ESCALATING THE ROLE OF REQUIREMENTS TRACEABILITY IN CHANGE MANAGEMENT

By

Tauheed Waheed

Enrollment No: 01-241172-056

Supervisor: Dr Shahid N.Bhatti

A thesis submitted to the Department of Software Engineering, Faculty of Engineering Sciences, Bahria University, Islamabad in the partial fulfillment for the requirements of a Masters degree in Software Engineering

July 2020

Approval Sheet

Thesis Completion Certificate

Scholar's Name: Tauheed Waheed Registration No: 53444
Programme of Study: Software Engineering
Thesis Title: Escalating the Role of Requirements Traceability in Change Management

It is to certify that the above student's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for Evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at 2% that is within the permissible limit set by the HEC for the MS/MPhil degree thesis. I have also found the thesis in a format recognized by the BU for the MS/MPhil thesis.

Principal Supervisor's Signature: _____

Date: 4th July,2020 Name: Dr. Shahid Nazir Bhatti

Certificate of Originality

This is certify that the intellectual contents of the thesis

"Escalating the Role of Requirements Traceability in Change Management"

are the product of my own research work except, as cited property and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare and extend the earlier work. Further, this work has not been submitted by me previously for any degree, nor it shall be submitted by me in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my degree.

Signature: _____ Date: 4th July,2020

Name of the Research Student: Tauheed Waheed

ABSTRACT

Every time the requirements are modified it doesn't maintain the origin of the requirements and requirements traces. We don't have the appropriate mechanism to track the origin of the requirements at first place. This research describes the pivotal role of requirements traceability in change management. The whole literature is developed to further emphasize the need of requirements traceability in change management. We have proposed requirements traceability metrics through which change in requirements of the product can be managed and after identification of changed requirements it can be tracked appropriately. Moreover, the requirements traceability metrics can operate as a control mechanism in order to prevent changes too far from the intended evolution of the product. We believe that the findings of this experimental-based study will enhance our and software engineers understanding of requirements traceability in perspective of change management. Further, it will make its contribution in the research domain as well.

Keywords: Requirements traceability, Change management, Control mechanism

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBRIVIATIONS

1. RT	Requirement Traceability
2. FT	Forward Traceability
3. BT	Backward Traceability
4. BDT	Bi-directional Traceability
5. CM	Change Management
6. RTM	Requirement Traceability Metrics
8. TORUS	Traceability of requirements using splices
9. TL	Trace Links
10. TRL	Traceability Representation Language
11. TLE	Trace Link Evolver
12. ALM	Application life cycle management

CHAPTER 1

INTRODUCTION

In the recent few years, change has been considered as the most challenging thing to adapt and cater in software development. The support of traceability creates much more rapid and reliable development process. In change management[1]the requirement traceability tend to identifies relationship between the requirements of customer through a (Requirement Traceability Matrix).Traceability intends to describe relationships among two or more entities in whole software development process.

Success of requirements traceability merely depends on the identification of appropriate traces, mainly from the traceability pool. What are those factors on which change management evaluate and accept particular traceability requirements? How the results of this research will affect the utilization of traceability in change management? To answer these entire questions research model is developed, and literature is about need for traceability. This experimental based empirical study discloses all the paramount factors that enhance the role of requirements traceability and metrics in perspective of change management. However, the need for identifying traces of suspicious quality is really important because it will largely affect the whole requirement traceability process and change management. Further, it really questions the abilities of software engineer to implement appropriate traceability approach to identify these traces and recognized them in order to make the requirements more traceable.

According to study published [2] in the Journal of Software Engineering shows that traceability support can enhance 50% accuracy of project and 24% the speed of a development project. That's why we are intended to implement the requirements traceability metrics in order to make the requirements traceable and inevitable to change.

Requirements tracing has been a focus of research and interest for decades. The traceability tool Pierce is commonly referenced as only historical example of the traceability tool. It was developed[3]"Requirements Tracing Tool" to aid change analysis and V&V by tracing the requirements throughout the various phases of software development. It was initially used on a naval system and afterwards it was reused to support the mission-critical project, related to cruise missile. The tool intends to focus on the development of a database that picks keywords from the specific requirements.

There are multiple ways to implement requirements traceability and every author has its own views on the role of traceability in software development. In the software development, traceability recognizes documents and the relationship among entities[4] from the set of appropriate structure. It tends to define the vital elements to recognize whether each related product is still paramount or not. There are several special features or characteristics that can differentiate requirements traceability as core of requirements engineering. Traceability[5] can serve various purposes, such as improvement, internal follow-up and evaluation of development efforts. . Implementing a process of traceability without a metrics is of course entirely possible but the performance characteristic of metrics provides the requirements traceability an even more trustable value.

The impact of traceability in software engineering is recognized as a paramount concern with considerable research and technical efforts invested in solving its hurdles or challenges. In order to track the requirements proactively, accuracy is especially important: requirements traceability metrics[6] intends to provide that proactiveness and accuracy. Traceability as a whole helps and protects against the software defects. It ensures that end product meets the user's requirements that were initially documented for the development. In order to achieve adequate requirement traceability, a more traceable territory must be well-established and protected. The environment created

should be composed of methods, techniques, approaches and tools to achieve the overall traceability process. However, requirements traceability[7] usually involves delving into a numerous of artifacts and trace links. Though artifacts consists of different natures, such as source code files, requirements specification and test cases. Further, these all artifacts obstructs in tracing a requirement through various abstractions. So, requirements traceability can be a time consuming and burden some task.

The information retrieved by utilizing traceability decreases the number of bugs in a product, which further increases[8] the overall quality and speed to complete the product. When we are referencing about removal of bugs or decreasing the bug rate, then we must consider or follow some quality standards. The traceable requirements importance can be observed in safety regulations, maturity models and the requirement standards. In general traceability is maintained, it has been managed by humans which can make wrong decisions. That's why existing traces can be of suspicious quality. However, these traces are responsible for the future very high impact strategies and decisions. Afterwards [9]requirements are implemented on the basis of traces between requirements and code reveal. These traces are essential for the understanding of code and change management.

The quality of traceability links and traces really impacts the whole system, specifically change management. The process of change management is quite vital when we consider change in requirements for real time or safety critical systems. The process of updating traceability matrix is cumbersome and time consuming; it always creates the risks of error from human side. That's why in the development of mission-critical systems like avionics and medical, the high risk can't be tolerable. Further, it is quite difficult to be proven by QA team.

The process of RT and implementing it in perspective of change management is not so straightforward due to the fact that the identification of quality traces and its appropriate linking required some expertise. Therefore, a guideline for RT can be useful for software engineers and requirement engineers for better understanding the concept of RT. We present a guideline for RT, as our first contribution, after categorizing them on the basis of features they provide. Requirements Traceability features present a rich ecology and understanding that where new traceability techniques get introduced in a system. In the same manner, we are intending to implement the change by assessing the

features covered by various traceability techniques. Afterwards, we will be in better position to implement appropriate traceability techniques and metrics with respect to degree of change in the system. We use our RT guideline to find out the impact and allocating the reasons for requirement traceability in change management.

The research has been carried out to understand the need for requirement traceability and what are the factors which enhance the role of RT in change management. This study tends to explore the inspiring energies in RT in perspective of change management. When the software engineers decide to change their system, they often remove the previous links or traces, so there must be some appropriate mechanism to fully utilize the potential of requirements traceability. The next sections of my paper will be providing some extensive literature review, thorough research methodology, a hypothesis that will be followed by an empirical study.

1.1 Guidelines for Requirement Traceability:

- The process of RT consists of traceable relationships and they tend to care basically connecting relationships among several requirements. Every team member is associated with specific requirement in the system and they have various roles to fulfill; analysts, architects/designers, development, verification, QA and finally customers who are more interested about proceedings. Therefore, the well-defined and well-connected relationships really helps in taking interested stakeholders onboard in vital decision-making process.
- Tracing in RT should be start in project initial phase with linking of requirements and requirements documents. However, the labeled requirements proceed as successive documents(specifications) those emerges from early requirements. Afterwards, the team members should be well briefed on the benefits of tracing as well as the standards followed

in creating key documents like complete implementation specifications or test plans.

- The real potential of RT is to collaborate and consult the requirements appropriately in the overall system. It allows you to capture decisions and keep that authentic information associated with each requirement. Later, down the road, if we need to revisit these decisions, all data is stored and will be easy to locate. Therefore, we need to connect conversations, data and decisions in a single system in the development process.
- The data is usually stored in several systems; which may further provides visualized coverage of specific trace relationships. In order to minimize risk and ensure quality, we should also consider of automating bi-directional RT.
- RT should be involving in all vital reviews. If any attribute the team decides to include in RT (e.g., lower level specifications, designs, code modules, test cases) should not be considered as accomplished until review has verified that all linked higher-level requirements have been rightly accounted.
- When teams are facing increasing complexity and pressure abide by industry regulations to scrutinize, track and connect inter-dependent requirements. So, we should conduct formal reviews in perspective of industry regulations and internal controls through built-in reporting. These formal reviews aid in achieving market demands and further teams work on traceable requirements.

Table1.1.1 Checklist of features provided by RT (Requirements Traceability) approaches

Requirements Traceability	SS	FR	TR	DM	TP	RM	QE	AR

Forward Traceability	✓	✓	✓	✓	✓		✓	
Backward Traceability	✓	✓	✓			✓		
Bi-directional Traceability			✓	✓	✓	✓	✓	✓

SS: Single system

FR: Formal reviews

TR: Traceable relationships

DM: Decision making

TP: Test plans

RM: Risk Mitigation

QE: Quality Enhancement

AR: Automating requirements

1.2 Motivation:

Traditionally, the requirements have been managed and traced through isolated set of documents. Eventually, it really halts the process of collaborative ways of managing requirements and particularly change management. In order to analyze the true picture of what's been built and if it's on track, it's vital to build connections between data, as well as to map the decisions and conversations associated with each requirement. The process of RT is often

misunderstood and under-utilized in managing requirements. These are the reasons that RT directly relates to change management and we are further analyzing the impact and need for RT in change management.

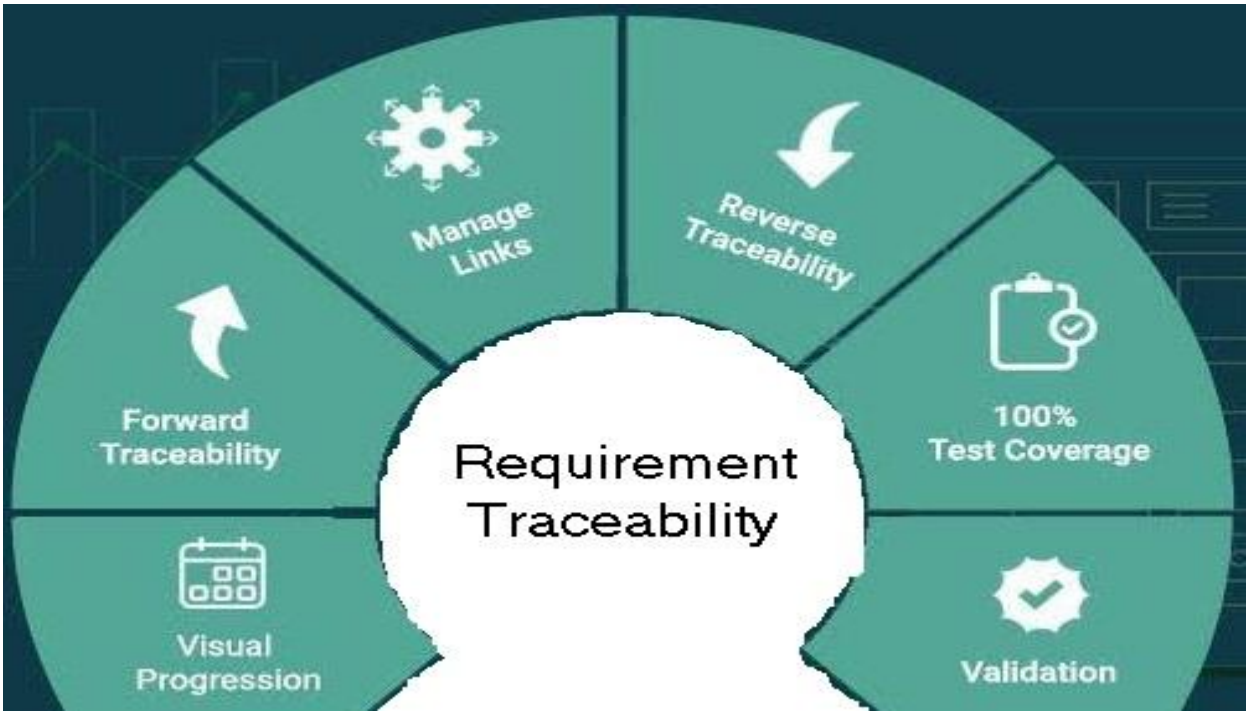


Fig 1.1: Requirement Traceability

1.3 Research Objectives:

- To provide guideline for RT that serves as a benchmark to help and define relationship between RT and change management. Later, it will aids in producing the adequate traceability of requirements.
- Developments of RT metrics through which overlapping requirements can be identify and then they can easily be traceable.

- To use our proposed RT guideline for finding out and attributing reasons for utilizing RT in change management. Further, it will implement change according to the associated doubt or risk related to traceability of the product.

1.4 Problem Statement:

A detailed study of RT and its approaches shows that methodologies used for requirements traceability are much vague and does not comprise of a complete strategy. Major challenge occurs in the software development, when the requirement of a user changes and if the system is unable to recognize it then the existing traces will remain suspicious and halts the process of RT. Further, these traces tend to muddle the process of requirements change management. There must be some traceability mechanism to effectively implement these changes in order to make the requirements traceable and inevitable to accept change in the system. Further, they can be tracked based on change associated to each requirement. It is evident from Figure 1.1 that RT and process of change management are co-related but still we haven't been able to utilize the true potential of RT in change management.

1.5 Proposed Solution:

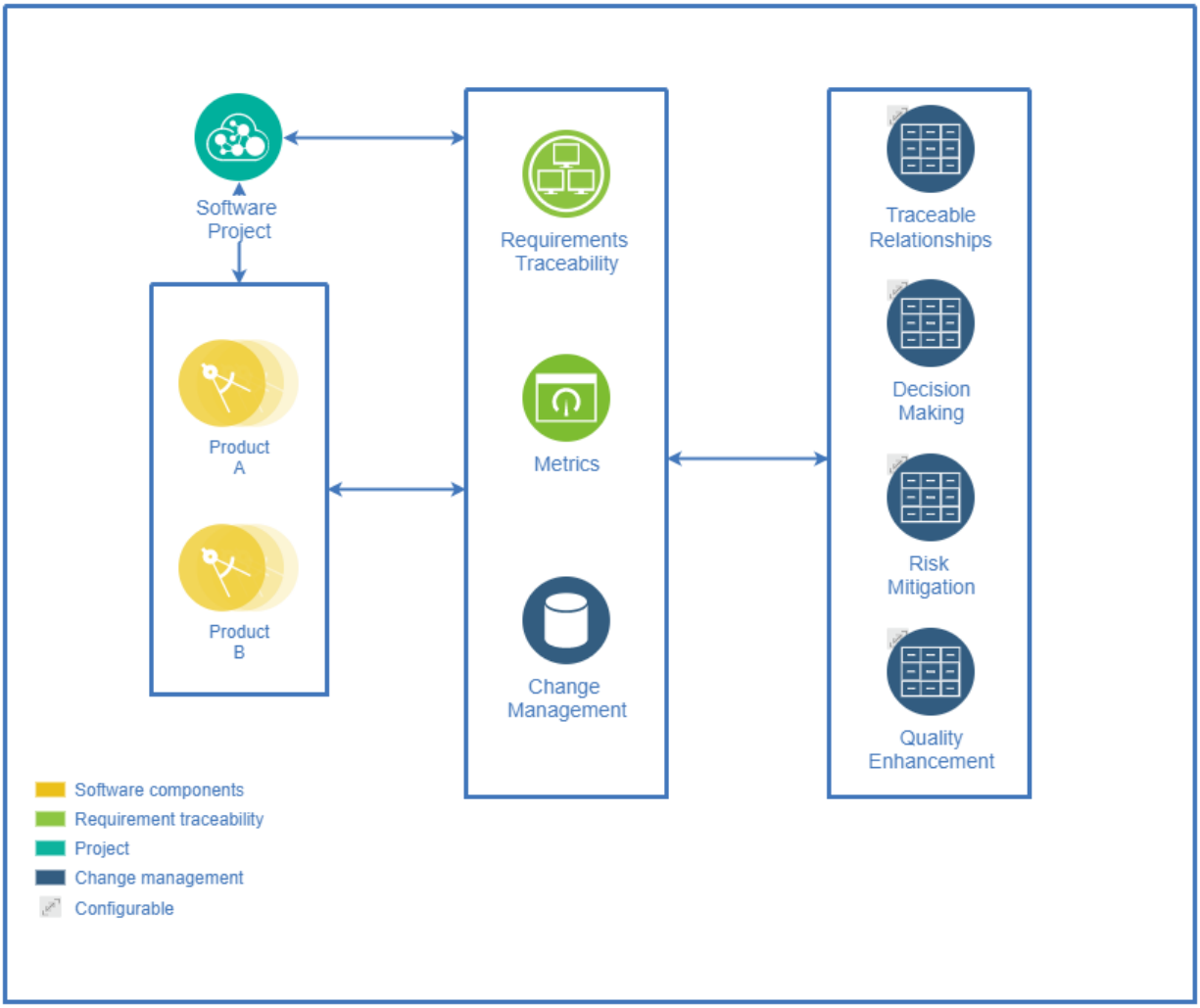


Fig 1.2:Proposed Framework

We present a guideline for RT, as our first contribution, after recognizing RT approaches according to services that they can offer in perspective of change management. The proposed framework consists of various software components shown as product A and B in the project. Afterwards, the traceability metrics behaves as a control mechanism between the processes of requirements traceability and change management. Further, the process of change management becomes more effective after

utilizing the perks provided by RT. On the basis of proposed framework changed requirements will then be tracked according to their associated risk.

