# An Effectual Approach for Accuracy of Requirement Traceability Links During Software Development

Mr. Vinayak M. Sale[1], Mr. H.M. Tamboli[2], Mr. N. M. Maske[3] & Dr. B. C. Melinamath[4]
[1, 2, 3] *Asst. Prof. Department of Computer Science & Technology, SVERI's COE, Pandharpur.*
[4] *Associate Prof. Department of Computer Science & Technology, SVERI's COE, Pandharpur.*
[1]vmsale@coe.sveri.ac.in, [2]hmtamboli@coe.sveri.ac.in, [3]nmmaske@coe.sveri.ac.in,
[4]bcmelinamath@coe.sveri.ac.in

***ABSTRACT:*** *Requirement fulfillment is most important factor to the success of software. The requirements are specified by the different stakeholders should be fulfilled to each point of the development of the software. Requirement traceability assists software engineers for tracing the requirement from its starting point to its completion. Traceability is used to ensure that source code of a system is consistent with its requirements. The only specified requirement has been implemented by developers. During software maintenance and evolution, requirement traceability links become marginal because no developer can devote effort to update it. However, to recover traceability links later is a very painful and monotonous task also it is costly for developers too. Traceability supports the software development process in various ways, like as change management, software maintenance and prevention of misunderstandings. But, while, in practice, traceability links between requirements and codes are not created during the development of software as it requires extra efforts. So developers rarely use such links during development. Why many challenges exist in traceability practices today? However, many of the challenges can be overcome through organizational policy, quality requirements traceability tool support remains the open problem.*

***Keywords: Traceability, requirement, management, requirement traceability approach (RTA), IR Technique (IRT)***

## 1. INTRODUCTION

Basically, for any software development task, a developer must know the project background, in particular, the architecture of the system, design, working, and the relations between the various components using any available documentation. Program conception occurs in a bottom-up method, a top-down method or some combination of both [1] [6].

The traceability is very most necessary for any software project, and it could be useful from different perspectives for the development. When we develop a system, the source code can be traced and become the same with the requirement and analysis because we develop a source code as per the requirements. [1] [2] A traceability linkage is a correlation involving the requirement and source code. Requirement traceability assists developers to categorize the requirement from its appearance to its completion [1] [4] [7]. A prime objective of traceability link is in the involvement of original requirements and other parts traceability links [2]. As a result, we should make clear the traceability from all the scale of traceability regarding scope and coverage.

Traceability of requirement is explained as, "the capability to show and go after the existence of a requirement, in cooperation with the go ahead and toward the reverse side" [2][4] [7] [14] [22] [23]. Although updating the software, the developers can put in, eliminate, or change features as per the users demand. Even if software development and maintenance, requirement traceability links become insignificant since developer cannot dedicate try to revise it. However, to recover traceability links later is a very hectic and

tedious task also it is costly for developers too [5] [6]. Source code is not modified time to time by developer with requirements. Hence, reduce in the textual match between source codes and requirements are different from each other [1] [3].

By the side of with the different methods, the fiction presented that information retrieval (IR) technique be able to recover traceability linkage mechanically with requests and associated source codes. On the other hand, precision and recall metrics of IR techniques are affected by various features, which affect the input of traceability process of requirements. Because of slight precision and/or recall, designers' assurance in the effectiveness of recovery of traceability link depressingly affected. However, to recover traceability links later is a very painful and tedious task also it is costly for developers too [6]. Goals of different stakeholders and source codes are different from each other, which decrease the textual similarity.

## TYPES OF THE TRACEABILITY LINKS

In the system execution there are four types of traceability links which are as follows:-

### 1. Forward from Requirement

There is change from requirement to requirement. To trace requirements designer to put together requirements such that a goal for definite section or part of the method that requests to be constructed, depends on that requirements. The testing side know how directly to relate with requests. Once testing is finished, a user is ready to trace a complex requests along to the definite test case and examine that the test resolve correctly validate the requirement.

### 2. Backward to Requirement

The design elements are built in backward. Starting with the simple requests and tracing in reverse and helps to unit testers who can see which part of module executes of the method, depends on necessities. If a fault is occurred during testing then the failure point is simple requirement. From this determined that the satisfaction of complex requirement would not be done.

