literature survey, user-stories prioritization is not properly done in Agile.

Therefore a method is required to prioritize user-story based on client importance and also the company so that one can get a software project with respect to quality and also agile.

Each PBI has its own priority value which is obtained from the priorities given from client and the company. In RScrum model, the following method is used to get the final priority value from companys and the clients priorities. For each of the PBI get the priority values (1,2,...,n where 1 being highest priority and n is the least priority)
and the total number of modules of the system) from both the company and the client. The company's priorities and the client's priorities are taken in two separate arrays and then compared. The lesser priority value is taken as the final priority value. If the final priority array has same values between the modules then the final priority goes for the company's side. If it repeats again then the client prioritized module for the previous one gets the next priority. And the same method follows till each module gets unique priority value Fig 6.2. Here the staging priority value is 2,1,1,2 and 5 and four of the modules have same two values. This staging array is converted into the final array with values 3,1,2,4 and 5 as per the above description.

6.7.3 RScrum Planning and Scheduling

In this phase, the prioritized PBIs are then converted into Sprint Backlog Items (SBI). SBI can be a collection of one or more PBIs which on implementation makes a potentially shippable increment Fig 6.3. Then the planning and programming for the execution of the SBI take place which involves which team to do the SBI, when to do the SBI, number of weeks for the SBI, etc. These planning and scheduling are done in Sprint Planning Meetings. The Scrum Master and the development team should be involved in these Sprint Planning Meetings (Vlietland and van Vliet 2015).

![Fig. 6.3 RScrum Planning and Scheduling](image)

6.7.4 RScrum Coding

The next phase of the RScrum process model is to pop out the top priority PBI/SBI from the prioritized stack and the development team starts to code that particular PBI. In Scrum framework as addressed before in during coding, we have the Developer Tester problem. To overcome this problem, the Extreme Programming (XP) frame-
Algorithm 4 Staging priority

BEGIN
STEP 1: Get clients priority to the module
For i=1 to n
STEP 2: Get C1[i]
Get companys priority to the module
For i=1 to n
Get C2[i]
STEP 3: Compare clients priority to companys priority for each module and finally assign the highest priority
if(C1[i]==C2[i])
Pi=C1[i]
Else if(C1[i]<C2[i])
Pi=C1[i]
Else
Pi=C2[i]
END if
END for
STEP 4: Generate Pi
OBTAIN Pi
END

work is merged with the RScrum framework. The Developer- Tester problem results in poor code quality and XP concentrates highly on code quality which in turn becomes the solution for the problem. The XP programmers code the PBI as per the scheduled duration as planned in the previous Planning and Scheduling phase. The Pair Programming concept is the key to solving the problems in coding and testing phases of the Scrum framework. In Pair Programming, two XP programmers working the same workstation. One programmer will do the coding for the PBI and the other writes the unit test cases for the code simultaneously. Sometimes the unit test cases are written for the code even before the programmer codes it which will help the programmer to analyze and plan before coding which directly reduces the testing time and cost. Herein RScrum, not only the Pair Programming concept is used but also the Pair Processing of two different process models, Scrum and XP are used. XP has the disadvantage of poor planning which is countered by the RScrum Planning and Scheduling phase. Thus Pair Processing works seamlessly by negating each processes weaknesses.

6.7.5 Iterative Release of PBI

The coding of the Sprint Backlog is done, tested and concluded by the development team and the toleration of the Product Owner, the Scrum Master will answer a last verification of the Sprint Backlog as it is a potentially shippable increment or not. If yes, then the Sprint Backlog is delivered as an increment to the customer. If no, then the
Scrum Master has to operate for either Sprint correction or Sprint Restart. Sprint Correction is performed based on the account given by Scrum Master. Largely based on company related matters. Sprint Restart occurs if the Sprint is totally wrong or the development team cannot cope up with a sudden change in the requirement the Scrum Master declares the Sprint as a failed Sprint and immediately the development squad has to go to Sprint Restart.

