ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

Study on Requirement Engineering and Traceability Techniques in Software Artefacts

Swathine. K, Dr. N. Sumathi

Research Scholar, Department of Computer Science, S.N.R SONS College (Autonomous), Coimbatore, Tamil Nadu, India

Associate Professor and Head, Department of Information Technology, S.N.R SONS College (Autonomous), Coimbatore, Tamil Nadu, India

ABSTRACT: Software have many artefacts all are built to getcomplete working software system by prototype way. Multiple versions of artefacts are existing. All these versions are combining together to make a complete working product. Artefacts of the software are requirements specifications, design specification, source code, test case for verification and validation of the system, among all source code is high priority artefact. All these artefacts are by product of the software. Artefacts are continuously changing and updating regularly, which need proper management in order to software work properly. An artefact inconsistency is a major problem managing them is a challenging one. Maintaining traceability links between all artefacts and updating those artefacts are solution to inconsistency and changing requirements. Traceability links provide good understanding relationships between artefacts. The effective way of doing traceability links between requirements is with their priority. Traceability links are represented by traceability matrix. This article summarizes requirements engineering task, requirement management process, types of traceability and review on traceability.

KEYWORDS*:* Artefacts, Traceability links, Requirements Engineering, Requirements Managements.

I. INTRODUCTION

Requirements are put together processed to custom artefacts. Requirements are categorized into three types of user:user requirements, system requirements, and software specifications. They are always incomplete and inconsistent due to lack of knowledge.

Requirements are input to the system. Key characteristics are unambiguous, complete, modified, traceable, verifiable, consistent, correct, necessary, prioritized, and feasible.

Requirements give sense through consistent, traceable, and modifiable. They are valid and worth pursuing.
New requirements occur during the development of software, change occurs in business needs and for better understanding of the system.

Requirement change is caused by business change, technology change, environment change, stakeholder change; care should be taken for better understanding of the problem in order to get best solution. Requirements engineering is set of action concerned with identifying and communicating the purpose of software system and its use. Conflicts in requirements based on customer from business perspective. During development of software, business and technical environment get changed.

Requirements Engineering deals with customer to collect all requirements and handle changing requirements. Requirements engineering tasks are understanding customer requirements and their requests, analyzing the possibility of the requirements, negotiating the reasonable solution, specification of a clear-cut solution, manages all the

DOI: 10.15680/IJIRCCE.2017. 0501016

requirements, finally transforming the requirements into operational system. Requirement engineering task are: listed below and also shown in fig: 1

Inception: Specifying the beginning of the software project.Establish the basic understanding of the project.

Elicitation: Requirements are discovered.It'sa difficult taskbecause of difficulty in understanding the problem, communication between customer and project team, volatility of requirements, sometimes customer keep changing the requirements.

Elaboration: Information about the requirements are expanded and refined. Functions, features and constraints are proposed.

Negotiation: Customer may demand for more. Project teamnegotiate the reasonable solution.

Specification: Requirements are represented in writtendocument, mathematical or graphical model. Developing software requirements specification for requirements.

Validation: Requirements are analyzed, ensuring thatthey are unambiguously. Requirements validation mechanism is Formal Technical Review (FTR). Validation means are we building right product.

Requirements Management: It is the process of managingchanging requirements during the requirements engineering process and system development. It identifies, control, track requirements and make changes to the requirements as the project proceeds. The main goal of requirement management is to measure, handle inconsistency, and manage software development process, give traceability links and CASE tool supports. CASE tool support is required for requirement storage, change management and traceability management as shown in Fig: 2. Requirement management uses CASE Tool, it manages project requirements clearly, reduces risk, improves and develops software development process, helps to prioritize the requirements, trace relationship between requirements, and capture change history. Requirement management tool that meets customer needs, complete within a time, within a budget [14]. Refer fig: 3 for change management process.

Traceability deals with relationship among requirements, their source and system design. Bi-directional traceability, which deals with various associated requirements enables user to find origin of requirements and track every changes that are made to the requirements. For this, every change should be documented. Traceability information is represented by a data structure or requirement traceability matrix. In row-column cell, one requirement dependent on another requirement is denoted by **D** and weak relationship denoted by **R.**

Types of traceability

1. **Source traceability:** These are basically the linksform requirements to customer, stakeholder, who propose these requirements.