### 3. Forward to Requirement

We go for the tracing of the documents as per in the higher that is forward to the requirement. A requirement analysis is necessary, to establish costs and possibility of a complex requirement changing. For developing any system to verify requirements as per users need.

### 4. Backward from Requirement

In this type the traceability is done into the backward direction that is from the requirement to source code in backward direction. Besides the above types of Requirement there are two types of the requirement traceability these are Pre-RS traceability and Post- RS traceability explain as follows in the fig. 1

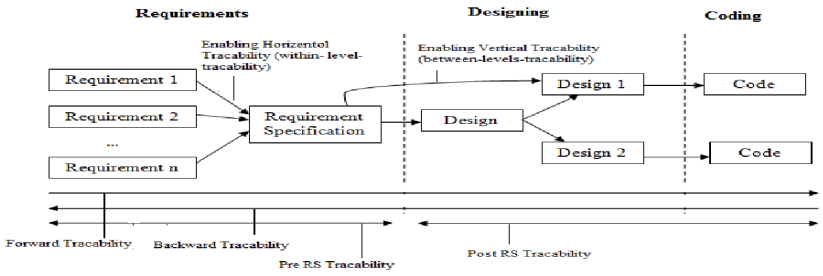


Fig 1 Different types of traceability

**The intention of traceability**

- To identify the resource of requests.
- To control the capacity of the development.
- To deal with modifications occur in requests.
- To calculate the affection of a modification occurs in prerequisites during development.
- To evaluate the impact of a failure of testing on requirements.
- To check that all requirements of the system are satisfied by the operation.

**Traceability information allows answering:**

1. What is the effect, when the requirement is changed?
2. Where is a requirement applied?
3. Are all requirements assigned?
4. Which require is deal with by a requirement?
5. Is this requirement essential?
6. What design decisions affect the implementation of a requirement?
7. What are benefits of this technique and what were the further options?
8. Is the implementation compliant with the requirements?
9. Is this design element necessary?
10. How do I interpret this requirement?

## 2. BACKGROUND AND RELATED WORK

This section presents an environment on IR technique and a review of the related work. Traceability approach can be separated into three main categories, i.e., dynamic, static, and hybrid.

Dynamic approach gathers and examines execution traces [11] to recognize technique that a software link has been carrying out in the particular scenario. However, it couldn't help to differ in overlapping circumstances, because there are some limitations to a single method. [6] The legacy system may not be applicable, due to bugs and/or some other issues. Thus, to collect execution traces is not possible.

Static traceability approaches [10], [17] use source code structure and/or textual information for recovering traceability associations among high-level and low-level software artifacts. The combination of static and dynamic information is hybrid traceability. The study shows that combination of dynamic and static information can perform better than the single IR technique [7].

For development of any software fulfillment of requirement is essential part. [1]All the requirements of different stakeholders must be fulfilled in each stage of the software development. The fulfillment of requirement is the purpose of development of software whether every part of the requirement has been traced during designing [3].

Requirement traceability helps software engineers for tracing the requirement from its starting point to its completion. [4] In the development of software part, traceability helps in various ways, such as change management, software maintenance, and prevention of confusions. [2] In the software maintenance and updating; the linkage of the traceability become outdated as the developers cannot adjust the characteristics of the system as per users' demand. Nasir Ali et al. [7] presented a variety of tractability revival approach.

There exist different parts which affect the traceability approaches' inputs. To recognize the requirement factors that affect on recovery of traceability link. O. C. Z. Gotel and A. C. W. Finkelstein [23] scrutinized and examined the life of the requests during development of system. Gethers in [5] developed the experiential discovery and offered an incorporated technique to merge orthogonal IR techniques. Hence, it has been mathematically discovered for generation of different outcomes.

Traceability requirement has achieved much detect over the past decade in the methodical literature. For recovering the linkage of traceability between complex documents, different authors suggested information retrieval (IR) techniques [6], [16], [17]. Stefan Winkler and Jens von Pilgrim [8] presented the capability to go behind the life of software part is interesting part of software developing for requirement engineering and model-driven developing.