6.7.6 Stack Condition

After the rescue of every increment, the next phase is the status of the stack. The development team will break the stack count, if it is greater than or equal to 1, then the next step is to get back to RScrum Planning and Scheduling and the whole process cycle is repeated till the upcoming stack condition. If the stack count is 0, then the next step is to mix all the Sprint Backlogs/increments and test the last product and present it to the customer.

6.7.7 Integration and Release

![Fig. 6.4 Integrating increments based on Cyclomatic Complexity](image)

When the prioritized PBI stack is empty, then all the Sprint Backlogs are completed and delivered incrementally to the client. Now, all the delivered increment shave to be integrated into a final single increment which is the software that is to be produced. In Scrum framework during the integration of the increments, as mentioned earlier in we have some issues. The major issue during the integration is that after integration,
the regression testing takes much time and cost from the overall test budget and plan and there is no proper logical algorithm for integration. These issues can be tackled by RScrum by the following methods Fig 6.4.

The concept of Cyclomatic Complexity is used as a solution to overcome the problem. The first step is to find the Cyclomatic Complexity value of all the increments or Sprint Backlogs which are delivered from the CC formula, which is

\[
\text{Cyclomatic complexity} = E - N + 2P
\]

where \(E\) = number of edges \(N\) = number of nodes and \(P\) = number of independent paths (Or) [Cyclomatic complexity]

\[
= d + 1 \text{ where } d = \text{number of decisions}
\]

Then next step is to integrate the increments which are having the same Cyclomatic Complexity value Table 6.1. Then calculate the CC value for the integrate increments and do the same. If the CC values are not equal then integrate the increments which are having the same number of disconnected paths Table 6.2. If both the CC value and also the P value are not same then integrate the increment shaving fewer difference values between them. This method helps in reducing the time and cost of regression testing mechanism. The reduction in time gives agility to the software process directly.

Later all the functionalities are trained, they are tested individually to check if they work correctly. If they do not work properly, then necessary changes are carried out. Nowadays, formerly all the modules are tested they are integrated all together, usually; regression testing is carried to make sure that the changes made have not caused unintended effects in previously working modules of software/application. If any exception

<table>
<thead>
<tr>
<th>S.No</th>
<th>INPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>E-N+2P</td>
</tr>
<tr>
<td>2</td>
<td>INPUT:E(number of edges)</td>
</tr>
<tr>
<td>3</td>
<td>N(number of nodes)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S.no</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CC(Cyclomatic complexity)</td>
</tr>
<tr>
<td>2</td>
<td>BEGIN</td>
</tr>
<tr>
<td>3</td>
<td>Calculate CC=E-N+2P</td>
</tr>
<tr>
<td>4</td>
<td>OBTAIN CC</td>
</tr>
<tr>
<td>5</td>
<td>END</td>
</tr>
</tbody>
</table>

Table 6.1 Cyclomatic Complexity Pseudocode Input(a)

Table 6.2 Cyclomatic Complexity Pseudocode Output(b)
occurs, then it is correct and again the regression testing is conveyed. Regression testing is carried on until there are no exclusions. But after integration, the regression testing takes much time and cost from the overall test budget and plan and there is no proper logical algorithm for integration. And then there is an alternative approach to defeating this issue, i.e., Conducting Regression Testing to all the test cases at one time makes it a complex task to detect mistakes and becomes unmanageable due to time and budget restraints. And then the trial instances are taken and conducted regression testing. Let Tn be the total number of test cases. Now we choose test cases step by step and conduct regression testing.

- Tsc1 = Test cases which have frequent defects are selected first and tested.
- Tsc2 = Test cases which represent core features are selected next and tested.
- Tsc3 = Test cases that have undergone more and recent changes are tested next.