2. **Requirement traceability:** Links betweenrequirements to another requirement, between dependent requirements.
3. **Design traceability:** Links from requirements todesign and software architecture**.**

Types of Traceability tables

1. **Source traceability table:** Identifies source of eachrequirements.
2. **Dependency traceability table:** Shows dependentrelationship between requirements
3. **Subsystem traceability table:** Collect all requirements and categorizes requirements by subsystem, module

DOI: 10.15680/IJIRCCE.2017. 0501016

that they govern.

4. **Interface traceability table:** Shows howrequirements communicate to internal and external system.

Requirements Traceability: Capability to show, follow**the** life and expiry of the requirements in both forward and backward direction. Capture all levels of requirement from customer for project, to meet cent percent expectation and satisfaction. It is backbone of software development process and helps in effective, accurate delivery to the customer.

Uses of Requirements Traceability

Requirements traceability means tracking all requirements in Software Requirements Specification (SRS). It helps to confirm that all requirements are properly designed and well tested. Requirements support work product, so testing is very important.

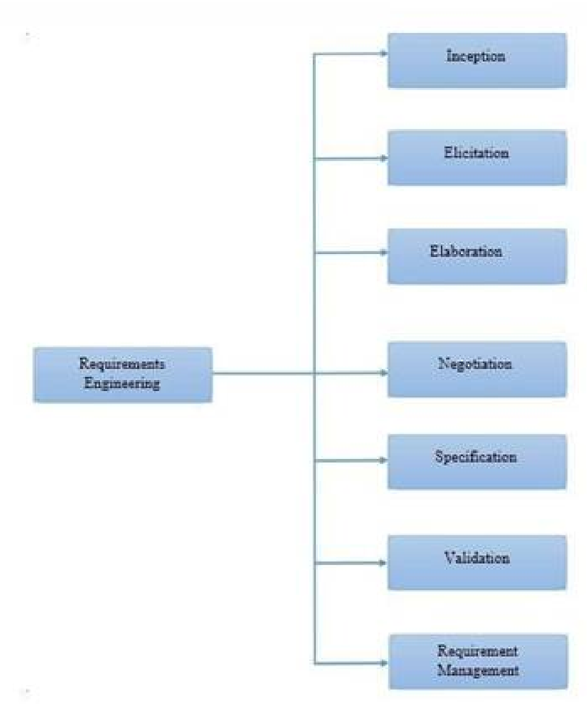

Fig 1: Requirement Engineering Task

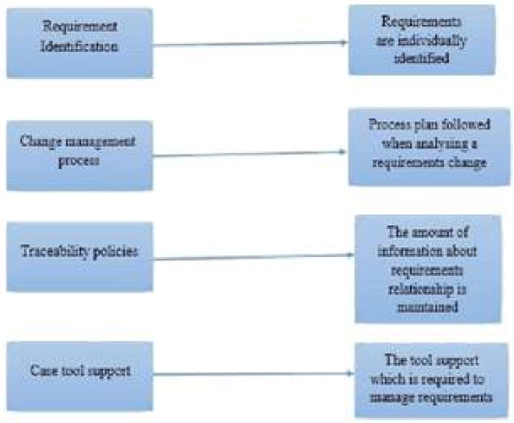
Requirement Identification
Requirements are individually identified
Change management process
Process plan followed when analysing a requirements change
Traceability policies
The amount of information about requirements relationship is maintained
Case tool support
The tool support which is required to manage requirements

Fig 2: Requirement Management process

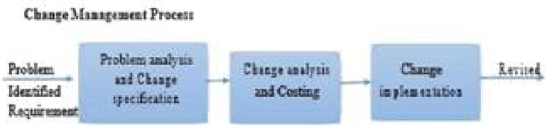
Change Management Process
Problem Identified Requirement
Problem analysis and Change specification
Change analysis and Costing
Change implementation
Revised

Fig 3: Change Management Process

II. LITERATURE REVIEW

Requirement Engineering is foremost phase of Software development process and it is most essential one. Because in this phase all basic and essential requirements are gathered through iterative, co-operative process on analyzing problem and customer needs. Requirement describes how software should work when it interact to the customer (clients). Requirement should be clear and give knowledge about domain and constraints of software. Wrong Requirement leads wastage of resource. Gathering requirement is a not easy task.