Ebert and Winter [10] shows a broad analysis on traceability, concerning to the entire development software process. Depend on the updates; development of traceability is done consistent with specific manners associated to traceability. This concept helps to the meta models for storing of traced information in graph-based repositories. For the improvement of the requirement quality when tracing procedure is not dependent on verification and validation [14]. For addressing the requirements a prototype RETRO (REquirements TRacing On-target) tool is used, to study these requests and also show the results of the analysis [15]. Giuliano Antoniol, et al. [18] presented a method based on IR for recovering of traceability linkage with source code and requirements. The mnemonics for identifiers frequently confines the written code; which link complex conception with source codes.

A model of end user expectation in an electronic commerce vendor developed and tested. Trust allocates end user to reduce awareness of risks and ambiguity [19] [20]. Siobhan Clarke et al. [21] presented object-oriented design models have not been as much of useful during SDLC. Design models are frequently outsized and monolithic, and the design structure is typically relatively different from that of requirements.

Roel Wieringa [22] showed the maintenance of traceability in a development project with the related links between requirements and system during the process. Requirements vary from different stakeholders for designing and performance of techniques. For developing and implementing such technique must recognize the requests for tracing information.

## 3. SYSTEM ARCHITECTURE

**Problem Statement:** "To design a technique, for making certainty of the source codes and requests of a system are reliable with each other. Design a system that maintain requirement traceability links become absolute, to limit developers effort to update them. Enhance system using proposed technique and tools to recover the traceability links mechanically. "

In spite of the benefits traceability uses in the software engineering industry for the development of projects. The problems of the projects can be recognized with the cost of time and effort. The maintaining of traceability is difficult during the change of requirements and various point views on goals of different project stakeholders, management problems and politics.

Traceability helps in various ways for the software development, such as change management, maintenance of software, and avoidance of confusions. However, several challenges can be rises through management policy, quality of requirements; sustainability of traceability tool remains the open problem. Throughout software updating and maintaining; the linkage of the traceability becomes useless because the developers cannot modify the features of the source code.

1. ***System Design:***

Basically evolution task of any software, a developer must know the project landscape such as the structural design of system, performance, and the relations between the different parts with any available documentation. Program conception arises in a bottom to up way, a top to down way, or permutation both. Developers utilize dissimilar types of knowledge during designing of system, from general programming knowledge to domain-specific knowledge.

The definition of traceability requirement is as, "the capability to express and track the life of requirement, in cooperation with onwards and towards the back direction". The linking of traceability involving the system requirements and its related program codes are useful in decreasing conception effort. In software maintenance and evolution tasks the tracing information also useful. The linkage of traceability is also necessary to make sure that a source code is dependable with its requests and the developers have implemented that all individual requirements.

***2. Proposed System:***

The proposed work depends on the IR-based RTAs method is usually classified in three most important steps. Figure 2 shows the IR-based RT links revival process.

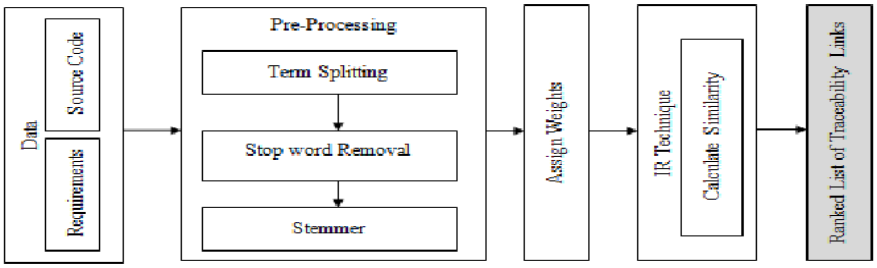


Figure 2: System block diagram

First, every the textual information with the requests and program code is taken out and pre-processed by term splits, removal of stop words which have less meaning and left over words are followed by stemming process for its origin word in the form of grammatical. The next, every the stemmed words are assigned weights by using a term weighting method. The IR technique calculates the likeness of requirements with source codes. After all, system creates a ranking confirmation of probable traceability linkage. An elevated comparison with two documents explains a probable similarity linkage with each other.