1.6 Significance of study:

Nowadays in modern day software development we often heard that an ultra-critical business requirement needs to change. However, it will hold to be in the future release. We really need to know how recommended change will impacts our work regarding the overall change management process. There are several unanswered questions arises when we have such scenarios in software development, particularly in perspective of change management. That's why the aim of this study is to utilize the true potential of requirements traceability and enhancing its role in change management.

However, the process of change management and requirements traceability has its own challenges for co-existence. This research has been carried out to recognize the challenge that affects requirements traceability in CM. In this regard, a guideline for requirements traceability has been proposed in order to enhance our understanding of requirements traceability. Further, the proposed guideline has been utilized to extract features or services provided by various requirements traceability approaches.

When we consider our working downstream and analyze its impact in change management. The term downstream symbolizes the process of requirements tracing both in forward traceability and later in backward traceability. Further, the research explores the various requirement management solutions and its compatibility with the process of requirements traceability. The unwillingness or lack of intent to utilize the perks of requirement traceability in software engineering and particularly in change management is a big problem.

In the context of CM, according to the literature and provided solutions, the role of RT is not clearly defined, and we haven't been able to utilize the true potential of RT in modern day software engineering. Therefore, after identifying the challenges of

requirements traceability, we have proposed appropriate requirement traceability metrics that intends to collaborate the processes of requirements traceability and change management effectively. The proposed RTM provides the missing control mechanism in order to utilize the true potential of RT in software development and CM.

According to[6]estimating the implementation risk of requirements is largely based on gut decisions. It can be disastrous in case of large-scale agile projects, where requirements tend to frequently change every now and there. So, the proposed RTM aids the process of effective decision making and risk mitigation in further implementation of changed requirements.

1.7 Research questions:

1. How the requirements will be effectively traceable?
2. How can management of various traceable requirements will impact the performance?
3. How the proposed mechanism can aids the overall effectiveness of requirements traceability?

1.8 Research Limitations:

The findings of this study should be analyzed within the limits and constraints imposed by the nature of methodologies used. As this study is conducted on selected projects in software industry, we should be cautious in generalizing the findings of this study on various software projects. If we do so, the accuracy and clarity of results had to be compromised because it depends upon the feasibility, scope and technology used in various software projects. It should be acknowledged that both quantitatively and theoretically the validation techniques for the proposed model need further experimentation.

This study investigates the impact of RT in software engineering and further in change management. As far as CM and RT is concerned it may varies depending upon the implementation methods and approaches used in various software projects. Therefore, the collaboration of strategies and utilization of comparative techniques in future exploration would bring a greater perspective.

1.9 Organization of Study:

The first chapter includes the introduction and overview of the research. The chapter of introduction further includes the detail guidelines of RT in software engineering. Moreover, it explains actually what the real motivation behind study was, the significance of this research, objectives, problem statement, proposed solution, research questions and its limitations.

The next chapter of this thesis includes Literature Review. All the second chapter and literature is related to our topic of research "Escalating the role of RT in CM" is studied and cited as well. The chapter includes the discussion on different approaches of RT along with various CM techniques. Further, how these have been utilized in order to tackle change in software industry. Moreover, the literature includes the need for RT and RTM in change management.

The third chapter is entitled as "Research Methodology". The methodology of research is defined in this chapter e.g. which technique is used for conducting our research, hypothesis of our research, development of research, push and pull factors, research design, how analysis is done on the collected data to retrieve results?

The fourth chapter "Proposed Methodology" explains the proposed model for research. The following chapter explains the proposed framework in detail.

Fifth chapter "Result Analysis and discussion" provide all the results that we get after performing analysis. Graphs, figures and all the result details are included this chapter.

Sixth and the last chapter "Conclusion" include the final remarks about the result and further recommendations related to the scope of research.

CHAPTER 2

LITERATURE REVIEW

Literature directly linked to requirements traceability and its impact in change management. We will start from the recent literature particularly from this year. The impact and need for traceability in modern software development has been explored in software engineering practices and its emerging forms. Traceability glorifies project management, CM and aids in the collaboration of individuals from various domains. Moreover, traceability is categorized as converge concern, extends in requirements engineering, change management, development and testing cycle of whole software engineering. In recent time, research papers related to traceability has been continuing to be published at credible international (journals/conferences).

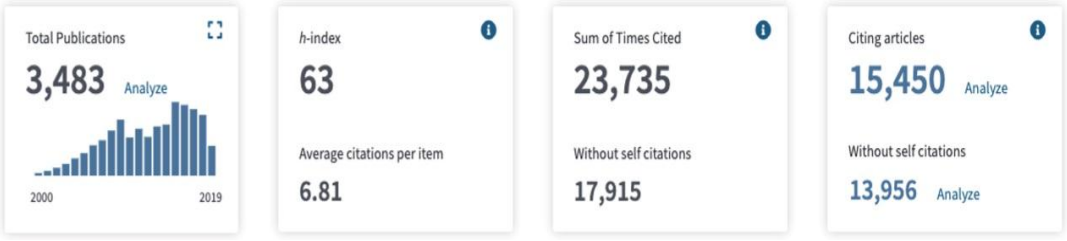


Fig 2.1: Citation report on "traceability" in CS and SE domains

However, this citation report in (Figure 2.1) shows that the amount of papers published related to traceability has been increasing tremendously consistently. The recent[10] developments has shown that software traceability is much pertinent than before and that will need various organizations to implement RT and change management practices in future appropriately. In software engineering, RT and CM remains an important topic in perspective of re-engineering, maintenance and reusability. In perspective of development, if traceability links[11] between source code and requirements are of poor quality, the development effectiveness is considerably decreased. During the process of CM, the software engineers cannot instantly alter the related module code.

In order to follow the process of reusability, we need traceability among various artifacts and requirements. According to author[12] we should consider the reuse of software artifacts and traceability between different artifacts is vital step towards effective computing system that considers higher-level goals and evolving from human activities in software development to various computing activities.

2.1 Requirements Traceability:

In some cases, the process of RT decreases through change management and incremental development. The traceability is also recognized as a paramount activity by several standards like CMMI (Capability Maturity Model Integration) and IEEE (Std. 830- 1998).Nowadays, the requirements are not managed appropriately because of short time in release of software .e.g., the conflict may arises between documents and other vital development activities, if these documents are not continuously updated while adding newer features and deleting existing features. However, this may arises further problems in maintainability of software system and process improvement .The software systems[12] that tends to achieve high-level goals involves great amount of effort from humans or the software products produced are restraint in their variability.

The ability to trace software artifacts through a variety of software products and product line is called traceability. Even though with so many researches, RT[13]still

remains the most problematic area. The problem of RT[14] identifies the inability to track the life of a requirement from its origin and how it impacts other software artifacts. The process of consistently validating that a service offered by product should meet system specifications involves RT. In order fulfill the intended purpose of product, so adoption of right traceability technique becomes vital. Further, we have discussed various traceability techniques in RT:

2.1.1 Forward Traceability:

The forward traceability checks whether the project progresses in the right direction and for the desired product. FT identifies the requirements that can impact in future. After the changes are made FT really helps the process of traceability. When the requirements are traced, it will aid the developers to convert the requirements into blocks that specify the particular module of the system based on that block requirement.

2.1.2 Backward Traceability:

The backward traceability initiates from low level requirements. It aids the process of RT to analyze which part of system module impacts the product based on each requirement.BT is really useful in identifying the defects for low level requirements. Therefore, changes can be done for high level requirements accordingly.

2.1.3 Bi-Directional Traceability:

The Bi-Directional traceability is implemented both in forward and backward direction. This approach for traceability gives complete over view of requirements e.g. it identifies sources of requirement to the end product and from the end product to sources of requirement).The traceability can be enhanced in forward and backward direction by utilizing the mentioned approach in Figure 2.2.After the requirements[15] are managed well traceability can be improved further from the sources of requirement to its lower level abstractions and from their lower level abstractions back to their sources of requirement.

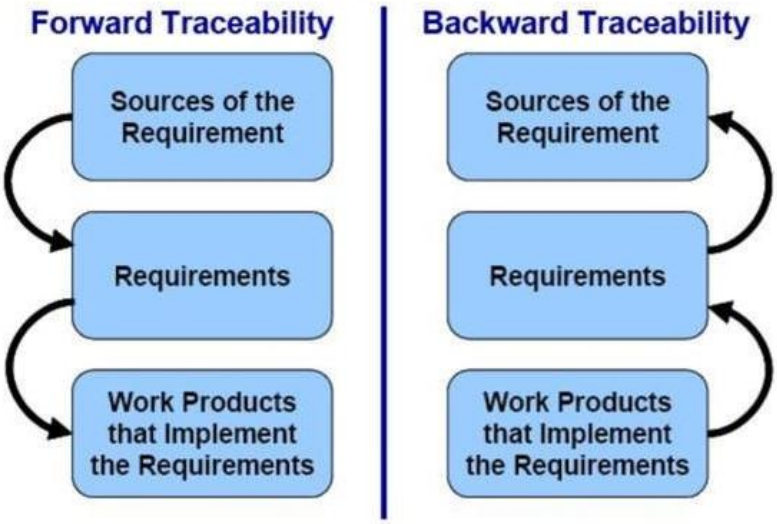


Fig 2.2: Bidirectional Traceability

2.1.4 Horizontal Traceability:

The horizontal traceability is an aspect or characteristic of identifying non-hierarchical interactions, mutual properties, similarities, etc. among work products and requirements. The links between requirements and other artifacts are established by horizontal traceability also categorized as extra-requirements traceability.

2.1.5 Vertical Traceability:

The vertical traceability is an aspect of identifying the source of requirements basically from requirements to design, development and to test cases. It describes[16] the relationship between products which build upon each other or derived from each other, e.g., from user requirements to acceptance test cases, between abstraction of requirements, among requirements and project plans. It further describes the relationships between requirements. Vertical traceability is really paramount to understand the requirements evolution and the impact of this requirements evolution in change management.

Researchers in[2]have proposed about the evaluation of requirements traceability and its usefulness considering the phase of software maintenance. The main challenge that has been addressed in this paper is to investigate that how important or crucial is to justify the cost of requirements traceability in significant support of development tasks. For that purpose a relatively controlled experiment is carried out on 71 subjects in perspective of maintenance and development of the software projects. The task was not assigned specifically to various subjects; half of the tasks were performed with traceability and the rest without the traceability of requirements.

Further, the related progress of every subject is measured. The performance is measured on the basis of time taken to solve a specific task. After selecting the maintenance tasks as specific benchmark or criteria, it can be much clearer that how their real developers solves the specific tasks frequently. This

will allow researcher to analyze the results and the benefits provided by requirement traceability in better perspective. The process of traceability provides several perks, like process improvement, sequence between requirements and evaluating development. However, it is very vital to justify requirements compliance and ultimately mitigation of risk involved (in meeting of customer or user expectations).

In software engineering, the impact of traceability is still considered as a paramount aspect to rediscover. So, it really requires considerable research and technical efforts in order to solve its hurdles. Researchers in [4] have proposed the future directions and trends of software traceability. By analyzing the prior knowledge and body of work on requirements traceability, they have been able to identify the compelling areas of research successfully. According to Mirakhorli [17]the major challenge in tracing of architectural decisions is to recognize the moreover appropriate views which aid our tactics and maintain VT. The main problem of traceability in practice is that, it is implemented temporarily or at ad-hoc basis. It is one of the vital reasons why we have not been able to fully utilize the perks of requirements traceability.

In the recent past, the researchers and authors have really recognized some vivid areas or problems of the traceability, promotion of strategic planning and developing more sophisticated tooling, implementing various techniques good enough for the trace creation and optimizing the whole process. According to author [18] the traceability researchers(industrial experts) collaborated to identify the challenges in making traceability ubiquitous and consistent. It also focuses on the development of new query languages and approaches that utilize trace links more effectively. Further, it implements traceability in particular domains like product line systems, agile project environments and Model Driven Development.

2.2 Requirement Traceability Metrics:

In recent past, the metrics are implemented to achieve comparative values of specific quality features in software. Further, these software metrics helps to measure current performance. Researchers in [14] have deduced some long lasting credible terms from empirical study of software metrics. After its extensive study, the base metrics are categorized into four clusters: process, product, project and organizational. The main challenge addressed by authors is the incompetence of requirement engineer to recognize the origin of their requirements and later on prioritizes the requirements to mitigate the risk.

Now days, the metrics related to product are often utilized in IT Industry. Recently, the empirical study was carried out in various software organizations. The main objective of this empirical study is the evaluation of constantly used metrics in the software organizations. Researchers in[6]have proposed metrics to quantify and assess relations of requirements. The study have been conducted on several industrial (agile-based projects). They have been able to found that the recommended metrics are more likely acceptable for appraise the implementation risk for requirement.

The main challenge addressed in this paper is that nowadays gut decisions are largely used in calculating the implementation risk for requirements. It can be disastrous in case of large scale agile projects, where requirements tend to frequently change every now and there. In Agile developments, it follows frequentative process with requirement management and phases of implementation sprints or categorized into modules. The appropriate test measures are considered for each sprint and changed requirements are selected from the product backlog. The results of these RT metrics provide justification for implementation risks in requirements. This justification aidsin product developers test measure selection and ultimately in requirement prioritization.

Researchers in[19] have proposed a methodology that uses a traceable metric to measure software artifacts from the phase of requirements to source code. The main challenge addressed in this paper is there are no metrics that can exhibit the quality (and the evolution) of the same software artifact in all development phases. However, the measurements are only implemented and tend to restrict in one specific development phase. Specifically, the results validate the approach on transactions defined in use cases or documentation and implemented in source code. Further, some initial

evaluation has shown that the recommended approach is promising to solve the problem of tracing software artifacts in the development process.

Researchers[20]have proposed some generalized traceability completeness measures. These traceability completeness measures further explain the role of RT completeness on the quality of software. The main challenge identified in this paper is about the evidence of expected benefits from the process of traceability. In this regard, the focused have been kept on supporting activities of requirements that aids in utilizing the process of traceability. It has been recognized[21] in recent study that traceability plays very important role in success and failure of various projects. Moreover, it also have immense effect on the quality of software.

According to [22] if the core developers identifies traces and the quality of these traces is good in its initial phase. However, As our system evolves the quality of these traces deteriorate The results have shown that degree of traceability completeness is co-related to defect rate and ultimately the quality. The proposed a solution to counter gray links. Further, it discusses the concept of value-based RT in detail.

Author [23] presents peculiar traceability framework for large scale and safety critical systems and projects. The proposed mechanism TORUS introduces splices. These splices involve the graph-based data structures that manage traces automatically. Moreover; it creates trace links among various requirements by considering inevitable change that occurs during the development process. According to [7] the TRL provides appropriate abstractions to requirements, trace links and various artifacts. Further, TL can be searched, filtered and retrieved .

Researchers in [8] have proposed a model in order to assess the defect density indicator of software in initial phases of development utilizing fuzzy logic and reliability metrics .The main challenge addressed in this paper is that ambiguous data is often not achievable in early stages of software development for reliability analysis. Although, theoretical value of software metric is also recognized in the early stages which have played pivotal role in reliability analysis. The model proposed is implemented on 20 real software projects and it is observed that the value of defect density indicator is greater in requirement analysis phase than that of the development and implementation phases. However, the framework is validated from the existing literature and these validation results are adequate.

2.3 Change Management:

In spite of growing product complexity and intense regulations, the majority of software organizations and teams don't have a standardize requirements management solution in first place. Mostly half of them use a tool that's not purpose-built[24] for the management of requirements, while almost one third have no specific system in place at all, relying heavily on shared documents and e-mail. According to recent literature and guidelines provided for RT in first chapter justifies that highly regulated industries utilizes an agile approach for change management.

The agile approach will dig deeper into how stakeholders and customers in various industries can benefit from a collaborative platform[12]throughout complex development cycles that ultimately helps them to manage requirements effectively. In large scale systems[25] when a feature or function has changed, it is really challenging to assume which test scenarios were assessing those specific parts of code. It is known as test-to-code traceability. However, it shows that very precise results can be achieved through good coding practices and naming conventions can ease the task. As the complexity increases, it shows that they are no solution for various traceability problems.

According to [26]only 15% of teams had invested in a purely RM and CM solution but unfortunately that number is too low because organizations doesn't understand the ever-lasting value and perks provided by RT. It plays vital role in understanding the relationship between requirements and how the requirements have been consistently evolving. The variability information is not often explicitly represented, which leads[27] organizations to the ad-hoc change management. Due to orthodox RM, we have observed that more than three out of four design teams have experienced product failures.

In short, poor management of requirements leads to product failures. The two most important reasons for the failure of product includes the exceeding cost and time due to poor RM and CM. Further, it also showed that organizations[28]utilizing dedicated RM platforms in regulated industries receives fewer instances of recalls, warnings, production stoppages and fines than those that didn't uses these dedicated platforms. However, nearly half teams reported that they are not experiencing these problems at all.