Stakeholder’s involvement should be high for successful requirement engineering task. But privation of requirement management skills lead to undesired requirement engineering. Once requirements are gathered that are represented into system design. System design that are diagrams such as Data Flow Diagram (DFD), Entity-Relationship (ER) Diagram, Use-case Diagram, State Transition Diagram (STD) used to gather requirement in detail. DFD is highlyrecommended by project team to understand the problem; it shows how data are processed to output from input given by customer. It also represents data flow from input to output, process, externalentity and data storage. Multiple level of DFD gives information in detail. ER diagram shows how system works by describing the logical structure. Entity is the information required from attributes that contains data. It shows relationships between entities. Use-case diagram shows boundary, external view of the system and how user interacts to the system. Mainly it is used to gather functional requirements. STD shows behavior of the system. It eliminates redundancy. Software Requirements Specification (SRS) describes internal and external behavior of the system that specifies both functional and non-functional requirements written by user or developer [1]. Managing those requirements is essential after requirement engineering which is handled by requirement management phase. Inconsistency occurs between requirements which are handled by traceability links among requirements.

Requirement management defines the success of the Software. Fail to manage requirements decrease the chance to meet user satisfaction and expectation. Common problem in requirement management is we cannot track changing requirement, it is difficult for the analyst to note down the requirements described by user. Project team failed to organize the requirements. Skills needed for requirement management are analyzing the problem,understand the purpose, need and cause of software to be built. Understand stakeholder needs requirements have many source they are customers, partners, end users, and domain experts are some high priority sources of requirements. Management, project team members, business policies, and regulatory agencies said as medium priority sources. We have to understand the customer needs on all perspectives. Define the system translating the requirements into meaningful form such as system architecture, design, specification and source code. The scope of the system software, should meet user expectation to fit according to the availability of time, people and money which are main resources for success. Refine the system definition includes developing and testing the software effectively and efficiently in any environment [2].

Tips for effective requirements management are prioritize requirement, prepare your requirement package, organize requirement review sessions, baseline your requirement, conduct impact analysis, trace your requirement, store requirement attributes, requirement versioning matters, requirement management tools help, maintain your requirement [5].For successful requirement management People, Process and Technology are three considerations to keep in mind. **Processes** guide through the development cycle. **People** need skills in technique and to know when, where andhow to apply. **Technology** enables system to work. This relationship is shown in fig 4.

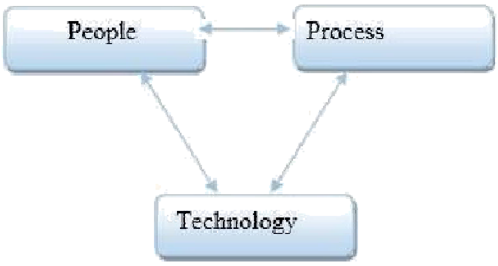

Fig 4: Development Cycle

To manage changing requirements, some changes in requirement are desirable. Activities involved are establishing the baseline, learning and following history of requirement, determining dependent requirement, tracing the relationship between requirements, maintaining version control. These skills are workflow of Requirement Management [2].

Jane Browning said all requirements are in text based format, managing these is called Doorstop. Hallmark of Doorstop is utilization existing version of the requirements, Doorstop is highly recommended to study the history of the requirements. Traditional requirement management tool hesitate to use Doorstop, until it provides multiple interface and formats [3].

Continuous and frequently changing requirements need proper requirements change management, so traceability is essential. **Scope, Coverage, and Analysis** is basis term fortraceability. Scope defines boundary for traceability, system boundary incorporates software and software requirements that derived from requirements. Coverage defines directional traceability. Four types of coverage are in traceability: They are 1) Origin-Requirements Traceability: defines source of the requirements 2) Requirements-Requirements Traceability: Includes four types of traceability i) Functional- Functional ii) Functional – Non-Functional iii) Non- Functional - Functional iv) Non-Functional - Non-Functional 3) Requirements-Other Artefacts Traceability: Traceability between different form of requirements that is requirement specification, design, implementation, test cases. 4) Other Artefacts - Other Artefacts Traceability: Provides links and dependencies in between artefacts other than requirements. Not all requirements satisfy these essential term [7].