***2.1 Proposed Work:***

For creating traceability linkage, we eliminate all the words which have less meaning from program code and terms from requirements. In this, IR techniques are used for generation of traceability linkage between requests and related program codes. IR techniques imagine that all the requirements are in the text format. For removing the identifiers of source code, a source code parser is used, which throw-outs additional information. The removal of the identifiers and expressions is followed by a sorting stop word removal, and stemming procedure.

### *2.2 Steps in proposed system:*

The primary step is term splitting. This step eliminates non-textual portion, i.e., various numerical, arithmetical code, brackets, and additional blank spaces etc., as of the text. Several words might be merged through a specific symbol, i.e. underscore. Therefore, divide all the merged terms to make them separate.

a) The following step is the stop word removal. The normalized text is input for this step that could include some general words. The normalized texts with articles, punctuation, are measured as noisy text because it does not shows the similarity between requests and source codes.

b) The next step is stemming which would recognize the terms “exce”, “excellence ”, and/or “excellent” since developed from the origin word “excel ”. An IR system calculates the likeness among the documents depends on same data in both documents.

Still, by reason of dissimilar postfix, IR method would judge them, e.g., add and addition are as two different documents, and the result would be simple similarity between two documents. Consequently, it turns into significant for performing the morphological investigation to exchange plural words into singular words and to take origin form from infected forms.

## *3. Proposed Algorithms*

### *3.1 Data Pre-processing:*

The data pre-processing is prepared to eliminate needless content from the text and to find out the origin form of the words. The pre-processing of the data is completed by the valid method such as stop word removal and stemming to the data composed from the customer.

### *3.2 Stop Word Removal:*

For work out, stop words are words that are sorted out preceding to, or following, processing of document. In this step the recoveries of similarities between two documents there is need to recognize the keywords which have less meaning in documents. Stop words are ordinary words that take less significant meaning than keyword. To prevent indexing useless words, such as a, an, the, of, for, with and so on are stop words, although they may come into view normally. Stop lists may vary per document set.

Stop-word elimination is the method of eliminating these words. To find out the words from a text all needless content must be removed, so it is needed to remove the stop words from the text put into an array.

**Algorithm:**

The following is an algorithm for stop word removal
1 Acquire the input
2 Eliminate needless content from the text
3 Remove the irrelevant words from the text put into an array

4 Find out the origin form of the words
5 Store result in the String Builder
6 Loop during all words
7 Come again string with words detached

***3.3 Stemming**:*

Words get from the input of the data are create to be too sparse to be useful as features for categorization as they do not simplify well. The presence of a large number of inflections of the same word, this is the common reason for stemming. Hence, the origin form of the word is to be taking out as a feature.

A grouping of various terms can allocate the same stemmed term. A recovery system desires to recognize cluster of terms where the terms are clustered in small syntactic alternatives of each other and bring together only the general term stem per cluster. For example, the cluster of terms collecting collected, and collects, allocate a general stemmed term, collect, and can be displayed in the various incidence of the same word. For decreasing derived words to their origin form the stemming method is used. Stemming program is commonly known as stemming algorithms.

Even as writing the sentence for grammatical basis, it contains various forms of a word, for example, collect, collection, collecting and/or collected. In many circumstances, it would be helpful for a finding for one of these words to revisit the word in the set to take away the required content from a given sentence. The aim of stemming is to decrease variation form and occasionally derived associated forms of a word with a general root form. For example: cluster, clusters, clustering.

***Stemming algorithm***

The stemmer algorithm consists of the following steps:
1 Remove plurals and -ed or -ing suffixes.
2 Convert the letter y to i when there is another vowel in the stem.
3 Record double suffixes to single ones: -ization, -ational, etc.
4 Accords with suffixes, -full, -ness etc.
5 Remove -ant, -ence, etc.
6 Eliminate a final -e

Stemming algorithm consists of different steps of stemming applied sequentially. Within each stage, there are various principles to select rules, such as choosing the rule from every rule group that applies to the longest suffix. The algorithm of stemming works as follows:

Rules Illustrations
S → clusters → cluster
EED → EE agreed → agree
(*v*) ED → plastered → plaster
(*v*) ING → cutting → cut

There are three main reasons for stemming algorithm, or stemmer, as follows.