The other important unaddressed issue is missing of requirements[29]in perspective of CM. Nowadays, as the complexity of product increases; teams without comprehensive CM solutions observed that product don't meet all requirements. There are various reasons why teams report that their products were shipped with missing key features and requirements, many of which can be linked to increased demand for embedded software, the need to adopt different changes and product complexity.

More importantly, shipping of product that doesn't meet requirements can lead to scrutiny from regulatory agency, which can permanently damage your position in the market and brand reputation. Systems are getting more complex and it justifies the need for the appropriate requirements solution. The researchers in[30] have observed that if we have a runtime failure and system is unable to meet various expectations like requirement management and quality of service. Then infrastructure of traceability can be utilized to understand which quality attributes are not satisfied. Ultimately, it concludes that more complexity means more time will be spent in tracking of requirements. According to Andrea[31] the consistency is the most paramount issue of traceability management, CM is directly related to the evolution of TL.

According to[29]recent report, the respondents those have to increase the complexity of their product due to change in requirements acknowledged that they are spending excessive time in tracking requirements. It all occurs due to poor management of requirements and inabilities to utilize the appropriate techniques of traceability in CM. However, the organizations are still reluctant, investing in the tools that would aid teams to manage the requirements considering recent complexity demands. In short, RT dives deeper into the relationship between

increasing product complexity (change) and effective change management. According to [32] GSD (Global Software Development)are usually related to collaboration and communication. It also identifies issues arises from cultural differences between teams.

According to author[33] traceability, change management and documentation are the major activities of requirement management process. It has been identified that traceability and CM are co-related. The most paramount and effective way of managing requirements is to assure requirements traceability from very start of system evolution and maintenance. More importantly, RT is a modern day need for a system where a change occurs randomly. The tools for traceability and requirements management can be effective in scenarios when requirements changed constantly. Further, the impact of this change will affect other artifacts of the system. There are several [34] long lasting maintenance projects in which impact analysis and RT and impact analysis are vital factors for reducing time and cost in the ALM.

Researchers in [5] have proposed a framework known as Eiffel. It is developed by the assistance of Ericson. It is designed in order to supply real time traceability in frequent delivery and integration. The major challenge that has been addressed in the paper is the importance of traceability. It recognizes traceability as a main hurdle for attaining persistent delivery and integration. Further, it also creates barriers in documentation of detailed industry developed framework. Results of the paper show that suggested framework have produced the desired results or productivity, when compared to previous traceability approaches or techniques.

Researchers in [1] have proposed metrics and its usage to enhance the requirement management process. The paramount challenge addressed in this paper is the impact of undefined or ambiguous requirements on the requirement management process. The author tends to highlight what are the main reasons in software projects that fail the process of requirements management. Therefore, requirement traceability plays really important role and requirements are the basic building block in every project planning and development. It is quite hectic to develop an appropriate system without the clarity and clearly stated requirements.

it is very hectic to develop an appropriate system. Internationally, the success rate of completed projects is below-average because of ambiguous requirements and fuzzy management process. In software projects, requirements play very critical role in successful management of projects and particularly change management.

Researchers have proposed the quality framework that identifies each attribute its (Traceability Gate) desirable state and undesirable's deviation or variance from this particular state. The undesirable variance is the most important traceability problem. However, they also formally describe that how both undesirable states or the desirable variation can be discover in order to assists developers and other stakeholders. Ultimately, it assesses the traceability of their project systematically. The main challenge addressed in this paper is that subsist traces often are of suspicious quality. However, these traces further lay the foundation for paramount or high impact commitments in perspective of development and change management. It was founded in recent study; the practitioners were unable to assess the quality of traces because they don't follow the appropriate guideline for RT. The author[35] proposed tool (TestAlgo) that generates test cases automatically. The tool really saves cost and time for testing phase. Further, it efficiently compensates the process of change management and traceability.

It further establishes for traceability that every traceability approach and problem should be given specific weightage in order to understand its sensitivity for the specific product requirements. However, these results should be utilized to quantify the impact of traceability and further prioritize the evaluation of traceability features. In order to enhance[36] the role of requirements traceability successfully forum and bug tracking techniques are used. The TL identifies[37] the impact of change in detail when various requirements are altered, deleted or changed. According to Cleland[38] TLE (Trace Link Evolver) automate requirements and trace links as per the frequent changes in the large scale project.

Several researchers around the world[39] have worked on optimization of the RM process. Managing of requirements is not an easy activity; in order to record change in requirements and manage the suitable requirements afterwards is the most ambiguous task. The extension of the RM system is based on the

optimization model in perspective of the design process. Further, we have analyzed some vital journal and conference papers regarding traceability metrics and change management.

Table 2.1: Analysis of Literature

Paper	Contributions	Drawbacks
[4]	✓ It resolves problem of traceability, while implementing continuous integration. ✓ Proposed a framework known as Eiffel. It is developed by the assistance of Ericson. ✓ It is designed for implementing real time traceability in parallel delivery and integration.	• Proposed model is hard to implement where different components and systems are configured together because every requirement has its own understanding of the system • They may overlap each other in some situations. • It doesn't explain the utilization of distributed data-bases to track down requirements.
[2]	✓ It resolves the issue of requirements traceability by investigating that how important or crucial is to justify the cost for RT. ✓ Results have shown that requirements traceability increases the accuracy of a project by 50% the speed of a development project by 24%.	• In the proposed model all the evaluation carried out for justifying the impact of RT in perspective of maintenance of software is kind of one dimensional . • However to justify the cost of RT in significant support of development tasks, we need some metrics to justify its cost.

[1]	✓ The paramount challenge resolve in this paper is the impact of undefined or ambiguous requirements on the requirement management process. ✓ The problem is resolved by the proposed metrics and its usage to enhance the requirement management process. ✓ It highlights that what are the main reasons in software projects for its failure in perspective of RM.	• The proposed model is hard to implement when requirements are continuously changing or when we have ambiguous and unclear requirements. • The proposed traceability metrics haven't been implemented accurately in order to create appropriate links between the defined relationships and dependencies.
[6]	✓ The issue addressed in this paper is that the traces are basically of suspicious quality. ✓ The practitioners were unable to assess the quality of traces because they don't follow the appropriate guideline for RT. ✓ It further establishes for traceability that every traceability approach and problem should be given specific weightage in order to understand its sensitivity ✓ The results should be utilized to	• Since all the quality models are based upon the user defined requirement if in case user defines the requirement wrongly it may result in a huge damage especially for critical systems. • Secondly, if the requirement of a user changes and it the system is unable to recognize it timely, then the existing traces will remain suspicious as well as the overall quality of the system. • Thirdly every time the requirements modified it will have to maintain the origin of the requirements by the

	quantify the impact of traceability and further prioritize the evaluation of traceability.	assistance of some requirements traceability metrics.
[14]	✓ The issues resolved are the incompetence of requirement engineer to recognize the origin of their requirements. ✓ Further, the process enhances to prioritize the requirements in order to mitigate the risk.	• The empirical studies have brought out several unaddressed issues but the most important one is the implementation and understanding of requirements traceability metrics. • Further, it doesn't explain the utilization of requirements traceability metrics to track down the requirements on the basis of risk associated.

[6]	✓ The gut decisions are largely used in calculating the implementation risk for requirements. ✓ They have proposed metrics to quantify and assess relations of requirements. ✓ The proposed RTM aids the process of effective decision making and risk mitigation in further implementation of changed requirements.	• It can be disastrous in case of large scale agile projects. • In Agile developments, it follows frequentative process with requirement management and phases of implementation categorized into several iterations. • The indicator is not good enough to support developers/stakeholders in requirement prioritization and decision making.
[8]	✓ The main challenge addressed in this paper is that ambiguous data is often not achievable in early stages of software development for reliability analysis. ✓ It is observed that the value of defect density indicator is greater in requirement analysis phase than that of the development and implementation phases.	• The framework is hard to implement because fuzzy logic mostly identifies partial truth. • Reliability analysis can be enhanced through software metrics by implementing the requirements traceability metrics regularly.
[20]	✓ The issue addressed by author is that the benefits from	• It is really hard to declare because the focused have been kept only on supporting activities of requirements.

	traceability are achievable. ✓ Further, it has shown that degree of traceability completeness is co-related to defect rate and ultimately the quality.	• The traceability is a continuous process and in order to achieve quality we have to focus on every requirement and supporting activity.

From the existing literature [6][8] it is quite clear that use of requirements traceability metrics in software development is the upcoming need. In Agile development, the indicator mentioned is not good enough to support developers and owners in requirement prioritization. Reliability analysis can be enhanced through software metrics by implementing the requirements traceability metrics regularly. After[20] the subsist traces often are of suspicious quality. However, these traces further lay the foundation for paramount or high impact commitments in perspective of development and change management. These traceability completeness measures further explain the role of RT completeness on the quality of software.

Mostly the decisions taken in Agile development are based on gut feeling, it's quite risky in perspective of change management. So, we must utilize the perks of the requirements traceability process for better understanding of system and later on it will help us in how to implement the change in the system effectively. If the requirement of a user changes and it the system is unable to recognize it timely, then the existing traces will remain suspicious as well as the overall quality of the system.

The results have shown in [2] that the requirements traceability increases the accuracy of a project by 50% the speed of a development project by 24%.But it is not enough to justify the cost of requirements traceability in significant support of development tasks, we need some metrics to justify its cost. In [14] [22] it doesn't explain the utilization of requirements traceability metrics to track down the requirements on the basis of risk associated. Every time the requirements modified it

will have to maintain the origin of the requirements by the assistance of some requirements traceability metrics.

It has shown that degree of traceability completeness is co-related to defect rate and ultimately the quality. In [1], [33] and [39] when requirements are continuously changing or when we have ambiguous and unclear requirements. The proposed traceability metrics haven't been implemented accurately in order to create appropriate links between the defined relationships and dependencies. Different components and systems are configured together because every requirement has its own understanding of the system and they may overlap each other in some situations.

CHAPTER 3

RESEARCH METHODOLOGY

It is indicated by the title; this chapter explains the methodology of our research. After, the reviewing of literature in various perspectives. This part particularly provides the detail of our research strategy. Further, the research model, hypothesis and data analysis techniques have been identified. This research is purely conducted to validate our research model.

3.1 Research Model:

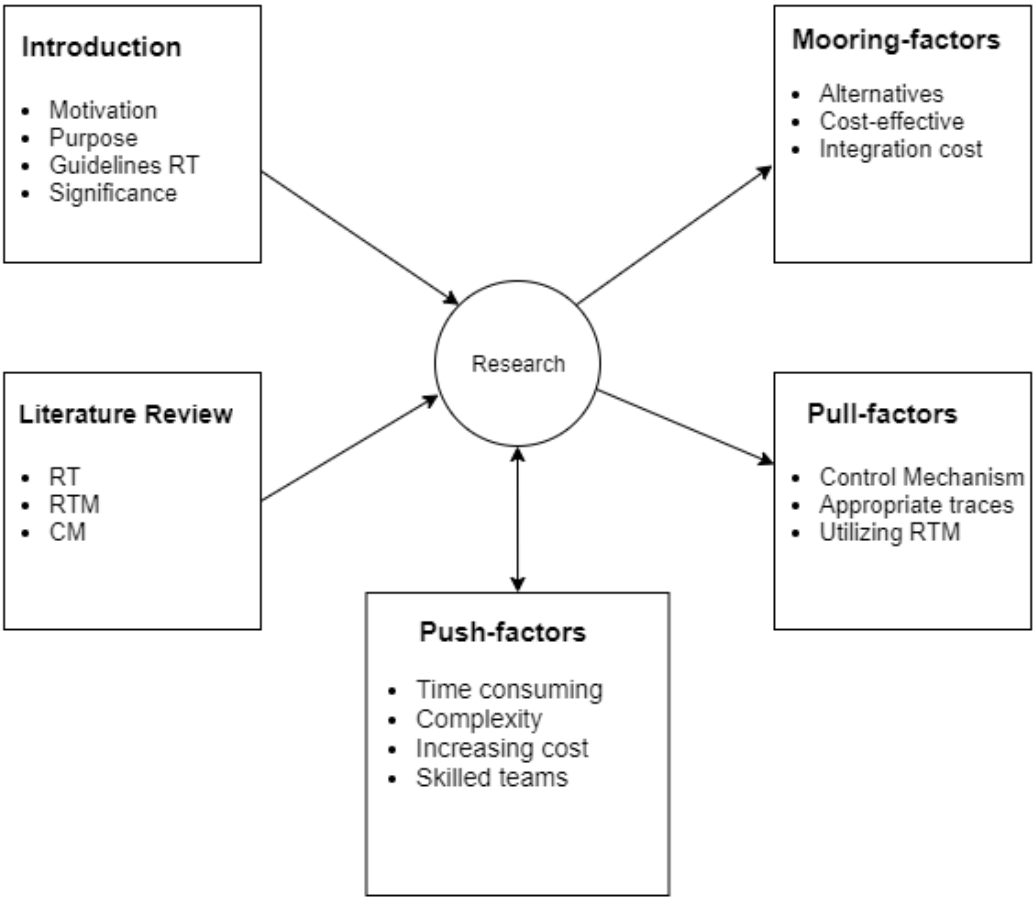


Fig 3.1: Research Model

The research model consists of push factors, pull factors and mooring factors. After reviewing of the literature, it identifies several push factors which really gives the true direction to our research. Push factors are the negative factors that tend to define our research in perspective of RT in change management. The main push factors in order to enhance the role of requirements traceability in change management are cost and time. However; the other two challenges are complexity of the system and number of skilled teams related to product development and process improvement. In the first chapter, the guidelines provided for requirements traceability tends to emphasize the impact of modern

features provided by various RT approaches. These guidelines and extracted features further helped us to identify the push factors for our research.

While reviewing the literature, it identifies various perspectives of our research on bigger canvas. After recognizing the push factors, we have been able to recognize some pull factors of our research. Pull factors are the factors that motivate us to carry on with our study. We have been able to extract the factors based on recent literature, industrial complexity and demand. The main pull factor is the appropriate utilization of requirements traceability and traceability metrics. Moreover, the type of our research is exploratory, and we will be utilizing both primary and secondary data for the better understanding.

The most important challenge is to develop RTM (Requirements Traceability Metrics) while considering the whole requirement management and ultimately the change management processes. We have studied recent literature on traceability metrics and various control mechanisms that tends to enhance the role of requirements traceability in process improvement and ultimately product improvement. But there are still gray areas in determining the quality of traces and linking of appropriate traces. That's why it is one of the important pull factors of our research.

After identifying the push and pull factors for our research, we have been able to recognize the mooring factors of our research. Mooring factors basically tend to anchors between push and pull factors. The main mooring factors in order to utilize true potential of requirements traceability are appropriate alternatives, cost-effective implementation and integration cost. The identified factors and research model intends to enhance the quality of our research in greater perspective.

3.2 Hypothesis:

The research model fully depicts the under-utilized role of requirements traceability in change management. The software systems and the products developed don't meet the quality standards due to high-costs of modern day management solutions. As the complexity in system increases, the practices followed by software products haven't been able to tackle the changed requirements appropriately. It implies that process of requirements traceability and change management is continuous and are co-related. Further, the costs of project and time duration are inversely proportional to change management solutions. Ultimately; it halts the utilization of requirements traceability process in change management.

3.3 Data Analysis Techniques:

Atlassian Jira (Version 7.3.0) is used for data analysis and Jira is project management software to assist several teams to complete their related tasks. It is advanced level project management tool. The software particularly suits the nature and scope of our research.

CHAPTER 4

PROPOSED METHODOLOGY

As indicated in the title, this chapter explains the proposed methodology of our research. In previous chapters, it has been recognized that effective utilization of requirement traceability in change management is possible particularly from the pull factors identified in our research model. This chapter provides the detail about the proposed methodology of our research. Further, the proposed framework for our research is explained in detail.