Traceability links which are established by semi-automatic way, need visualization tool for proper requirements management in an incremental software development. Model driven development is used to produce artefacts, traceability links are defined manually which is labor-intensive and error-prone. For modelling and storing these artefacts relationship graph database approach is used. Treemap and hierarchical tree visualization techniques integrates overall overview of traceability. But hierarchical structure is not well communicated and not scalable. For evaluation process, tool processed against expected functionality of semi-automatic artefacts consistency management. Precision and recall describes the accuracy of traceability. It is more generic platform for artifacts without enforcing special methodologies [8].

Neo4j is a graph database that represents the link by node, node properties and edges, which translate original artefacts into XML format. Complete prototype will support all aspects of consistency management [9]. Visualizing the traceability is very important. Matrices, lists and trees are old visualization techniques. Sunburst visualization is a graph which nodes are in a radial layout, links are made through edges. Visualization is a radial space filling techniques for displaying tree like structure. In sunburst nodes are drawn on adjacent rings representing a tree structure. Sunburst shows overview of specification. Netmap visualization is segments of exactly one ring. Treemap is a space constrained visualization of hierarchical structure; it's very effective in showing attributes of leaf nodes using size and color coding [10].

Steven introduced CLIME tool that works incrementally which is responsible for detecting inconsistency without imposing particular methodology. CLIME designed to tell the developer when artefacts became unsynchronized, CLIME is constraint model, works only for build in constraints for visualizing the link[11]. Visualizing the links between software artefacts helps developer to recover, maintain and browse relationship effectively and efficiently. Treemap and hierarchical tree visualization techniques combine together to give graph based visualization techniques which reduces visual clutter, highly scalable and interactive. Graph based visualization represents artefacts as nodes and traceability links between artefacts as edges to form a graph. It shows overall overview of relationship [12]. Automated traceability facilities generate links between software artefacts without any error. However result contains both false and true links which are evaluated by an analyst [13].

Generating traceability for all functional and non-functional requirements is big task, selecting the requirements is very important, if the requirement is with priority plays important role in the selection of requirements to generate traceability links. Requirement prioritization is a process of arranging the requirements in an order; it is a part of requirement engineering and crucial in decision making activity. Some of prioritization techniques are Analytical Hierarchy process (AHP), Hierarchy AHP, Minimal Spanning Tree (MST), Cumulative Voting (CV),Hierarchical Cumulative Voting (HCV), Priority Groups (PG), Binary Priority List (BPL), Bubble sort (BS), Numerical Assignment (NA). Karlsonintroduced Cost-Value approach to help prioritizing requirements based on cost and valuesof the requirements. Smith proposed Value-oriented prioritization (VOP) framework targeting business values.Ramzan contributed value based intelligent requirement prioritization focus on multi aspects of requirements i.e. technical aspects andbusiness aspects [22].Traditional method of requirement priority is considering only static (stable) requirements that cannot give fully satisfied working software. Dunhuiyu focused on individual requirements characteristics and developed algorithm called Individual Functional Requirements Priority Ranking (IFRPR) which give priority to the requirements based on their type, role, time and frequency [24].

Richard R. Maiti focused on both functional requirements(FR) and non-functional requirements (NFR) because success of a system depends on both. He developed CEPP (Capturing, Eliciting, Predicting and Prioritizing) model for NFRs metadata.FRs state what system should do, describes input, behavior and output of the system.NFRs state how the system is supposed to achieve the behavior, NFRs is nothing but features of the system. Neglecting NFRs has negative impact on system. Requirements prioritization plays an important rolein requirements negotiation. Integrating NFRs in the early phase of software development can lead to high customer satisfaction and profit maximization [25].
Falak said success of software is based on selection of well prioritized requirements. Requirements prioritization plays an important role in requirements negotiation. During Requirements prioritization various factors influenced business and technical aspects they are risk, cost, customer expectations, complexity, time, dependencies, scalability, resources, speed, value, effort, type, role, size[26]. Consolidated result of literature survey is shown in table 1.