a) The first reason of a stemmer is to cluster the words according to their theme. Many words are the root from the same stem, and we can consider that they belong to the same concept (e.g., act, actor, action). The different forms are created by attaching affixes (prefixes, infixes, and/or suffixes) but, in English considering only suffixes, as normally prefixes and infixes change the meaning of the word, and bit of them would lead to errors of bad topic resolve

b) The next reason of a stemmer is openly associated to the IR process, as containing the stems of the words agree to some point of the IR process to be better, among which we can stress the ability to index the documents according to their theme, as their terms are clustered by stems (that are similar to concepts) or the expansion of a query to obtain more and more accurate results. The extension of the query permits it, for refining by replacing the terms; it covers with the related topics, which are also there in the collection.

c) Finally, the conflation of the words allocation the same stem leads to a decrease of the vocabulary to be taken into the process, as the entire terms contained in the natural input collection of documents can be decreased to a set of topics. This directs to a decrease of the space needed to store the formation used by an IR system and after that also lightens the computational weight of the system.

### *3.4 Assign Weights:*

An IR technique (IRT) changes every the stemmed words into vectors to calculate the likeliness among the words. To change stemmed words into vectors, each stemmed word is allocated a weight with a particular category. Widely used weighting systems are differentiated as probabilistic.

The term identifiers refer all source entities with term frequency (TF) for linkage of two documental terms. If a request comes out many times in a particular or several source codes, then IRT would suggested that source code is related to a request. Still, various terms do not demonstrate that it is the main term. Following two main factors are considered important:

a) **Term frequency (TF):** If a request occurs several times in a source code, then the term would be allocated high term frequency.

b) **Global frequency (GF):** If a particular term occurs in various source codes then the term is known as global. It is also known as inverse document frequency (IDF).

### *3.5 IR Techniques:*

For the identification of concepts in the source code with creation sets of traceability links, the IR technique is used; the recovering of traceability links by using different IR techniques.

**Precision and Recall:** For the accuracy of traceability links we apply two IR metrics, precision and recall, for the evaluation of research outcomes. The result values are measured in the range [0, 1]. Calculation the value of precision and recall for all the linkage of traceability are generated between requirements and source codes.

**Precision** is ratio of intersection of related documents and retrieved documents to the number of documents that are retrieved. If the retrieved documents are correct then precision value is 1.

$$\text{Preceision}=\frac{\left|\{\text{related documents}\}\cap\{\text{retrieved documents}\}\right|}{\left|\{\text{retrieved documents}\}\right|}$$

**Recall** is ratio of intersection of related documents and retrieved documents to the number of documents that should be related. If the all related documents have been retrieved then the recall value is 1

$$\text{Recall}=\frac{\left|\{\text{related documents}\}\cap\{\text{retrieved documents}\}\right|}{\left|\{\text{related documents}\}\right|}$$

An IR method frequently retrieved to exchange precision for recall or recall for precision. The generally used to exchange is the F-score, is the choral mean of recall and precision:

$$\text{F\_Score}=\frac{\text{recall}\times\text{precision}}{(\text{recall}+\text{precision})/2}$$

The choral mean discourages a system that sacrifices one measure for another too significantly.

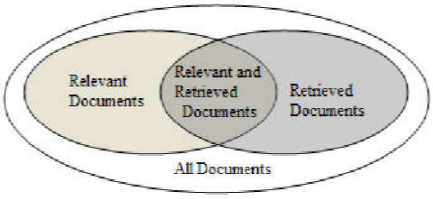


Figure 3: correlation among the set of retrieved and the set of relevant documents.

Similarity calculation of two documents: Similarity among a documents and requests depends on term frequency which shows incidence of similarity relative term in the frequency table. For measuring similarity between documents with various methods proposed metrics depends on similar term occurrences or document vectors. A proposed cosine similarity is described as below. Let v1 and v2 be two term vectors.

$$\text{sim}(v_1,v_2)=\frac{v_1.v_2}{|v_1||v_2|}$$

where the product $v_1.v_2$ is the standard vector scalar product which is described as $\sum_{i=1}^{t} v_{1i}.v_{2i}$ and the standard |v1| in the denominator is defined as $|v_1|=\sqrt{v_1.v_2}$

### *3.6 Ranking the traceability links' sets*

For the evaluation the efficiency of two RTAs, we produce different set of traceability links' with different threshold values. We have utilized the sets of traceability link to calculate precision and recall metrics values for the evaluation, this technique is efficient than many other.