Let T be the time taken for conducting regression testing to all the test cases. Let t be the time taken for conducting regression testing to selective test cases.

\[ t = t_1 + t_2 + t_3 \]

where

- \( t_1 \) = time taken for Tsc1
- \( t_2 \) = time taken for Tsc2
- \( t_3 \) = time taken for Tsc3

\( t < T \) (time taken for Selective Case Regression Testing is less than that of Time taken for Regression testing)

6.8 RScrum Mangements

6.8.1 Project Management

Software project management is the art and a scientific discipline to plan and lead the software development projects. It is said to be labour management sub-discipline that includes software projects plan, implement, monitor and control. The R-Scrum projects are maintained by the Scrum Masters, who are assigned to their respective projects by the Management (Moe et al. 2010).
6.8.2 Resource Management

Resource Management is the process of managing the resources of the project. The resources involve time, cost, manpower, etc. Time management is the key in RScrum as it restricts the power of the software process. Price determines the total amount spent and ROI for a software project. Better software engineering results in better ROI. Thus management of cost is very significant. Human resource management involves identifying the correct individual for the proper user. In RScrum the human resource management plays a major character because the Scrum Master and the development team is the key to RScrum success. Thus good HR management gives better productivity and higher success rate.

6.8.3 Change Management

Change Management is one of the most important aspects in R-Scrum because adapting change is the key feature in Agile Fig 6.5. Change management is a method that includes steps like identify, document, analyze, prioritize and agree on changes to. Then it also includes controlling the changes and informing that to specific stakeholders [RashinaHoda, et. al., 2013]. A part of vital importance in Software Engineering process is the analysis of change impact of altered or new scope, that has Requirements analysis at the change level; whereas changed needs and requirements of an end-user or client are identified by the software developers and business analysts. Once these requirements have been identified then they can make some changes or re-design a solution. In general, every change may impact the budget and time required for any software project. Hence risk-based analysis should be included prior to approval.

The Version Control Management is the subset of Change Management that handles all patches, versions, and updates released by the company for particular software.
6.9 Results and Discussion

In Scrum, there is no specific prioritization technique to prioritize requirements. The product owner gathers the requirements from clients and stakeholders, go ahead to do a feasibility survey and prioritize according to his knowledge Table 6.3.

**Table 6.3 Feasibility Survey**

<table>
<thead>
<tr>
<th>S.No</th>
<th>Requirements</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>User interface</td>
<td>5</td>
</tr>
<tr>
<td>R2</td>
<td>Security</td>
<td>3</td>
</tr>
<tr>
<td>R3</td>
<td>Portability(Platform Independent)</td>
<td>1</td>
</tr>
<tr>
<td>R4</td>
<td>Performance(Speed,Reliability)</td>
<td>2</td>
</tr>
<tr>
<td>R5</td>
<td>Memory Efficiency</td>
<td>4</td>
</tr>
</tbody>
</table>

\[ \text{Mean}(x') = \frac{\text{Sum of priority of each requirements}}{\text{Total number of requirements}} = \frac{1}{n} \sum_{i=0}^{\infty} x^i \quad (6.2) \]

Where \( n \) = number of requirements

\( X_i = \) priority of \( i^{th} \) requirement, \( i=1,\ldots,n \)

\[
\text{Mean} = \frac{(5 + 3 + 1 + 2 + 4)}{5} = \frac{15}{5} = 3 \quad (6.3)
\]

Disagreement is the fact or quality of one requirement being different from one another. It is the average of the squared deviations from the mean.

\[ \text{Variance}(\sigma^2) = \frac{1}{n} \sum_{i=1}^{\infty} (x_i - x')^2 \text{ Where } x' = \text{Mean Priority} \quad (6.4) \]

\[
\text{Variance}(2) = \frac{1}{5}((5 - 3)2 + (3 - 3)2 + (1 - 3)2 + (2 - 3)2 + (4 - 3)2) = \frac{1}{5}(4 + 0 + 4 + 1 + 1) = \frac{10}{5} = 2 \quad (6.5)
\]

Standard deviation measures the variation or dispersion amount for a data value set. A standard deviation that is close to 0 denotes that data points are tending to be real high to mean that is also called as the expected value of that band and a greater standard deviation denotes that the data points are distributed over a range of values. Standard deviation is the square root of the variance.
\[ \text{Standard deviation}(\sigma) = \sqrt{\text{Variance}} = \sqrt{2} = 1.414 \] (6.6)

In RSCRUM we follow a prioritizing technique called staging priority to prioritize requirements to meet both customer and company Table 6.4.