S.No	Authours	Year	Algorithm	Advantage	Disadvantage
1	Dhirendra Pandey &Vandana, Pandey	2012	Formal Requirement Engineering Process	Easily understandable	Making perfect SRS is impossible
2	Jace Browning, Robert Adams	2014	Lightweight developer friendly interface	Tight interaction with source code	Hesitate to use due to formats
3	Saeed Number and Mehdi Mirakhorli	2015	Strategic traceability patterns	Both forward and backward links for requirements	Redundancy occur in links
4	Paul Ralph and Rahul Mohanan	2015	Dichotomous Effects of Requirements Engineering	Reviewing the system is easy	Decoupling of requirements
5	K. Kamalabalan, T. Uruththirakodeeswaran, G. Thiyagalingam, D. B.	2015	Levenshtein distance	Handles inconsistency of requirements	More generic platform, Distribution development environment is minimal.
6	Ildiko Pete and DhariniBalasubramaniam	2015	Holistic framework	Graph based approach for visualization links	Lack of complete prototype support
7	Richard R. Maiti, Frank J.Mitropoulos	2015	Risk driven algorithm	Links for both functional and non-functional requirements	Not all NFR are gathered
8	Wentao Wang, Nan Niu, Hui Liu† and Yuting Wu	2015	Information retrieval and select trace algorithm	Color representation of links	Hard to retrieve the trace links
9	Falak Sher, Dayang N.A. Jawawi, Radziah Mohamad, Muhammad Imran Babar	2014	Review protocol	Concern about of aspects of the system	Results not are generalized
10	G. Spanoudakis, A. d'AvilaGarcez, A. Zisman	2014	Machine learning algorithm	Fellows rules for traceability links	Not all requirements are gathered

11	Thorsten Merten, Daniela Juppner, Alexander	2011	Sunburst and netmap	Uses colors for understanding	Not filtering the visualization links
12	Xiaofan Chen1, John Hosking, John Grundy	2012	Trace analyzer	Maintain inner relationship between requirements	Not clearly distinguish
13	Mahmoud KhraiweshandAsim El Sheikh	2009	Capability maturity model	Requirements are validate empirically	Suits for large long term projects
14	Muhamamd Farhan Bashir, Muhammad Abdul Qadir	2006	Traceability technique	Requirements scope, coverage, tool support, level of details, domain specific, and validation are viewed.	Cannot manage all the requirements
15	Steven P. Reiss	2006	CLIME	Requirements have constraints	Works only in build-in constraints
16	Jane Cleland-Huang and RafalHabrat	2007	Vector space and probabilistic network	Displays candidate links of requirements	Traceability links won't support speed and accuracy

Table 1: Literature survey

III. CONCLUSION

In this literature survey paper, we have discussed requirement engineering task, reasons for requirements inconsistency. For above mentioned problem author recommend requirements traceability matrix is developed with priority. Various authors' contribution towards Requirements inconsistency is also reviewed in literature review.

REFERENCES

[1] Dhirendra Pandey &Vandana, Pandey "Importance Of Requirement Management A Requirement Engineering Concern", International Journal of Research and Development a management review (IJRDMR), vol-1, pp no-66-69,2012.

[2] "Applying Requirements Management with Use Case",2000.

[3] Jace Browning, Robert Adams, "Doorstop: Text Based Requirements Management Using Version Control", Journal of Software Engineering and Applications, 2014, 7, 187-194, vol-7, pp no-187-194, 2014.

[4] Mahmoud KhraiweshandAsim El Sheikh"Empirical Validation of RequirementsManagement Measures", The International Arab Journal of Information Technology, vol-6, pp no 196-202, 2009.

[5] "10 Tips for Effective Requirement Management- Business Analyst"

[6] "Requirement Management Practice: the essence", 2004.

[7] Muhamamd Farhan Bashir, Muhammad Abdul Qadir,"Traceability Techniques: A Critical Study", IEEE, pp no- 265-267, 2006.

[8] K. Kamalabalan, T. Uruththirakodeeswaran, G. Thiyagalingam,D. B. Wijesinghe, I. Perera,D. Meedeniya, "Tool Support for Traceability of Software Artefacts", IEEE, pp no- 318-323, 2015.