## 4. HOW TO REPRESENT TRACEABILITY

Program conception occurs in a bottom-top way, a top-bottom way, or mixture them [4] [8]. Developers use knowledge throughout program comprehension, from domain-oriented knowledge to common programming knowledge. Traceability links between source code and sections of the documentation, e.g., requirements, aid both top-down and bottom-up comprehension [1]. Traceability links between the requirements of a system and its source code are helpful in reducing comprehension effort.

Requirement traceability is defined by [6] [23], "the capability to demonstrate and go after to the life of a requirement, in both onward and toward the back direction". This traceability information also helps in software maintenance and evolution tasks. For

traceability links, it is essential to represent them in a form that is suitable for its purpose [1]. The different ways (traceability matrices, graphical models, cross references) exist to represent traceability links, which are also supported by tools.

- **Traceability matrices:** Traceability links are represented in matrix form. The traceability matrix is the association between, horizontal and vertical dimensions are the. The values in the matrix stand for links between the artifacts in the matrix [21].
- **Graphical models:** Entity Relationship Model (ERM), various UML diagrams support the representation of traceability links embedded in the different development models [21].
- **Cross references:** Traceability associations between different parts are represented as links, pointers or annotations in the text [21].

## 5. EXPERIMENTAL RESULTS

The experimental outcomes of the work are given below. Here we have given the result of proposed work, which implements the data preprocessing with a stemming algorithm to identify the similarity between requirement and source code.

Our goal is to develop software which should be able to trace the source code according to requirement by applying steaming algorithm. Traceability links are useful in decreasing understanding effort linking the requirements of a system and its source code. The traceability information is also useful to the software maintenance and development tasks. For instance, with traceability links, a user is able to simply trace what software parts must be changed for the development of a new requirement.

| Test Cases | Current Technique (%) | Proposed Technique (%) |
|---|---|---|
| Test Cases 1 | 59.55 | 72.73 |
| Test Cases 2 | 42.28 | 63.64 |
| Test Cases 3 | 71.79 | 73 |
| Test Cases 4 | 71.89 | 73 |
| Test Cases 5 | 80 | 90 |
| Test Cases 6 | 79 | 80 |
| Test Cases 7 | 77 | 78 |

Table 1 Comparison for Precision of proposed Technique and Current Technique

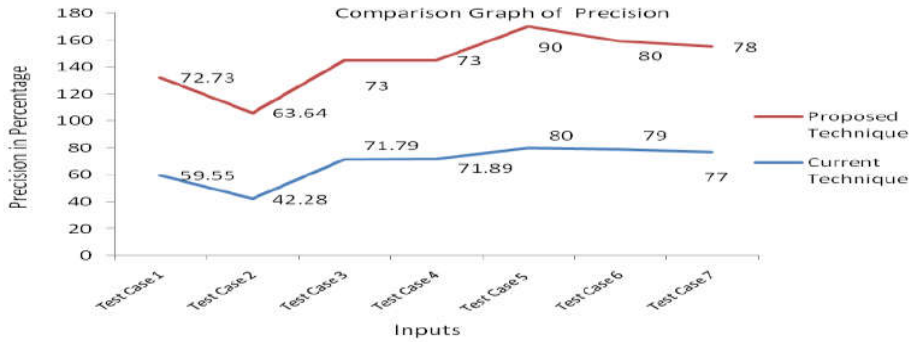


Fig 4 Graph Comparison for Precision of proposed Technique and Current Technique

The above graph in figure 4 shows comparison of precision in which the proposed technique performs higher than existing in the traceable links of IR.

| Test Cases | Current Technique (%) | Proposed Technique (%) |
|---|---|---|
| Test Cases 1 | 59.14 | 59.55 |
| Test Cases 2 | 52.46 | 53.28 |
| Test Cases 3 | 92.76 | 93 |
| Test Cases 4 | 93.73 | 93.80 |
| Test Cases 5 | 94.73 | 95 |
| Test Cases 6 | 82.60 | 83 |
| Test Cases 7 | 92 | 92.50 |