**Table 6.4 Staging Priority**

<table>
<thead>
<tr>
<th>S.No</th>
<th>Requirement</th>
<th>Client Priority</th>
<th>Company Priority</th>
<th>Staging Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>User Interface</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>R2</td>
<td>Security</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>R3</td>
<td>Portability (platform independent)</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>R4</td>
<td>Performance (Speed, Reliability)</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>R5</td>
<td>Memory Efficiency</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ \text{Mean Priority} \left( x' \right) = \frac{\text{Sum of priority of each requirements}}{\text{Total number of requirements}} = \frac{1}{n} \sum_{i=0}^{\infty} x^i \] (6.7)

Where \( n \) = number of requirements

\( X_i = \) priority of \( i^{th} \) requirement, \( i=1, \ldots, n \)

\[ \text{Mean} = (1 + 2 + 3 + 2 + 1)/5 = 9/5 = 1.8 \] (6.8)

Disagreement is the fact or quality of one requirement being different from one another. It is the average of squared differences from the mean.

\[ \text{Variance} \left( \sigma^2 \right) = \frac{1}{n} \sum_{i=1}^{\infty} (x_i - x')^2 \text{ Where } x' = \text{Mean Priority} \] (6.9)

\[ \text{Variance} \left( \sigma^2 \right) = 1/5((1 - 1.8)^2 + (2 - 1.8)^2 + (3 - 1.8)^2 + (2 - 1.8)^2 + (1 - 1.8)^2) \]
\[ = 1/5(0.64 + 0.04 + 1.44 + 0.04 + 0.64) \]
\[ = 2.8/5 \]
\[ = 0.56 \] (6.10)
6.10 Comparing the standard deviation of Scrum and Rscrum processes

\[
StandardDeviation(\sigma) = \sqrt{\text{Varience}} = \sqrt{0.56} = 0.75 \tag{6.11}
\]

The standard deviation of the RScrum Process Model is less than that of the SCRUM process.

\[(0.75 < 1.4114)\]

6.11 Result Analysis Graph

The standard deviation of both Table 6.5 and Table 6.6 Scrum and R-Scrum are plotted along a graph Fig 6.6 and Fig 6.7 for a neat and clear understanding and comparability.

**Table 6.5 Scrum Result Analysis(a)**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Mean</th>
<th>Varience</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RScrum</td>
<td>1.8</td>
<td>0.56</td>
<td>0.75</td>
</tr>
<tr>
<td>Scrum</td>
<td>3</td>
<td>2</td>
<td>1.414</td>
</tr>
</tbody>
</table>

**Table 6.6 Scrum Result Analysis(b)**

<table>
<thead>
<tr>
<th>Timeline</th>
<th>Sum of Tasks</th>
<th>Sum of Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Scrum</td>
<td>Rscrum</td>
</tr>
<tr>
<td>0</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>2</td>
<td>29</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>25</td>
<td>26</td>
</tr>
<tr>
<td>8</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>10</td>
<td>23</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>18</td>
<td>21</td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td>11</td>
</tr>
</tbody>
</table>

Lesser the standard deviation, better the priority is turned over. Thus Staging Priority followed in RScrum produces better consequences than that of Scrum Process. For integration, we are applying the concept of Cyclomatic complexity which helps us in discovering the defects earlier and in detecting the location of the fault. Once integrated they have to undergo through a step by step testing process as mentioned above to realize that everything has been set up correctly without disturbing the old.
Fig. 6.6 Comparison of Scrum and Rscrum SD

Fig. 6.7 Comparison of Scrum and Rscrum ST
6.12 SUMMARY

The proposed RScrum process model overcomes the technological issues faced in Scrum methodology. The requirements are prioritized in an efficient way using staging priority with a less standard deviation which means that priority is well distributed between modules. The modules that are developed completely integrated based on Cyclomatic complexity which helps in cutting down the time and monetary value of regression testing mechanism. The decrease in time gives agility to the software process. This step by step measure of the testing procedure makes work easy to aid in the detection of faults faster when compared to the current Scrum model. Software quality would be better attained if defects are detected earlier in the software development life cycle model.