4.1 Proposed Methodology:

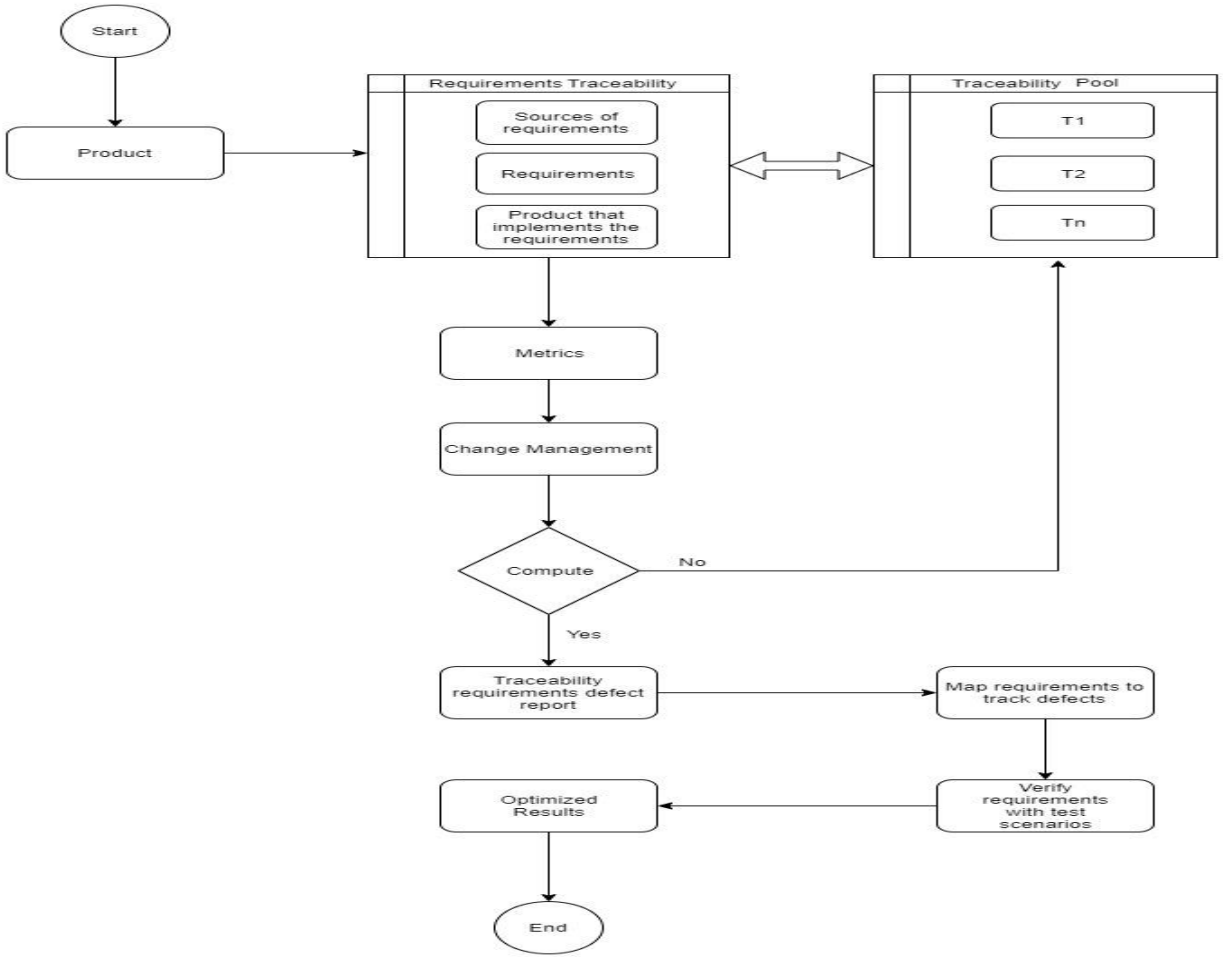


Fig 4.1: Proposed Methodology

The proposed methodology identifies the relation between all responses that can be generated during requirements traceability and traceability pool. In order to analyze the process of requirements traceability, we have to consider the origin of requirements and what were the sources of requirements. The most paramount concern is to follow the life of requirements and how they have evolved. Afterwards, it impacts the functionality of the product and over-all system quality.

After going through literature, we have been able to identify various approaches of traceability for effective utilization of requirements traceability in management of requirements and ultimately in the whole process of change management. So, the process of requirements traceability basically has to trade-off between the various traceability approaches in order to tackle the change and process of change management. Further, it will aid in the appropriate development and implementation of metrics in perspective of change management.

Table 4.1: Requirement Traceability techniques

Requirement Traceability	**Weightage**
FT	3
BT	4
BDT	1
HT	5
VT	2

FT: Forward Traceability

BT: Backward Traceability

BDT: Bi-Directional Traceability

HT: Horizontal Traceability

VT: Vertical Traceability

In table above, the various techniques for RT has been given weightage on the basis of their effective utilization in perspective of product development. The highest weightage is given to BDT (1) because it really covers the short-comings of FT and BT. The lowest weightage is given to HT (5) because in other words it is considered as extra-requirements traceability which will be widely covered by following other four

techniques simultaneously. The below Fig.4.2 explains the linking of RT traces. Previously, we have observed that most of the problems are created by the linking of the requirements traceability traces. So, the above mentioned weightage for several traceability techniques will aid the suspicious quality of RT traces.

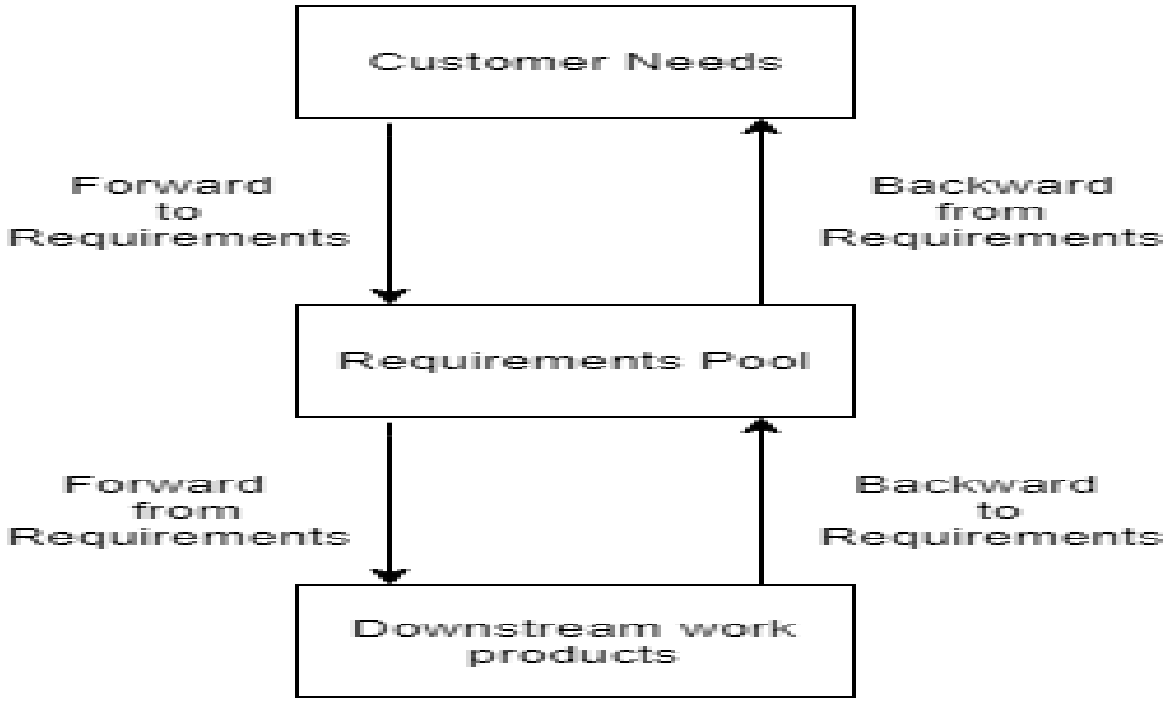


Fig 4.2: Description for requirements traceability links

It is evident from the above figure that the effective utilization of requirements traceability in its linking process largely depends upon the quality of traces. The quality of these links have been improved after implementing the appropriate measures provided by various traceability techniques as mentioned in (Table 4.1).The four links identified are really important in order to understand the process of traceability and ultimately change management. In the figure the link (forward to requirements) aids in understanding the customer needs and technical assumptions in order to linked them to appropriate requirements. Afterwards, if there is any change in customer requirements, then this traceability link is utilized for impact analysis. The link (forward from

requirements) identifies how one or multiple requirements have been assigned to several system components. Similarly, this type of link is really effective in evaluating the impact of requirements change. The rest of the two links aids in verifying and validating of requirements according to technical assumptions and customer needs.

The guidelines provided for RT contributes a lot in better understanding of requirements traceability. The various features provided by requirements traceability doesn't comprises of full strategy. So, the mentioned features in (Table 1.1) have to be utilized in perspective of change management. Therefore, the process of change management basically has to trade-off between the several requirements traceability activities which further enhances the role of RT in change management. The number of activities has been short listed after reviewing literature and analyzing related work from various authors.

Table 4.2: Tasks carried out in RT

Activities	**Weightage**
FR	6
TR	1
DM	2
TP	4
RM	3
QE	5

FR: Formal reviews

TR: Traceable relationships

DM: Decision making

TP: Test plans

RM: Risk Mitigation

QE: Quality Enhancement

In table above, the various activities provided by requirements traceability have been given weightage based on their effective utilization in perspective of change management. The highest weightage is given to TR (1) because it really impacts the overall quality of the product and gives the in-depth understanding of how the requirements have been keep on evolving in the project. The lowest weightage is given to FR(6) because in other words it is considered as unorthodox while having modern day change management solutions.

4.1.1 Requirement Traceability Metrics:

Table 4.3: Metrics to enhance the role of RT

Serial Number	**Metrics**	**Traceability Utilized**	**Prioritization**	**Responsibilities**
RTM-1	Traceable Relationships	TU- I, II	PR- IV	It enables us to track the origin of the requirements in over-all project.
RTM-2	Decision making	TU- III	PR- V	It tends to take appropriate decisions on the basis of traceable relationships
RTM-3	Test plans	TU-V	PR- III	It is responsible for appropriate mapping of requirements through testing cycle
RTM-4	Risk Mitigation	TU- III, V	PR- II	It enables to mitigate the risk

				factor involves in project development
RTM-5	Quality Enhancement	TU- IV, V	PR- I	It tends to enhance the quality of project and management.
RTM-6	Formal reviews	TU- IV	PR- IV	It tends to review the progress of team and their productivity.
RTM-7	Standard Deviation	TU-III,V	PR-II,V	It enables to recognize the variance or change in the project
RTM-8	Issues Resolved	TU- III	PR- III	It resolves the issues because of their timely detection in testing cycle.
RTM-9	Average Resolution Time	TU-IV,V	PR-III,IV	It is responsible to over-view the resolution time which further aids in prediction.
RTM-10	Bugs Per Sprint	TU- III	PR-II,III	It enables us to analyze the product complexity.
RTM-11	Time Duration	TU-III,V	PR- II, V	It is responsible to ease the progress of the project.
RTM-12	Process Optimization	TU- V	PR- IV, V	It is responsible to optimize the process of traceability in every sprint.

Table 4.4: Utilization of traceability techniques

Traceability Utilized	**Utilization Number**
FT	I
BT	II
BDT	III
HT	IV
VT	V

Table 4.5: Services provided by process of RT

Prioritization	**Prioritization Number**
Process Optimization	I
Project Decisions	II
Development Quality	III
Team Collaborations	IV
Business Value	V

Based on proposed measures changed requirements will then be tracked according to their associated risk appropriately. All the un-changed requirements generated at the same time will be executed in next iteration through backward traceability and calculating the risk that is associated to this requirement. Afterwards, it will generate requirements traceability defect report which map requirements to track

defects. Further, it verifies requirements through test scenarios which enhance the process of change management by utilizing the perks provided by RT.

4.1.2 Test scenarios:

The process of requirements traceability can be very helpful in building of appropriate test cases and test scenarios. The phase of testing is really interesting phase to work on, even though many testers get confused in common testing terminologies. However, there is a quite difference between test scenario and test case. Test case identifies the process of (how to be tested), while test case identifies the process of (what to be tested). The proposed control mechanism and methodology have really recognized these unaddressed issues in testing phase by the effective utilization of requirements traceability process. When the change management process is carried out the effective management of changed requirements becomes really important. Therefore, In the testing phase these test scenarios are developed to verify appropriate mapping of requirements.

4.2 STLC:

STLC stands for (Software Testing Life Cycle).It is well-planned testing process[40] which tends to validate various paramount activities in development of the product. However, the various phases of STLC deal with verification and detecting of bugs. As shown in the proposed methodology, the effective utilization of requirement traceability really aids in tracking of defects. Ultimately, it improves the quality of product by effective tracking of defects related to appropriate requirements. It is possible due to previous good work of requirements traceability in managing of

requirements. Following are the stages of STLC, although different organizations implement these stages differently depending upon the complexity and scope of the project.

- Planning of test process.
- Development of test.
- The environment established for test process.
- Test execution

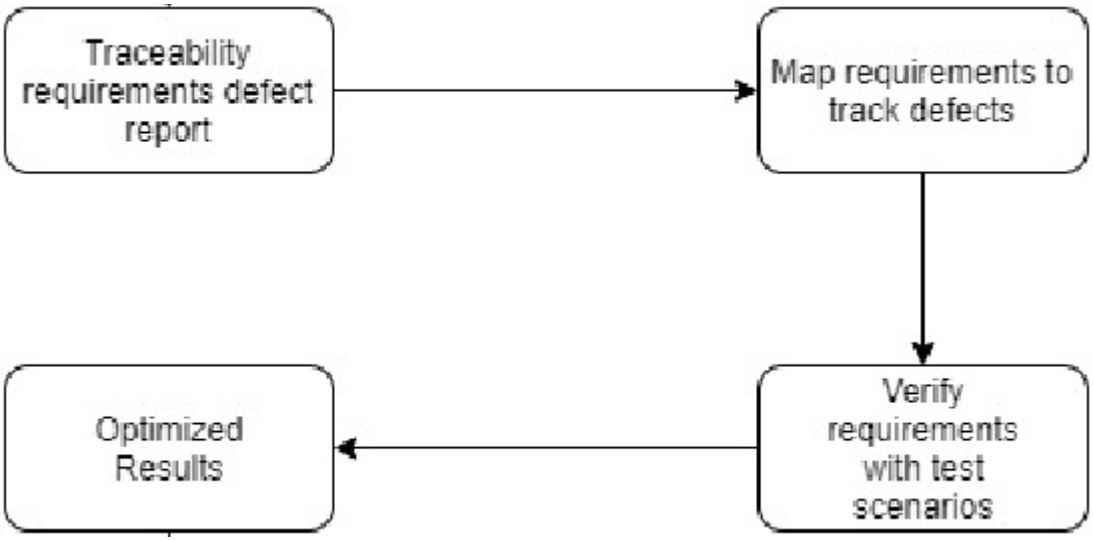


Fig 4.3: Testing phase:

As shown in the proposed methodology after utilizing the perks provided by requirement traceability it can save a lot of cost and time related to testing phase as discussed in STLC. The testing scenarios identifies where the actual test should be executed.

4.3 Agile Development:

Agile development [41] is modern way of compensating project management. The agile manifesto was presented in 2001 by various researchers and industrial experts. In the last two decades agile has been consistent and adaptable as far as complexity is concerned. Agile software development consists of several methodologies. Every methodology [42] has its own perks like Scrum, Extreme Programming, Crystal and Kanban. In all these agile methodologies the most vital or common attribute are their sprints and iterations.

However, it is quite difficult to implement traceability in agile development [43] because the project is developed at very rapid pace. That's why we haven't been clear to implement traceability at start or end of the project in agile software development and Jira. So, we tend to recognize the perks provided by requirements traceability moreover in large scale projects. Therefore, the Kanban provides these privileges for continuous improvements in processes and standards to enhance the efficiency of project teams.

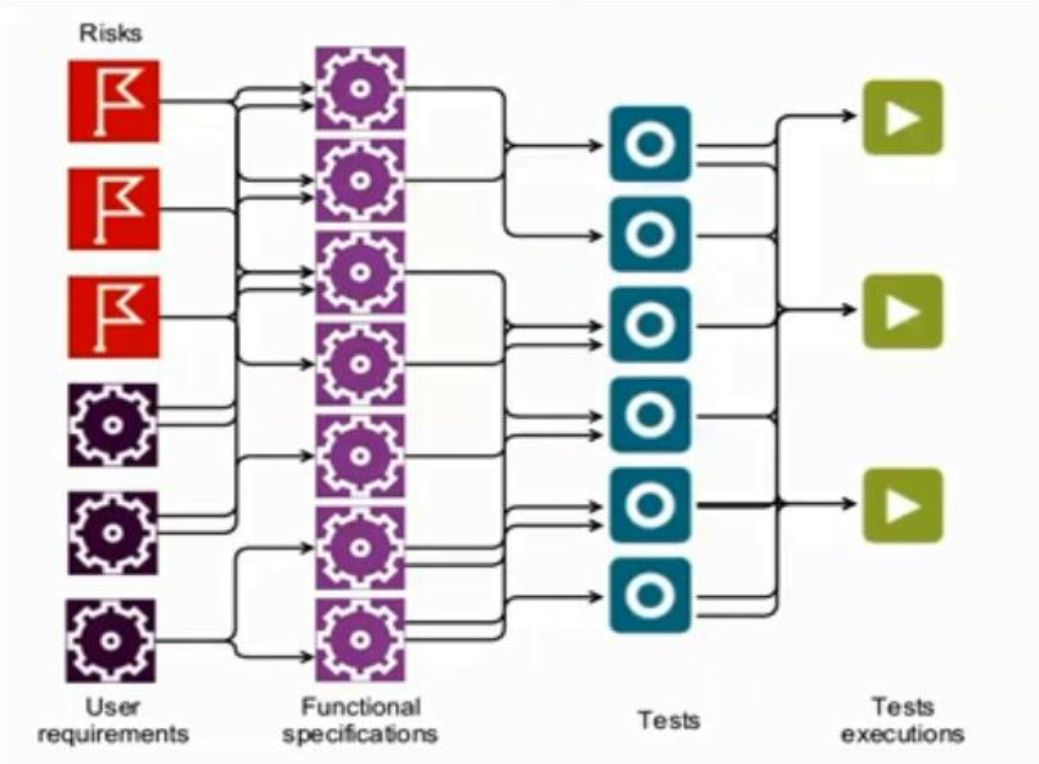


Fig 4.4: Traceability in Agile Development

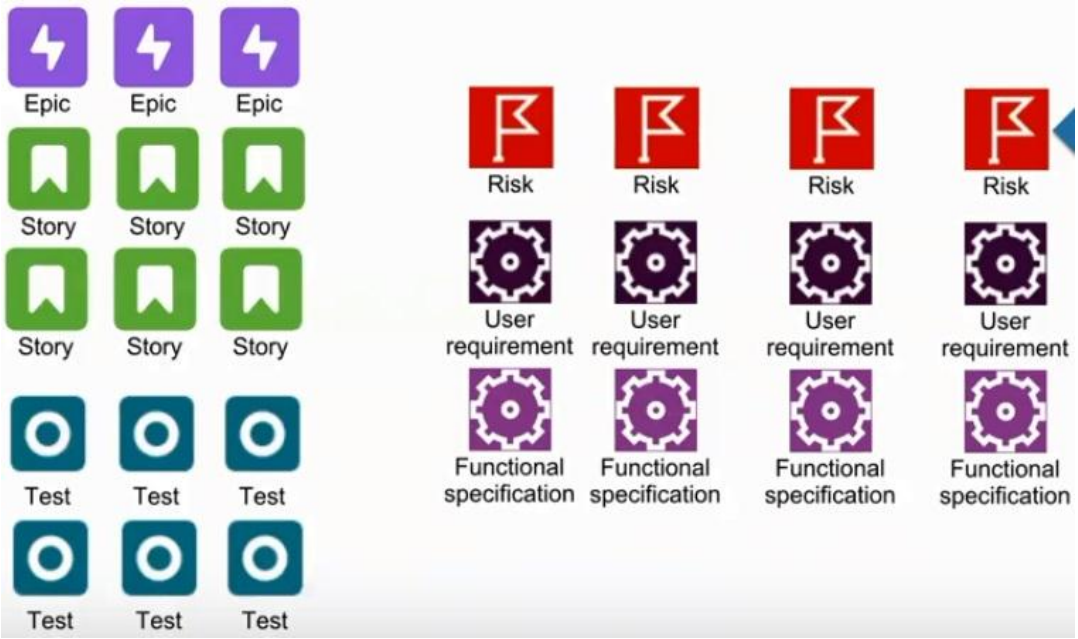


Fig 4.5: Traceability identifies Risk for each requirement

CHAPTER 5

RESULTS AND ANALYSIS

5.1 CRM Next

5.1.1 Purpose:

The purpose of CRM (Customer Relationship Management) is to ensure the long-lasting relationship with customers. It is effective business strategy because it provides customer data in greater perspective. The focus of CRM is to bring value for the customer and organization in future. It frequently enables the organizations to have an advantage over other suppliers or competitors that provides similar services.