[9] Ildiko Pete and DhariniBalasubramaniam, "Handling theDifferential Evolution of Software Artefacts: A Framework forConsistency Management" IEEE, pp no- 599-600,2015.

[10] Thorsten Merten, Daniela Juppner, Alexander, "ImprovedRepresentation of Traceability Links in Requirements Engineering Knowledge Using Sunburst and NetmapVisualization", ResearchGate 2011

[11] Steven P. Reiss, "Incremental Maintenance of Software Artifacts", IEEE computer society, pp no- 682-696 , Vol-32,2006.

[12] Xiaofan Chen1, John Hosking, John Grundy, "Visualizing Traceability Links between Source Code and Documentation", IEEE Symposium, pp no-119-125,2012.

[13] Jane Cleland-Huang and RafalHabrat, "Visual Support In Automated Tracing", Second International Workshop onRequirements Engineering Visualization (REV 2007) IEEE Computer Society, 2007.

[14] Dr. Jean-Claude Franchitti, "Software EngineeringG22.2440-001 Session 1 – Sub-Topic 1 Requirements Definition & Management Processes and Tools".

[15] Saeed Namber and Mehdi Mirakhorli, "Towards Actionable Architecture Traceability", 8th International Symposium on Software and System Traceability, pp no-36-42, 2015.

[16] Paul Ralph and Rahul Mohanani, "Is requirements Engineering Inherently Counterproductive?" IEEE/ACM 5th International workshop on the Twin Peaks of Requirements and Architecture, pp no-20-23, 2015

[17] Hatem Herchi and Wahiba Ben Abdessalem, "From user requirements to UML class diagram"

[18] Swathine.k and J.KomalaLakshmi, “A Survey of Requirement Engineering Practices in Software Development”, International Journal of Engineering And Computer Science, Vol 3,2014, pp no:8690-8694,2014.

[19] Steven P.Reiss,”Incremental Maintenance of Software Artifacts”, 21stIEEE International Conference on Software Maintenance,2005.

[20] Swathine.k and J.KomalaLakshmi,“Requirement Elicitation for Requirement in Software Engineering”, International Journal Of Engineering Sciences & Research Technology, pp no- 354-357,2014.

[21] Swathine.k and J.KomalaLakshmi “A Survey of Defect Prediction on Software Leads to Quality Product”, International Journal of ComputerScience & Engineering Technology,pp no-1038-1040, Vol-5,2014.

[22] Falaksher, Dayang N.A, Jawawi, Radziah Mohamad, Muhammad Imran Babar, “Multi-aspects Based Requirements Prioritization Techniques for Value-Based Software Development”, IEEE,2014.

[23] “Ten steps to effective Requirements Management”, IBM Omar Badreddin, Omar Badreddin, Timothy C. Lethbridge,”RequirementsTraceability: A model driven approach”, pp no-87-91,IEEE,2014.

[24] Software, Requirements definition and management, 2013.

[25] Dunhui Yu, Keqing He, Jian Wang, “An algorithm for priority ranking of individualized functional requirements in networked software”, IEEE,2008.

[26] Richard R. Maiti, Frank J.Mitropoulos, “Capturing, Eliciting, Predicting and Prioritizing (CEPP) Non-Functional Requirements Metadata during the Early Stages of Agile software Development”,IEEE,2015.

[27] Falak Sher, Dayang N.A.Jawawi, Radziah Mohamad, Muhammad Imran Babar, “Requirements Prioritization Techniques and different Aspects for Prioritization”, IEEE, 2014.

[28] Ritu , Dr. Nasib Singh Gill,” A Comparison among Various Techniques to Prioritize the Requirements”, PP no-601-607,ol no-12,2012.

[29] G. Spanoudakis, A. d’AvilaGarcez, A. Zisman,” Revising Rules to Capture Requirements Traceability Relations: A Machine Learning Approach”, IEEE, 2014.

[30] Wentao Wang, Nan Niu, Hui Liu† and Yuting Wu, “Tagged in Assisted Tracing”, IEEE, pp no-8-14, 2015.