Table 2 Comparison for Recall of proposed Technique and Current Technique

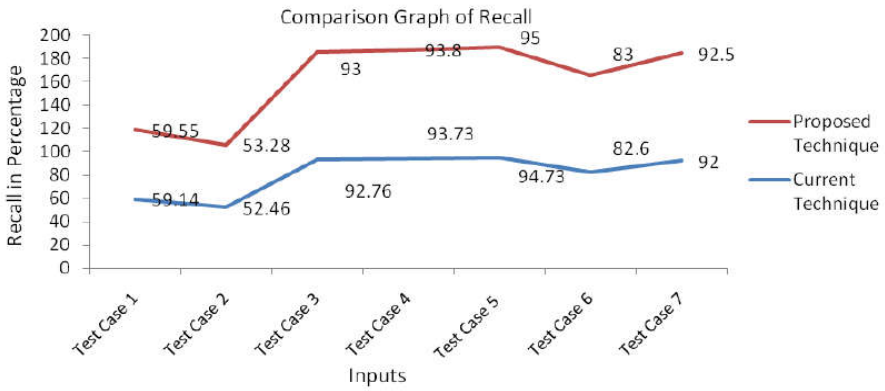


Fig 5 Graph Comparison for Precision of proposed Technique and Current Technique

The above graph in figure 5 shows comparison of recall in which the proposed technique performs higher than existing in the traceable links of IR.

## 6 FUTURE WORK

In future work, a broader survey, more diverse set of stakeholders for multiple domains. A different repositories and traceability recovery activities will be used for replicating the study of traceability. The overall goal is to enhance the state-of-the-art by addressing the shortcomings in current research that uses IR technique to mine vague software repositories and to persuade future researches to follow the lead.

Furthermore, a number of methods to get better the accuracy of the proposed traceability improvement technique. An appropriate cost-beneficial model for traceability needs further study as regards the costs of sustaining traceability and regarding for the benefits of traceability for other tasks in process of software development. This should be must focal point of the future work.

## 7. CONCLUSION

The traceability is most important factor and precious from different point of views for the development for any software project. We have developed a source code for the system that source code can be traced and become identical with the requirement analysis because we develop a source code as per the requirements.

From the results, it shows that our approach is performing better than other techniques. It provides better consistency compared to existing systems. Thus, this technique is useful for to minimize the execution time during the software development.

For development of any software, requirement traceability plays an vital role in the maintenance of software. Creating traceability links manually is one of the costly and lengthy works. The major finding of evaluation that is the proposed system generates accurate traceability links during development with high precision and recall. Requirements specification for requirements traceability is formed alongside all the investigations, which drives both their direction and focus.

# 8 REFERENCES


[1] Prof. Santaji K. Shinde, Mr. Vinayak M. Sale, "*A Survey on Accuracy of Requirement Traceability Links During Software Development*" Int'l Conf. on Innovations & Technological Developments in Computer, Electronics and Mechanical Engineering (ICITDCEME), 2015

[2] Divya K.S., Dr. R. Subha, Dr. S. Palaniswami "Similar Words Identification Using Naive and TF-IDF Method" *I.J. Information Technology and Computer Science,* 2014, 11, 42-47 Published Online October 2014 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijitcs.2014.11.06 Copyright © 2014 MECS *I.J. Information Technology and Computer Science,* 2014, 11, 42-47

[3] Prashant N. Khetade, Vinod V.Nayyar "Establishing a Traceability Links Between The Source Code And Requirement Analysis, A Survey on Traceability " Int'l Conf on Advances in Engg & Tech – 2014 (ICAET-2014) 66 | Page (IOSR-JCE) e-ISSN: 2278-0661, p-ISSN: 2278-8727 PP 66-70

[4] S. Muthamizharasi, J. Selvakumar, M.Rajaram "Advanced Matching Technique for Trustrace To Improve The Accuracy Of Requirement" Int'l Journal of Innovative Research in Science, Engg and Tech - (ICETS'14) Volume 3, Special Issue 1, February 2014

[5] N. Ali, Y.-G. Gue´he´neuc, and G. Antoniol, "Trustrace: Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links" IEEE Trans. Software Eng., vol. 39, no. 5, pp. 725-741, May 2013

[6] N. Ali, Y.-G. Gue´he´neuc, and G. Antoniol, "*Trust-Based Requirements Traceability*", Proc. 19th IEEE Int'l Conf. Program Comprehension, S.E. Sim and F. Ricca, eds., pp. 111-120, June 2011.