5.1.2 Over-view of Project:

Usually there are three main types of CRM software systems operational, analytical and collaborative. However, this particular project is related to operational type of CRM system. The previous version was known as CRM. So, it's the new version that intends to cover more customers and regions known as CRM-Next. Following are the main users of CRM-Next:

- **Project Admin**

  The project administrator manages the over-all activities of related project appropriately.

- **Project Managers**

  The project manager deals with customer and their appropriate data.

- **Deal Manager**

  The role of deal manager is to utilize every business opportunity at optimum level.

- **Customer**

  The individuals who really want to buy our products and services.

5.1.3 Tools:

- **Jira Over-view**

Initially Jira was utilized just as a bug tracking tool. But recently, Jira has been globally recognized to tackle numerous project management issues and particularly change management. Jira software basically belongs to family of

products introduced by Atlassian. This tool really suits our research for understanding the role of requirements traceability from industrial perspective.

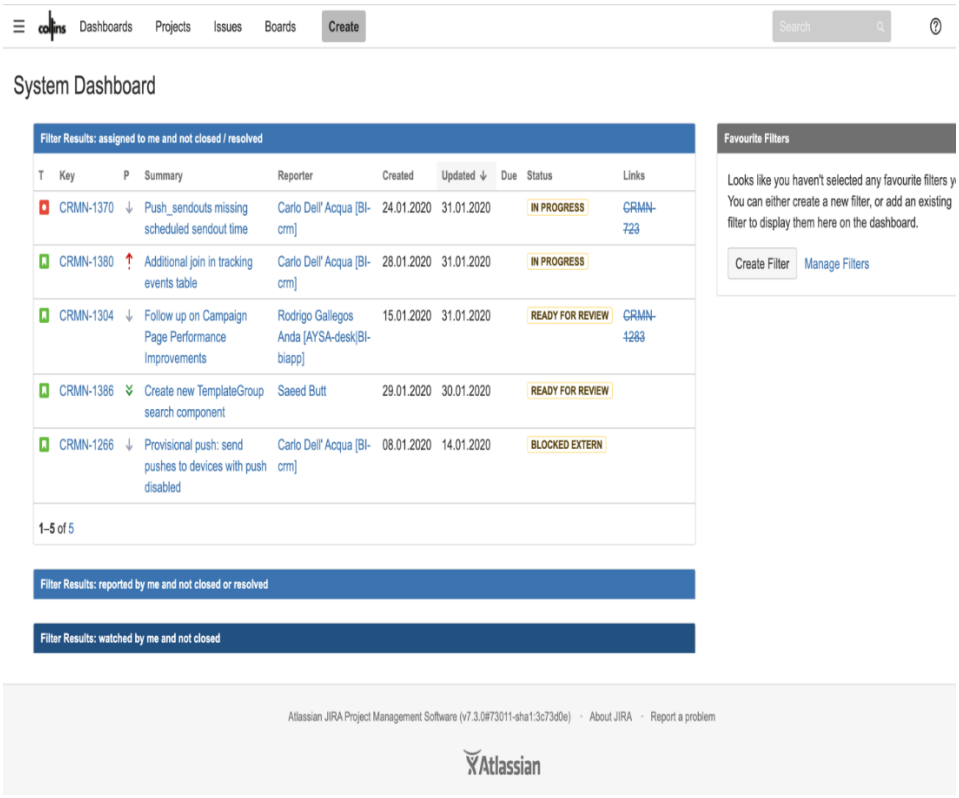


Fig 5.1: System Dashboard

Jira has particularly two types of board for agile software development.

- Kanban board
- Scrum board

- **Kanban board**

A kanban board is also a project management tool used in agile development. It is designed to aid in the progress of work, change management and enhancing the efficiency of the project. The kanban board uses card, various columns and continuous improvement techniques to tackle evolving technology or complexity. It really helps several teams commit to appropriate amount of work. Kanban board provides really unique features to complete the tasks.

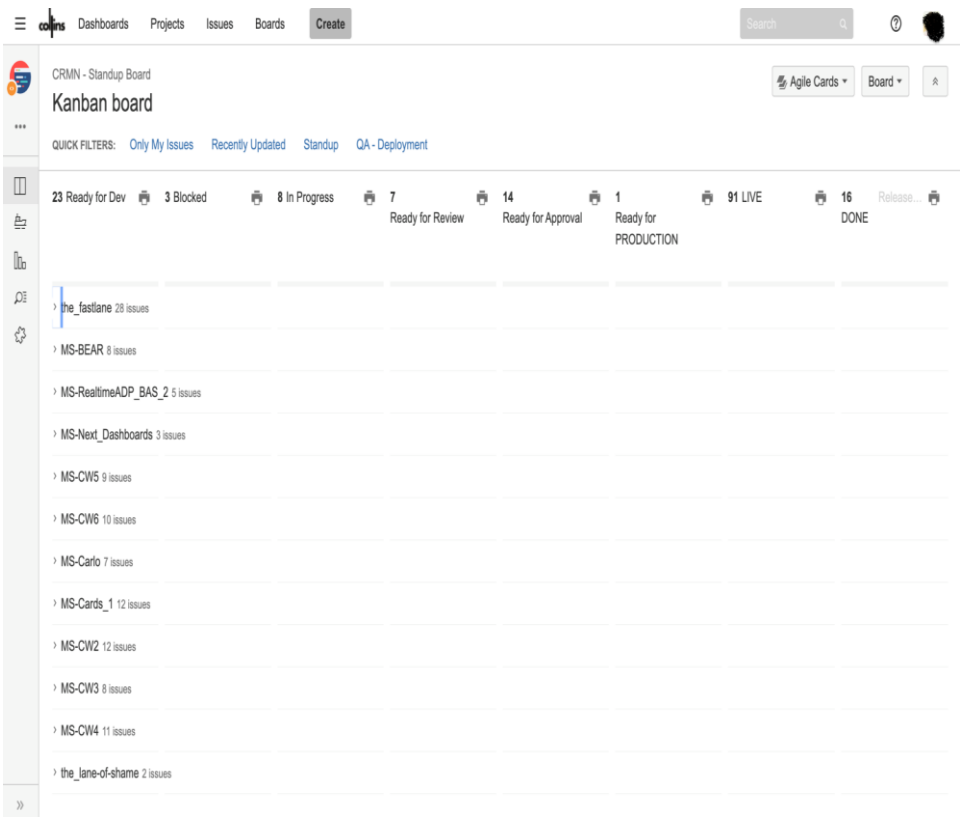


Fig 5.2: Kanban board

- **Standup board**

The creation of multiple boards depends upon the nature of our project and its requirements. However, this project is divided into further three boards planning board, channels requests board and the most paramount one the standup board. Standup board really plays important role to understand the role of requirements traceability in change management. As we can observe in below Fig.5.3 that boards can be categorized into 5 components.

- Visual signals
- Columns
- Work-in-progress limits
- Commitment point
- Delivery point

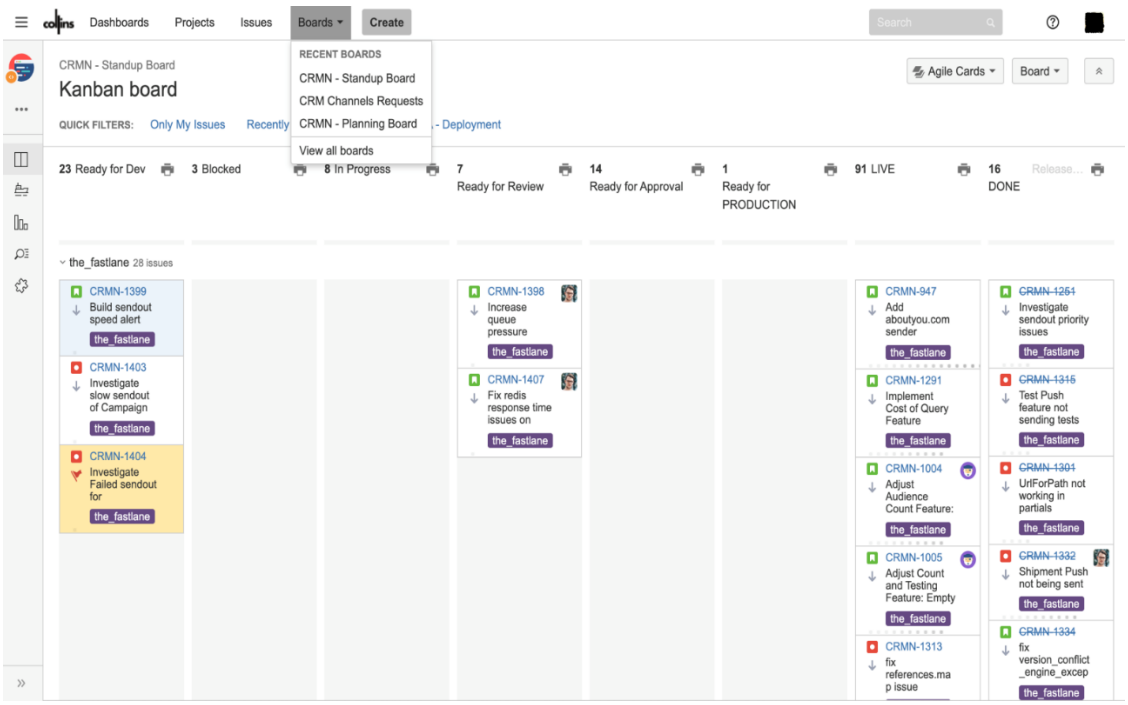


Fig 5.3: Standup board

- **Channels Requests**

This board is designed for better understanding of various tags, tasks and components in the agile project management. Sprints have been categorized on the basis of its priority and utilization. As it is evident from the Fig. 5.4 the new channel request have been made by the ticket categorized as major or important from other sprints.

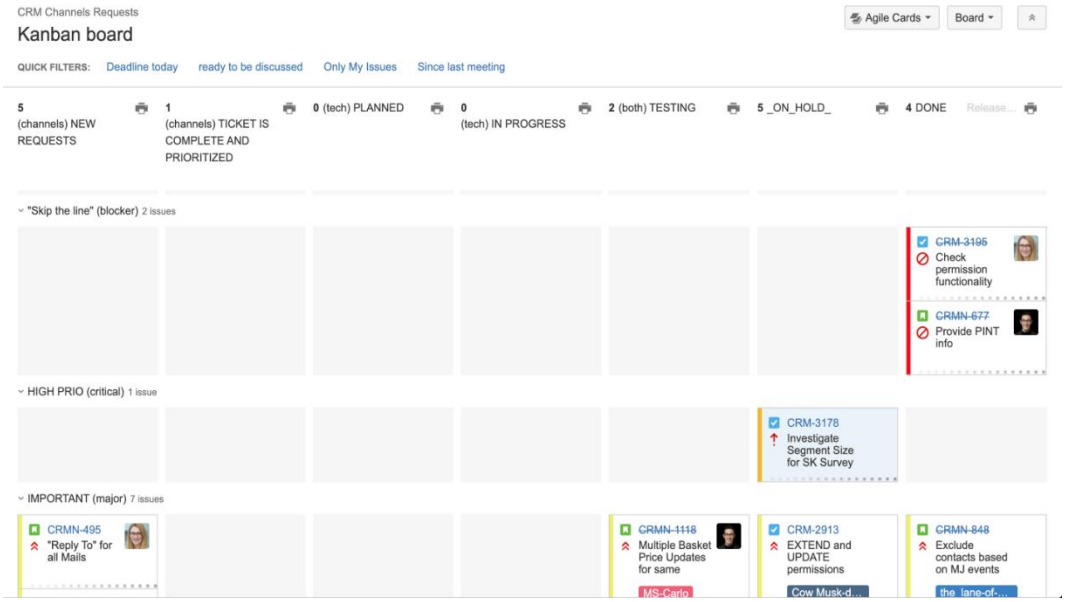


Fig 5.4: Channels Requests

- **Planning board**

The role of this board is to track the progress of the project and how new features are impacting the over-all progress of the project. As it is evident from the Fig.5.5 that planning board is running ticket for permission management in CRM-Next.

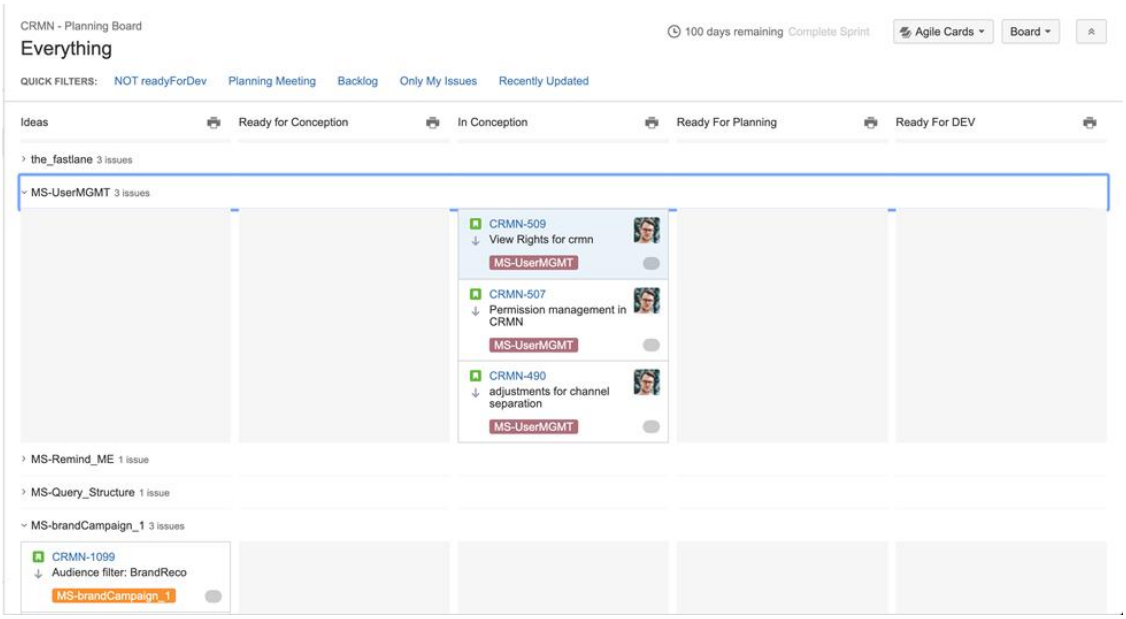


Fig 5.5: Planning Board

5.2 Jira Reports

Following reports can be generated through Jira for teams and software engineers to better utilized the process of change management and traceability:

- Version Report
- Control Chart
- Velocity Chart
- Pie Chart Report
- Sprint Report
- Epic Burn Down
- Resolution Time Report
- Created vs. Resolved Issues Report
- Cumulative Flow Diagram
- Recently Created Issues Report
- Time Since Issues Report

  Here we will discuss various reports for better understanding of our research. Control chart is comprehensive chart that helps team members and experts to over-view the progress of project in greater perspective.

5.2.1 Control Chart

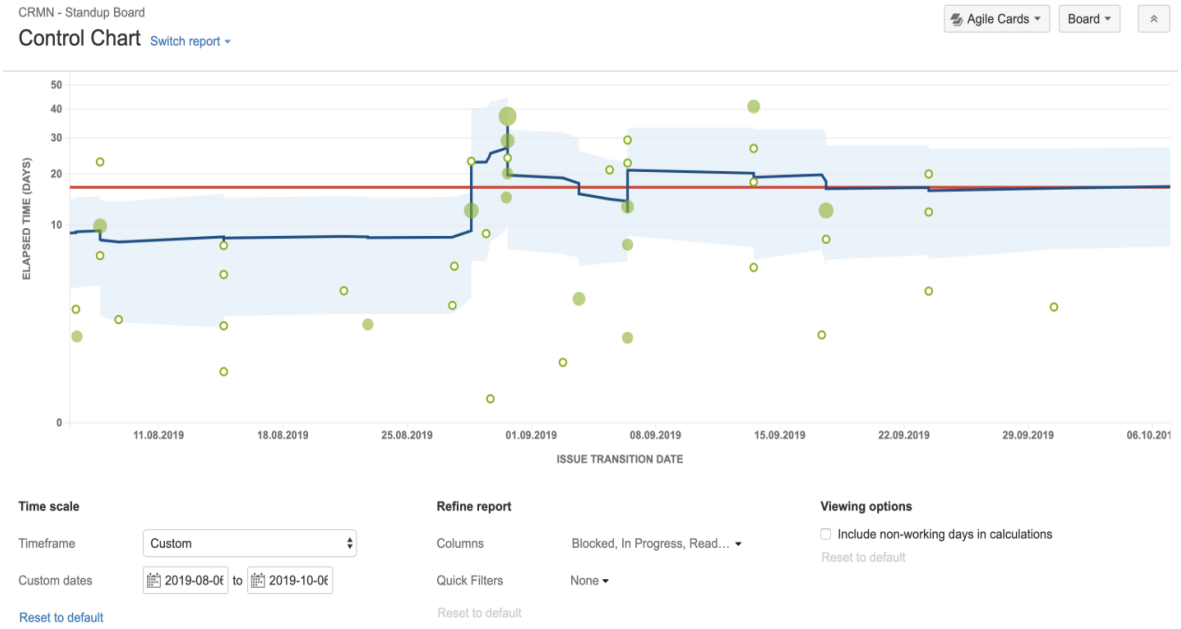


Fig 5.6: Control Chart (1)

The control chart shows cycle time of version, product and sprint. It basically tells us about the time spent on critical issues over specified time. As it is evident from the Fig.5.6 that it frequently specifies the workflow problems which further aid us in more predictable estimates. The x-axis represents time and y-axis represents number of days spent to resolve issues. The solid green dots represent clusters of issues and open green dots are single issues. Their position on the chart indicates the day they were updated or resolved and how long it took when work began or completed. This period is known as cycle time. The red line represents the over-all average cycle time. The blue line indicates the rolling average cycle time. If the blue line is below the average line then team is working efficiently otherwise its possible bottleneck.

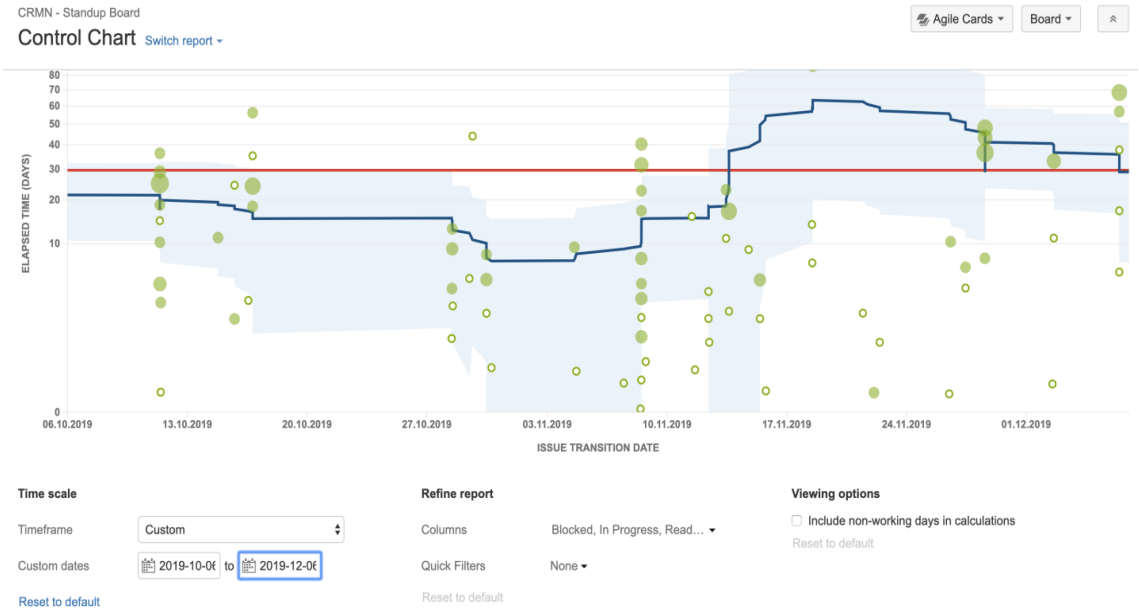


Fig 5.7: Control Chart (2)

5.2.2 Cumulative Flow Diagram:

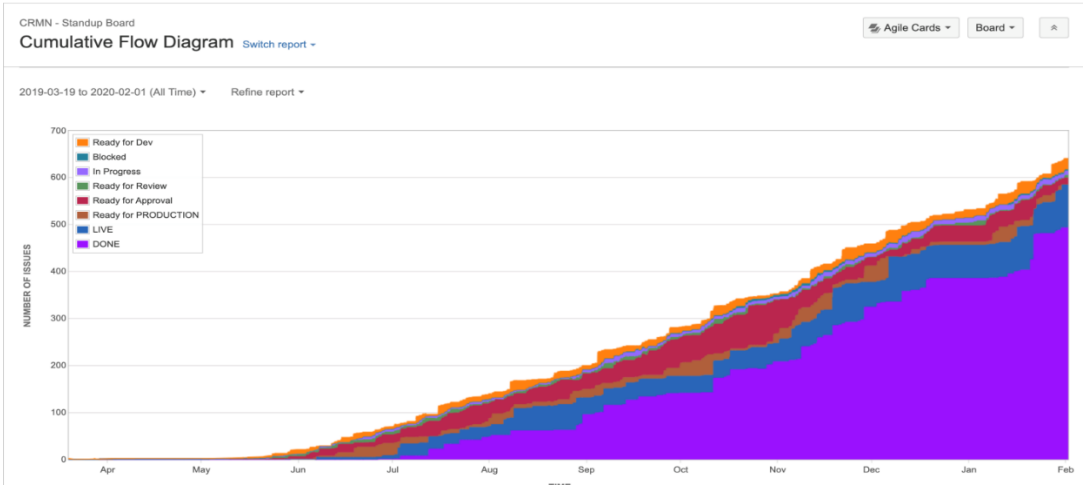


Fig 5.8: Cumulative Flow Diagram (1)

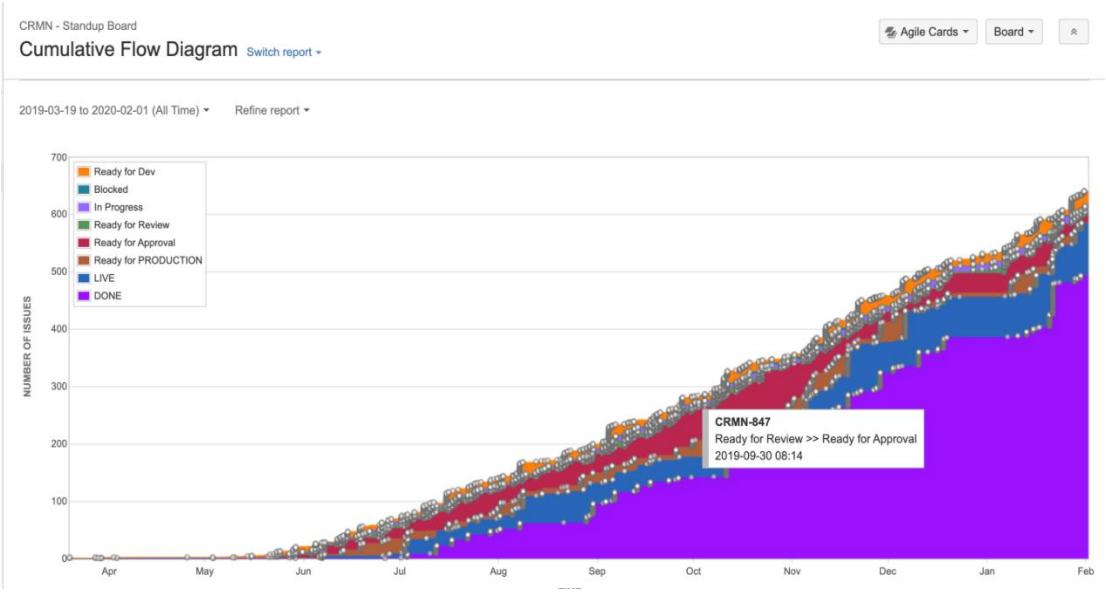


Fig 5.9: Cumulative Flow Diagram (2)