[7] N. Ali, Y.-G. Gue´he´neuc, and G. Antoniol, "*Factors Impacting the Inputs of Traceability Recovery Approaches"*, A. Zisman, J. Cleland-Huang, and O. Gotel, eds. Springer-Verlag, 2011.

[8] Winkler, S., & Pilgrim, J. A survey of traceability in requirements engineering and model-driven development. Software & Systems Modeling, vol. 9, issue 4, pp. 529-565 (2010)

[9] Schwarz, H., Ebert, J., and Winter, A. Graph-based traceability: a comprehensive approach. Software and Systems Modeling (2009) [10] J. H. Hayes, G. Antoniol, and Y.-G. Gue´he´neuc, "*PREREQIR: Recovering Pre-Requirements via Cluster Analysis*," Proc. 15th Working Conf. Reverse Eng., pp. 165-174, Oct. 2008.

[11] D. Poshyvanyk, Y.-G. Gue´he´neuc, A. Marcus, G. Antoniol, and V. Rajlich, "*Feature Location Using Probabilistic Ranking of Methods Based on Execution Scenarios and Information Retrieval*," IEEE Trans. Software Eng., vol. 33, no. 6, pp. 420-432, June 2007.

[12] Heindl, Matthias, and Stefan Biffl. A Case Study on Value-Based Requirements Tracing. Proc. of the 10th European Software Engineering Conference. Lisbon, Portugal, 2005: 60-69

[13] Lehman, M., Ramil, J. Software Evolution – Background, Theory, Practice Information Processing Letters, Vol. 88, Issues 1-2, October 2003, pages 33-44

[14] A. Marcus and J. I. Maletic, "Recovering documentation-to source-code traceability links using latent semantic indexing," in Proceedings of 25th International Conference on Software Engineering, 2003, pp. 125–135.

[15] von K nethen, A .Change-Oriented Requirements Traceability. Support for Evolution of Embedded Systems Proc. of International Conference on Software Maintenance, October 2002, pages 482-485

[16] Cleland-Huang, Jane, Carl K. Chang, and Yujia Ge. Supporting Event Based Traceability Through High-Level Recognition of Change Events. Proc. of the 26th Annual International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment. Oxford, England, 2002: 595-602.

[17] G. Antoniol, G. Canfora, G. Casazza, A.D. Lucia, and E. Merlo, "*Recovering Traceability Links between Code and Documentation*," IEEE Trans. Software Eng., vol. 28, no. 10, pp. 970-983, Oct. 2002.

[18] Ramesh, B., Jarke, M. Toward Reference Models for Requirements Traceability IEEE Transactions on Software Engineering, Vol. 27, No. 1, January 2001, pages 58-93

[19] Clarke, Siobhán, et al. Subject Oriented Design: Towards Improved Alignment of Requirements, Design, and Code. Proc. of the 1999 ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications. Dallas, TX: 325-329.

[20] Lindvall, M., Sandahl, K. How well do experienced software developers predict software change? The Journal of Systems and Software 43, 1998, pages 19-27

[21] Wieringa, R. An Introduction to Requirements Traceability. Technical Report IR-389, Faculty of Mathematics and Computer Science (1995)

[22] Gotel, O. & Finkelstein, A. An analysis of the requirements traceability problem. In Proceedings of the First Int'l Conf. on Requirements Engineering, pp. 94-101 (1994)

[23] Ramesh, B., Edwards, M. Issues in the development of a requirements traceability model. In Proceedings of the IEEE International Symposium on Requirements Engineering, pp. 256-259 (1993)

[24] Annibale Panichella, Collin McMillan, Evan Moritz, Davide Palmieri, "*When and How Using Structural Information to Improve IR-based Traceability Recovery*"