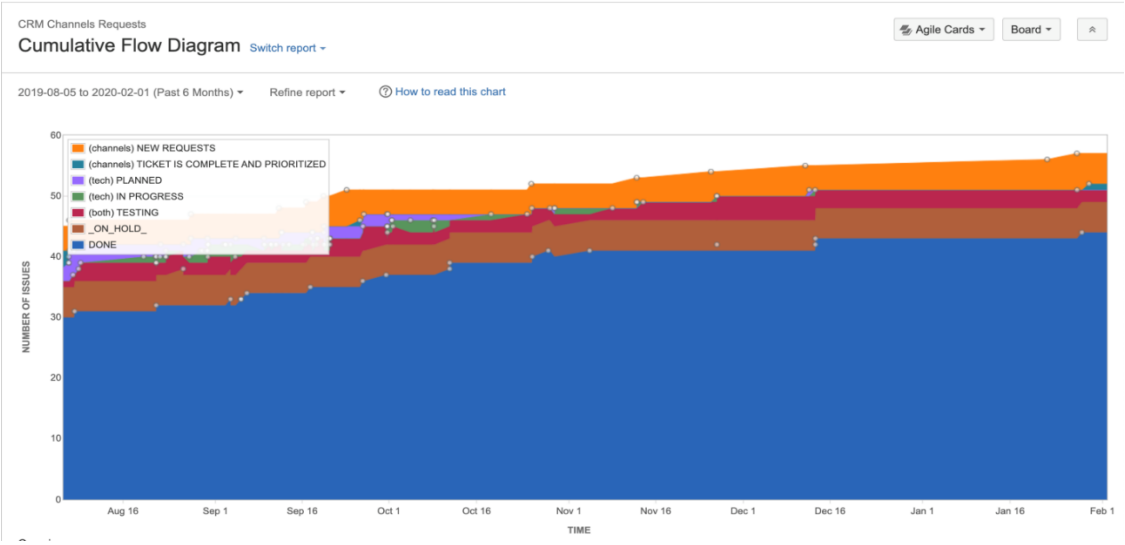


Fig 5.10: Cumulative Flow Diagram of Channels Requests

5.2.3 Resolution Time Report:

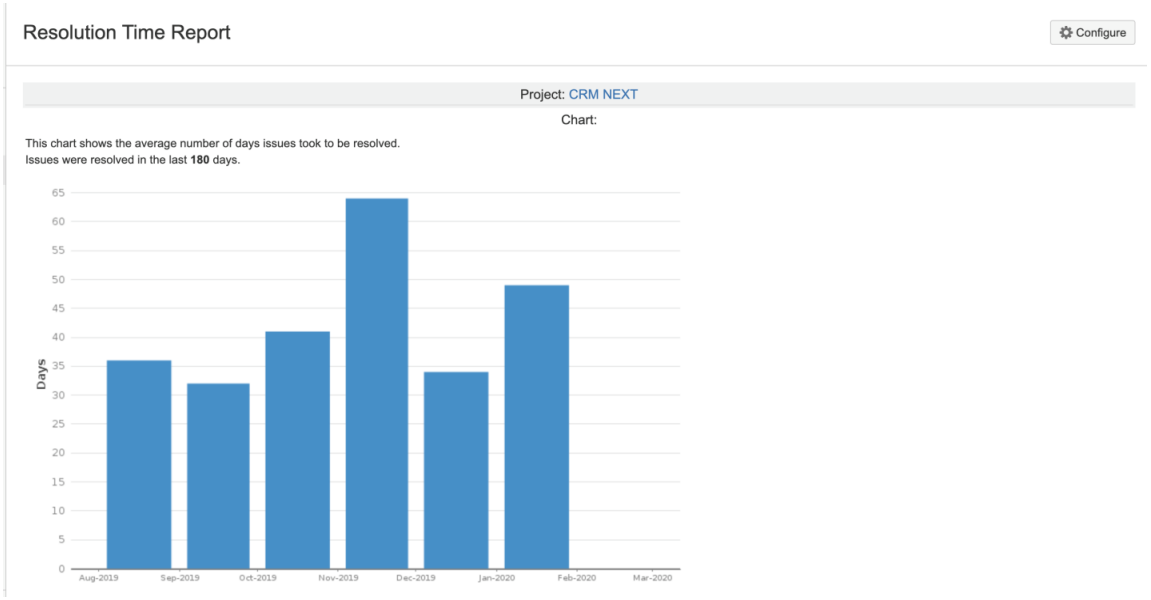


Fig 5.11: Resolution Time Report

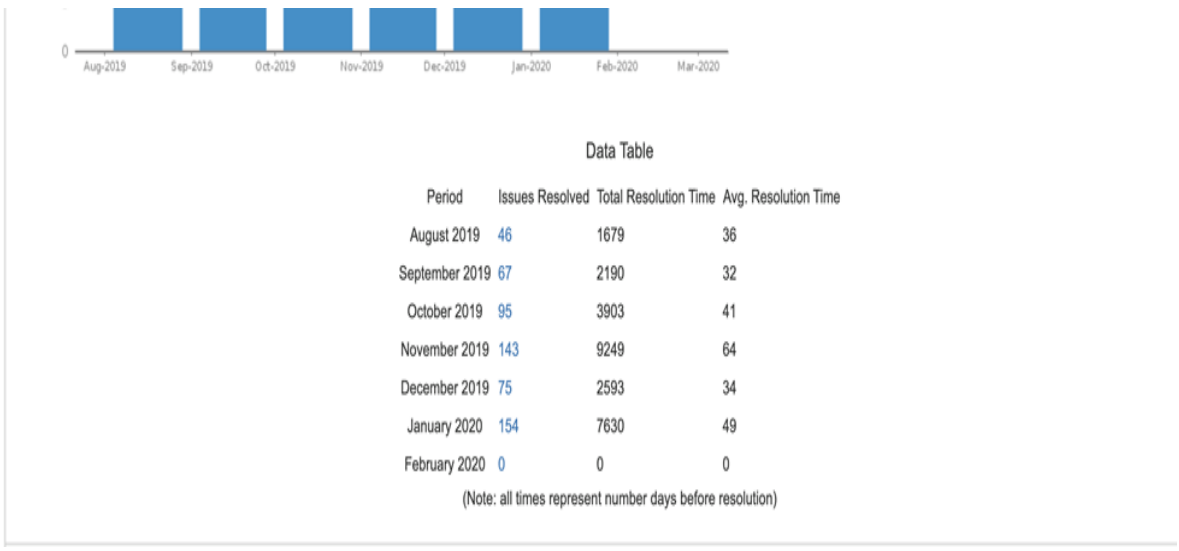


Fig 5.12: Data of Resolution Time Report

5.2.4 Pie Chart Report:

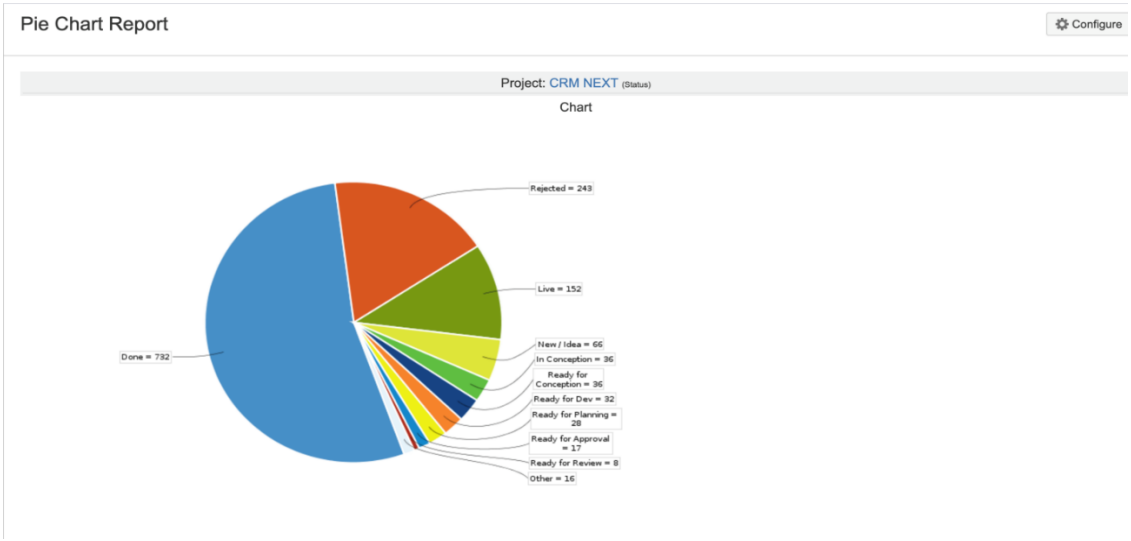


Fig 5.13: Pie Chart Report

5.2.5 Burndown Chart:

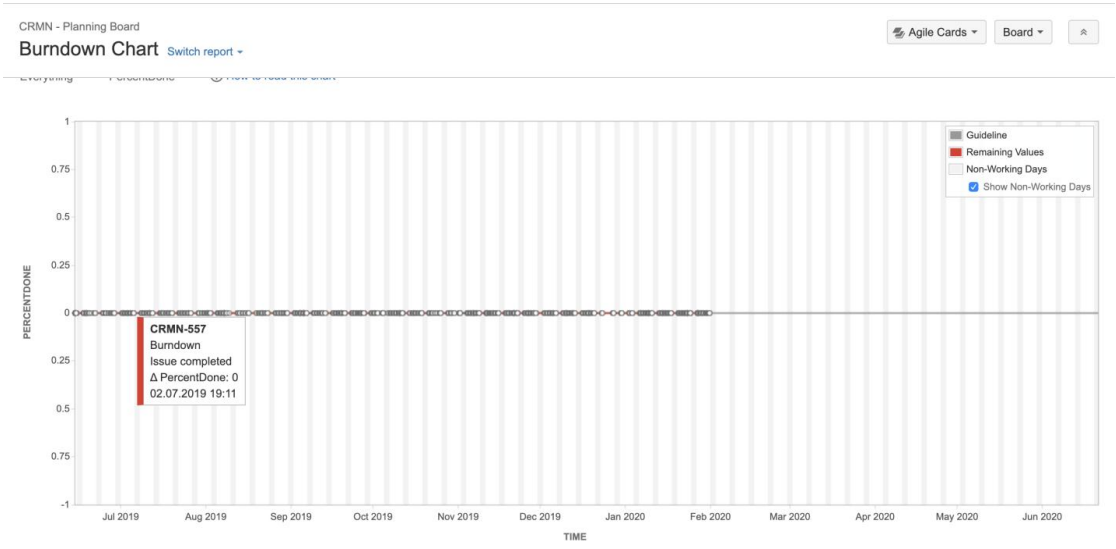
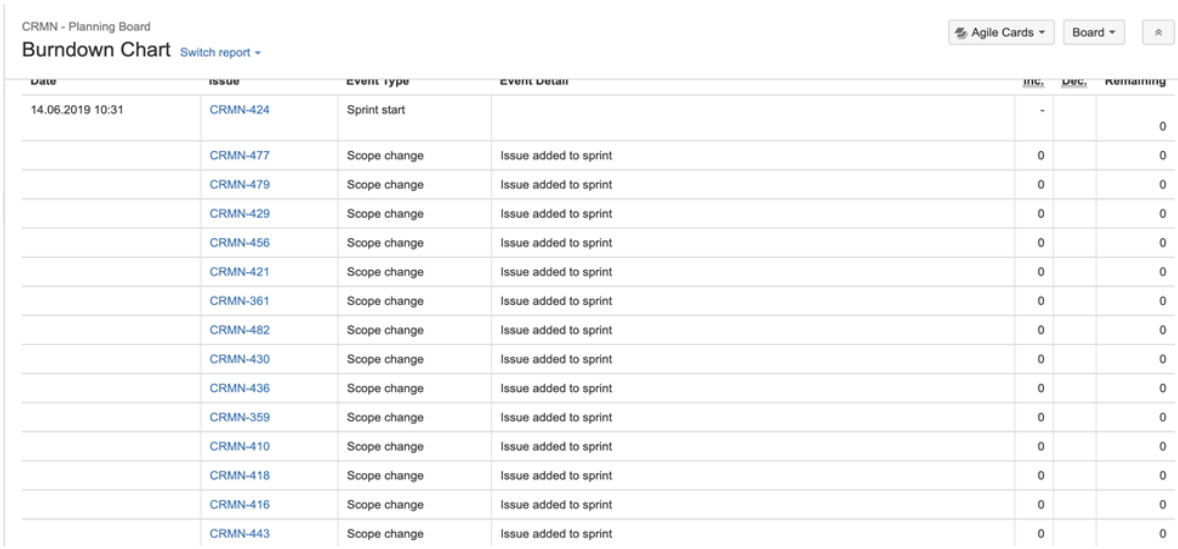


Fig 5.14: Burndown Chart of Planning Board (1)



Date	Issue	Event Type	Event Detail	Inc.	Dec.	Remaining
14.06.2019 10:31	CRMN-424	Sprint start		-		0
	CRMN-477	Scope change	Issue added to sprint	0		0
	CRMN-479	Scope change	Issue added to sprint	0		0
	CRMN-429	Scope change	Issue added to sprint	0		0
	CRMN-456	Scope change	Issue added to sprint	0		0
	CRMN-421	Scope change	Issue added to sprint	0		0
	CRMN-361	Scope change	Issue added to sprint	0		0
	CRMN-482	Scope change	Issue added to sprint	0		0
	CRMN-430	Scope change	Issue added to sprint	0		0
	CRMN-436	Scope change	Issue added to sprint	0		0
	CRMN-359	Scope change	Issue added to sprint	0		0
	CRMN-410	Scope change	Issue added to sprint	0		0
	CRMN-418	Scope change	Issue added to sprint	0		0
	CRMN-416	Scope change	Issue added to sprint	0		0
	CRMN-443	Scope change	Issue added to sprint	0		0

Fig 5.15: Burndown Chart of Planning Board (2)

5.3 Acceptance Criteria

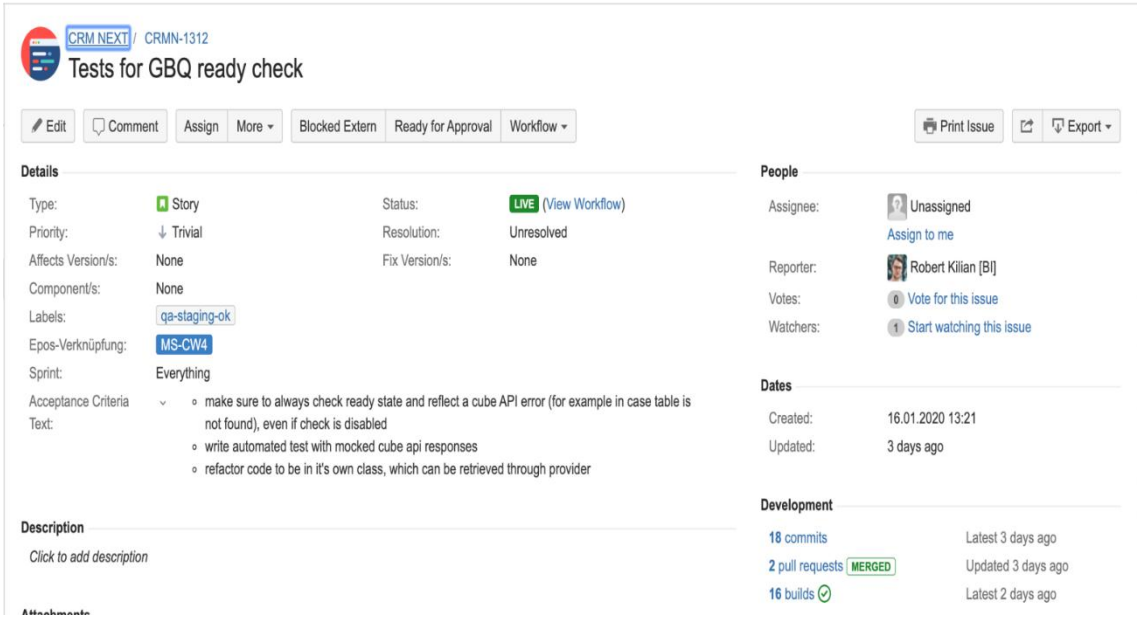


Fig 5.16: Tests for CRMN-1312

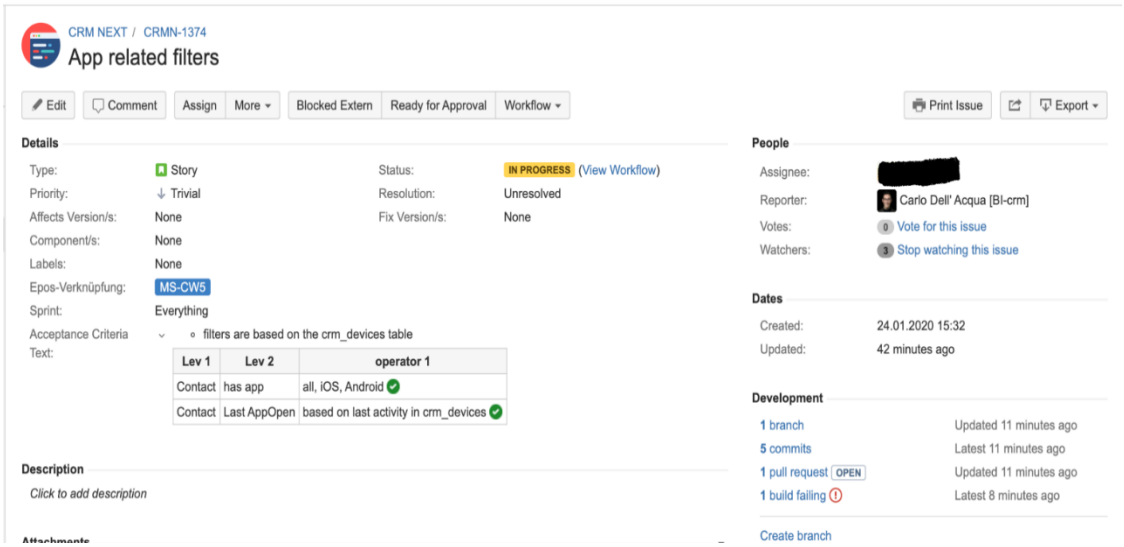


Fig 5.17: Tests for CRMN-1374

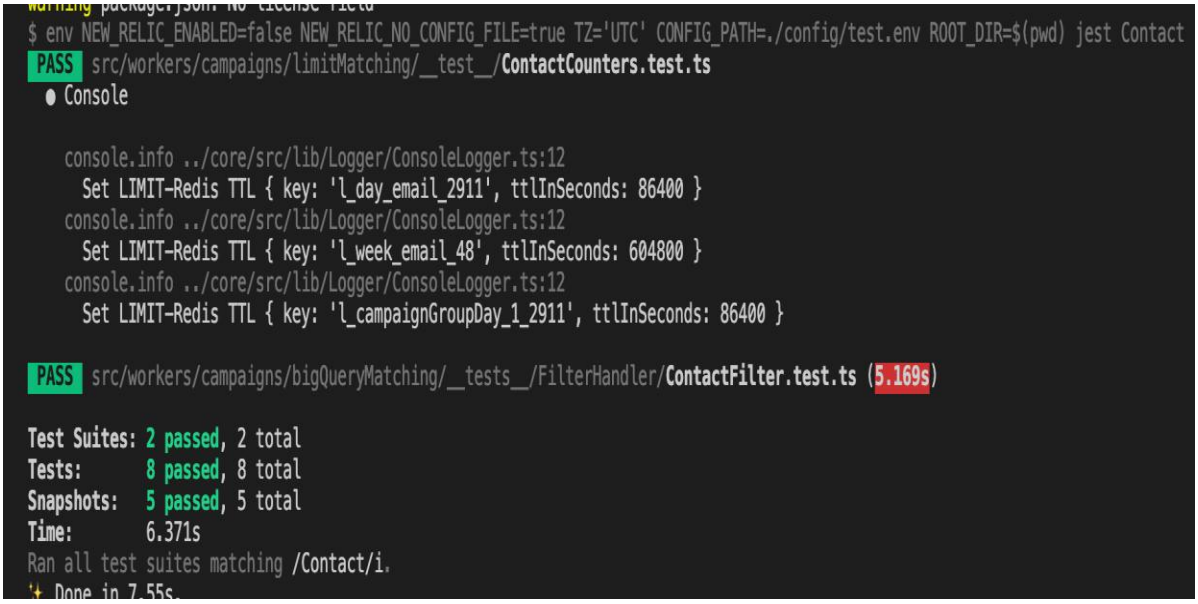


Fig 5.18: Ran test suits (1)

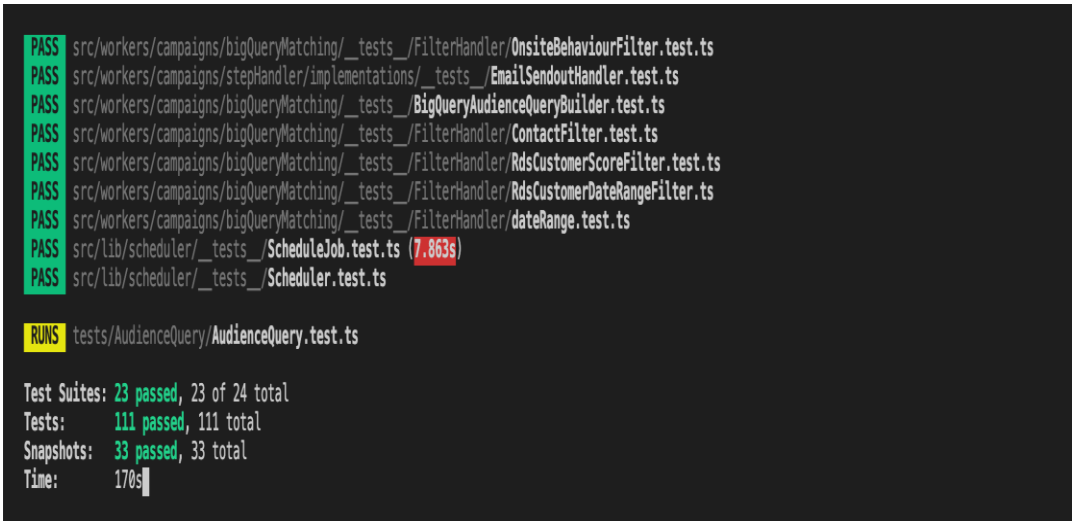


Fig 5.19: Ran test suits (2)

As it is evident from control chart and cumulative flow diagram that number of issues have been frequently increasing as the project progress. The standup board and channel requests board plays the role of requirement traceability for planning board in order to follow the process of change management effectively. It can be observed in Fig.5.13 and Fig.5.14 that there is hardly any work pending in planning board because the perks provided by requirements traceability has been utilized effectively by following proposed metrics.

Standup board tracks and records all the vital points, changed requirements and present situation of project in much better way. The channel requests board deals with all the recommended changes from the testers and customers. Further, these frequent changes are discussed in sprint standup meeting and recorded or handled in standup board. It has been shown in Fig.5.10 that the highest number of issues has been resolved by utilizing the process of traceability. The cumulative flow diagram gives us view about the change management and how by adding of new features can halts the progress of project in various ways.

Further, it can be observed in control charts that due to frequent changes it creates issues and ultimately it disturbs the working of teams. However due to effective utilization of requirements traceability, we can see the improvement in graph regardless of frequent change or variance in the project. As it is a large scale project, the Kanban

and requirements traceability always provides us the comfort zone in perspective of process improvement.

Table 5.1: Results attained in change management

Serial Number	**Metrics**	**Traceability Utilized**	**Results**	**Responsibilities**
RTM-1	Traceable Relationships	TU- I, II	Fig 5.6	It enables us to track the origin of the requirements in over-all project.
RTM-2	Decision making	TU- III	Fig 5.10,Fig 5.12	It tends to take appropriate decisions on the basis of traceable relationships
RTM-3	Test plans	TU- V	Fig 5.19	It is responsible for appropriate mapping of requirements through testing cycle
RTM-4	Risk Mitigation	TU- III, V	Fig 5.7	It enables to mitigate the risk factor involves in project development
RTM-5	Quality Enhancement	TU- IV, V	Fig 5.12,Fig 5.19	It tends to enhance the quality of project and management.
RTM-6	Formal reviews	TU- IV	Fig 5.5	It tends to review the progress of team and their productivity.
RTM-7	Standard Deviation	TU-III,V	Fig 5.6	It enables to recognize the variance or change in the project
RTM-8	Issues Resolved	TU- III	Fig 5.11	It resolves the issues because of their

				timely detection in testing.
RTM-9	Average Resolution Time	TU-IV,V	Fig 5.12	It is responsible to over-view the resolution time which further aids in prediction.
RTM-10	Bugs Per Sprint	TU- III	Fig 6.1	It enables us to analyze the product complexity.
RTM-11	Time Duration	TU-III,V	Fig 6.1,Fig 6.2	It is responsible to ease the progress of the project.
RTM-12	Process Optimization	TU- V	Fig 6.2	It is responsible to optimize the process of traceability in every sprint.

How the requirements will be effectively traceable?

The requirements will be effectively traceable by utilizing the process of requirements traceability. It has been evident from our literature that requirement traceability is still under-utilized process. Moreover, the real hurdle in requirement traceability is the quality of traces that aid in effective tracking of requirements. So, the research have really help us to understand the concept of requirements traceability and how the quality of traces can be improved. The effective utilization of requirements traceability in its linking process largely depends upon the quality of traces. The quality of these links has been improved after implementing the appropriate measures provided by various traceability techniques. However, if there is any change in customer requirements, then these traceability links are utilized for impact analysis. The link (forward from requirements) identifies how one or multiple requirements have been assigned to several system components. Similarly, this type of link is really effective in evaluating the impact of requirements change. Further, these links aids in verifying and validating of requirements according to technical assumptions and customer needs.

How can management of various traceable requirements will impacts the performance?

It is evident from control charts and cumulative diagrams that the processes of traceability and management are directly related to each other. Therefore, the process of change management basically has to trade-off between the several requirements traceability activities which further enhances the role of RT in change management. The traceability is a continuous process and in order to achieve desired performance, we have to focus on every requirement and supporting activity. Later on, if the requirement of a user changes and if the system is unable to recognize it timely, then the existing traces will remain suspicious as well as the overall quality of the system.

So, management of various traceable requirements and its related traces has a great impact on over-all performance of the system. We have identified that market or industrial experts are reluctant to use any modern-day management solutions because of its complexity and high costs. The effective management of traceable requirements ensures the availability of requirements and how the requirements have evolved in over-all system. Further, the availability of requirements directly impacts the performance and progress of project in perspective of change management.

How the proposed mechanism can aids the overall effectiveness of Requirements Traceability?

It has been recognized that the process of requirements traceability is really vital in the development of various projects. The proposed measure really helps in understanding the features provided by requirements traceability and how it can be utilized frequently to resolve our project issues. The change management has rediscovered the process of requirements traceability at larger perspective. The metrics are providing all the available privileges of requirements traceability to implement the process of change management effectively. As it is evident from the results that the effective utilization of requirement traceability really aids in tracking of defects. Ultimately, it improves the quality of product by effective tracking of defects related to appropriate requirements. It is possible due to previous good work of requirements traceability in managing of requirements. After achieving satisfactory results, it verifies that the proposed mechanism truly aids the over-all effectiveness of requirements traceability in larger perspective.

CHAPTER 6

CONCLUSION

As the complexity is increasing, it is really vital for us to understand the need for requirements traceability in modern software engineering. Managing of requirements and change management are not easy tasks. The change management is really paramount phase in development of any project. It has been quite clear from our findings that requirements traceability keeps the track of all requirements from its origin to evolution. Recently; it has been observed that market experts are reluctant to utilize modern day management solutions due to its higher cost and time. It has been evident from the control charts that after utilizing the perks provided by requirements traceability, we have been able to improve the efficiency of our work regardless of the frequent variance or change in the system. The process of change management can't be effective without requirements traceability. The research identifies that requirement traceability should be recognized as heart of requirement engineering. It really glorifies and aids in all the phases of development especially in change management and testing. As far as system traceability is concerned, requirement volatility is still the main challenge in software engineering and it is now under-utilized but findings of our research have paved the way for other researchers on bigger canvas.

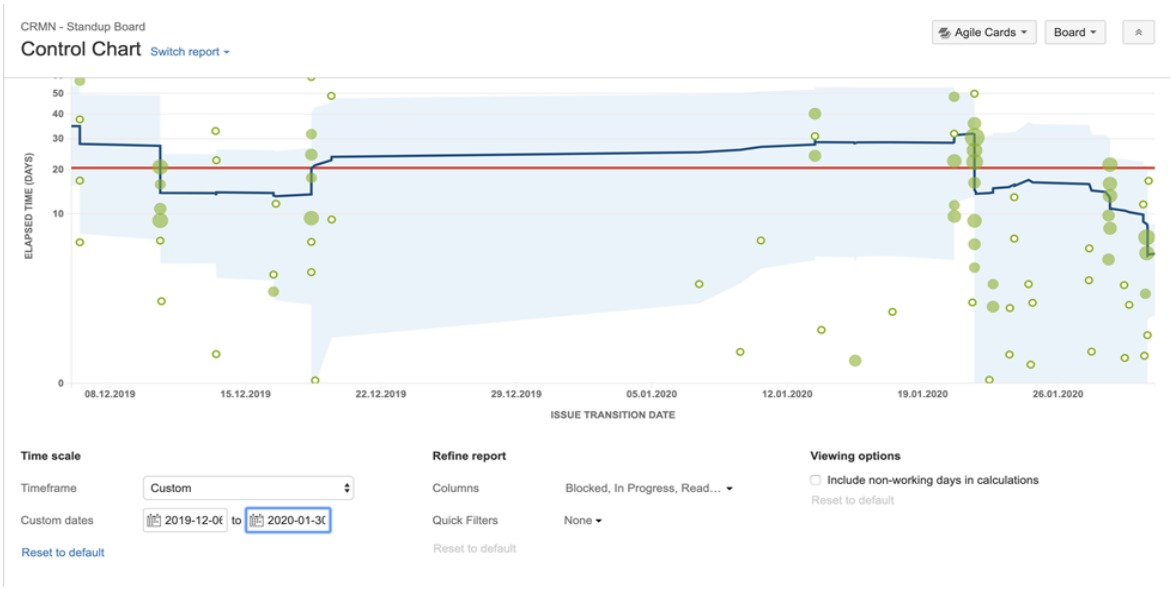


Fig 6.1: Final Control Chart (1)

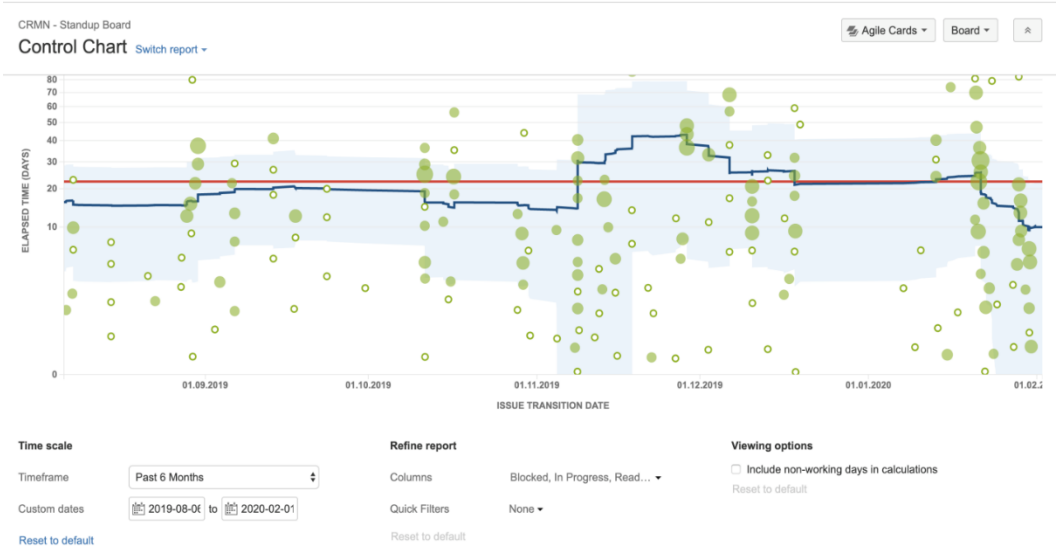


Fig 6.2: Final Control Chart (2)

REFERENCES

[1] B. Venkatesh and L. Balani, “Requirement Management a Key To Successful Project Management,” vol. 5, no. 1, pp. 49–51, 2016.

[2] P. Mäder and A. Egyed, “Do developers benefit from requirements traceability when evolving and maintaining a software system?,” *Empir. Softw. Eng.*, vol. 20, no. 2, pp. 413–441, 2015, doi: 10.1007/s10664-014-9314-z.

[3] D. Farrar and J. H. Hayes, “A comparison of stemming techniques in tracing,” *Proc. - 2019 IEEE/ACM 10th Int. Work. Softw. Syst. Traceability, SST 2019*, pp. 37–44, 2019, doi: 10.1109/SST.2019.00017.

[4] J. Cleland-Huang, O. C. Z. Gotel, J. H. Hayes, P. Mäder, and A. Zisman, “Software traceability: Trends and future directions,” *Futur. Softw. Eng. FOSE 2014 - Proc.*, pp. 55–69, 2014, doi: 10.1145/2593882.2593891.

[5] D. Stahl, K. Hallen, and J. Bosch, “Continuous integration and delivery traceability in industry: Needs and practices,” *Proc. - 43rd Euromicro Conf. Softw. Eng. Adv. Appl. SEAA 2017*, no. October, pp. 61–65, 2017, doi: 10.1109/SEAA.2017.19.

[6] P. Rempel and P. Mader, “Preface,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9013, pp. V–VI, 2015, doi: 10.1007/978-3-319-16101-3.

[7] A. Marques, F. Ramalho, and W. L. Andrade, “TRL - A Traceability Representation Language,” *Proc. ACM Symp. Appl. Comput.*, vol. 13-17-Apri, pp. 1358–1363, 2015, doi: 10.1145/2695664.2695745.

[8] H. B. Yadav and D. K. Yadav, “Early software reliability analysis using reliability relevant software metrics,” *Int. J. Syst. Assur. Eng. Manag.*, vol. 8, no. December, pp. 2097–2108, 2017, doi: 10.1007/s13198-014-0325-3.

[9] A. Ghabi and A. Egyed, “Code patterns for automatically validating requirements-to-code traces,” *2012 27th IEEE/ACM Int. Conf. Autom. Softw. Eng. ASE 2012 - Proc.*, pp. 200–209, 2012, doi: 10.1145/2351676.2351705.

[10] J.-P. Steghöfer, N. Niu, J. L. C. Guo, and A. Mahmoud, “SST’19 - Software and Systems Traceability,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 44, no. 3, pp. 43–47, 2019, doi: 10.1145/3356773.3356806.

[11] R. Tsuchiya, T. Kato, H. Washizaki, M. Kawakami, Y. Fukazawa, and K. Yoshimura, “Recovering traceability links between requirements and source code in the same series of software products,” *ACM Int. Conf. Proceeding Ser.*, pp. 121–130, 2013, doi: 10.1145/2491627.2491633.

[12] R. Elamin and R. Osman, “Towards Requirements Reuse by Implementing Traceability in Agile Development,” *Proc. - Int. Comput. Softw. Appl. Conf.*,

vol. 2, pp. 431–436, 2017, doi: 10.1109/COMPSAC.2017.250.

[13] F. Khursheed and M. Suaib, “A Survey on Importance of Requirement Traceability in Software Engineering,” *Int. J. Eng. Innov. Technol.*, vol. 5, no. 4, pp. 109–112, 2015.

[14] M. Gupta and A. Kalia, “Empirical Study of Software Metrics,” *Res. J. Sci. Technol.*, vol. 9, no. 1, p. 17, 2018, doi: 10.5958/2349-2988.2017.00003.1.

[15] “Computing Technologies.” [Online]. Available: http://computingtechnologies.blogspot.com/. [Accessed: 21-Feb-2020].

[16] T. Mens, R. Van Der Straeten, and J. Simmonds, *A Framework for Managing Consistency of Evolving UML Models*, no. February 2015. 2011.

[17] S. Namdar and M. Mirakhorli, “Toward Actionable Software Architecture Traceability,” *Proc. - 2015 IEEE/ACM 8th Int. Symp. Softw. Syst. Traceability, SST 2015*, pp. 36–42, 2015, doi: 10.1109/SST.2015.17.

[18] O. Gotel *et al.*, “The quest for Ubiquity: A roadmap for software and systems traceability research,” *2012 20th IEEE Int. Requir. Eng. Conf. RE 2012 - Proc.*, pp. 71–80, 2012, doi: 10.1109/RE.2012.6345841.

[19] G. Arévalo, G. Robiolo, and M. M. Soler, “Traceable complexity metric from requirements to code,” *ESEM 2010 - Proc. 2010 ACM-IEEE Int. Symp. Empir. Softw. Eng. Meas.*, vol. 1917, no. 1033, p. 4503, 2010, doi: 10.1145/1852786.1852866.

[20] P. Rempel and P. Mader, “Preventing defects: The impact of requirements traceability completeness on software quality,” *IEEE Trans. Softw. Eng.*, vol. 43, no. 8, pp. 777–797, 2017, doi: 10.1109/TSE.2016.2622264.

[21] M. Seiler, P. Hubner, and B. Paech, “Comparing traceability through information retrieval, commits, interaction logs, and tags,” *Proc. - 2019 IEEE/ACM 10th Int. Work. Softw. Syst. Traceability, SST 2019*, no. Il, pp. 21–28, 2019, doi: 10.1109/SST.2019.00015.

[22] N. Niu, W. Wang, and A. Gupta, “Gray links in the use of requirements traceability,” *Proc. ACM SIGSOFT Symp. Found. Softw. Eng.*, vol. 13-18-Nove, pp. 384–395, 2016, doi: 10.1145/2950290.2950354.

[23] R. Sinha, B. Dowdeswell, G. Zhabelova, and V. Vyatkin, “TORUS: Scalable requirements traceability for large-scale cyber-physical systems,” *ACM Trans. Cyber-Physical Syst.*, vol. 3, no. 2, 2018, doi: 10.1145/3203208.

[24] “Jama Software.” [Online]. Available: https://resources.jamasoftware.com/. [Accessed: 21-Feb-2020].

[25] V. Csuvik, A. Kicsi, and L. Vidacs, “Source code level word embeddings in aiding semantic test-to-code traceability,” *Proc. - 2019 IEEE/ACM 10th Int. Work. Softw. Syst. Traceability, SST 2019*, pp. 29–36, 2019, doi: 10.1109/SST.2019.00016.

[26] “ENGINEERING.com | Information & Inspiration for Engineers.” [Online]. Available: https://www.engineering.com/. [Accessed: 21-Feb-2020].

[27] I. Hajri, A. Goknil, L. C. Briand, and T. Stephany, “Change Impact Analysis for Evolving Configuration Decisions in Product Line Use Case Models,” *2019 IEEE/ACM 10th Int. Symp. Softw. Syst. Traceability*, vol. 17, no. 3, pp. 11–11, 2019, doi: 10.1109/sst.2019.00011.

[28] “Linknovate | Your discovery engine.” [Online]. Available: https://www.linknovate.com/. [Accessed: 21-Feb-2020].

[29] “i-Scholar.” [Online]. Available: http://www.i-scholar.in/. [Accessed: 21-Feb-2020].

[30] M. Vierhauser, J. Cleland-Huang, J. Burge, and P. Grunbacher, “The interplay of design and runtime traceability for non-functional requirements,” *Proc. - 2019 IEEE/ACM 10th Int. Work. Softw. Syst. Traceability, SST 2019*, pp. 3–10, 2019, doi: 10.1109/SST.2019.00010.

[31] A. De Lucia, F. Fasano, R. Oliveto, and G. Tortora, “Recovering traceability links in software artifact management systems using information retrieval methods,” *ACM Trans. Softw. Eng. Methodol.*, vol. 16, no. 4, 2007, doi: 10.1145/1276933.1276934.

[32] A. Vizcaíno, F. García, I. G. R. De Guzmán, and M. Á. Moraga, “Evaluating GSD-aware: A serious game for discovering global software development challenges,” *ACM Trans. Comput. Educ.*, vol. 19, no. 2, pp. 1–23, 2019, doi: 10.1145/3218279.

[33] M. Shahid, S. Ibrahim, and M. N. ri Mahrin, “An evaluation of requirements management and traceability tools,” *World Acad. Sci. Eng. Technol.*, vol. 78, no. 6, pp. 596–601, 2011.

[34] S. Akman, M. Özmut, B. Aydın, and S. Göktürk, “Experience report: implementing requirement traceability throughout the software development life cycle,” *J. Softw. Evol. Process*, vol. 28, no. 11, pp. 950–954, 2016, doi: 10.1002/smr.1824.

[35] A. Gupta and P. Bera, “A software tool to convert requirements to test cases,” *Proc. - 2019 IEEE/ACM 6th Int. Work. Requir. Eng. Testing, RET 2019*, pp. 9–12, 2019, doi: 10.1109/RET.2019.00009.

[36] D. A. Kumar and P. Raviraj, “Enhancing Requirement Traceability Link Using User ’ s Updating Activity,” vol. 3, no. 3, pp. 1939–1943, 2014.

[37] M. Johara and T. Hemalatha, “Improve the Accuracy of Requirement Traceability Links using Requirement Severity,” *IJERT*, vol. 3, no. 3, 2014.

[38] M. Rahimi and J. Cleland-Huang, “Evolving software trace links between requirements and source code,” *Proc. - 2019 IEEE/ACM 10th Int. Work. Softw. Syst. Traceability, SST 2019*, p. 12, 2019, doi: 10.1109/SST.2019.00012.

[39] H. MS, F. Rasheed, and K. MR, “An Improved Model for Requirement Management System,” *J. Inf. Technol. Softw. Eng.*, vol. 07, no. 01, pp. 1–3, 2017, doi: 10.4172/2165-7866.1000196.

[40] “SDLC vs STLC: What’s the Difference?” [Online]. Available: https://www.guru99.com/sdlc-vs-stlc.html. [Accessed: 21-Feb-2020].

[41] K. V. J. Padmini, H. M. N. Dilum Bandara, and I. Perera, “Use of software metrics in agile software development process,” *MERCon 2015 - Moratuwa Eng. Res. Conf.*, pp. 312–317, 2015, doi: 10.1109/MERCon.2015.7112365.

[42] H. Sedehi and G. Martano, “Metrics to evaluate & monitor agile based software development projects - A fuzzy logic approach,” *Proc. 2012 Jt. Conf. 22nd Int. Work. Softw. Meas. 2012 7th Int. Conf. Softw. Process Prod. Meas. IWSM-MENSURA 2012*, pp. 99–105, 2012, doi: 10.1109/IWSM-MENSURA.2012.22.

[43] R. Tommy, M. Mhaisekar, S. Kallepally, L. Varghese, S. Ahmed, and M. D. Somaraju, “Dynamic quality control in agile methodology for improving the quality,” *2015 IEEE Int. Conf. Comput. Graph. Vis. Inf. Secur. CGVIS 2015*, pp. 233–236, 2016, doi: 10.1109/CGVIS.2015.7449927.