A VISUAL APPROACH FOR REQUIREMENT TRACEABILITY

ABDULKADIR AHMAD MADAKI

UNIVERSITI SAINS MALAYSIA

2022

A VISUAL APPROACH FOR REQUIREMENT TRACEABILITY

by

ABDULKADIR AHMAD MADAKI

Thesis submitted in fulfilment of the requirements
for the degree of
Master of Science

December 2022

ACKNOWLEDGEMENT

First of all, I would like to thank the Almighty Allah for giving me the strength, patience and ability to complete this thesis. I want to express my deep sense of thanks to my supervisor **Associate Prof. Dr. WAN MOHD NAZMEE WAN ZAINON**, for his thoughtful comments, guidance, attention and encouragement. Furthermore, I would like to thank him for the invaluable advice, assistance and the idea of the thesis in the first place.

I want to extend my sincere appreciations to my friend Muhammad Jinjiri for the continuous aid, generosity, useful advice in every aspect of academic and otherwise. I want to extend my grateful appreciation to all who contributed directly or indirectly to the completion of this study.

I would also like to thank my father Alhaji Abdulkadir Madaki and my mother Hajiya Hauwa for their love, prayers, assistance and support throughout my entire life. Also, I would like to thank my brothers and sisters for their encouragement. Special thanks go to my beloved wife Zahra'u and my daughter Hajara for all support, patience, understanding and being helpful throughout my study. The successful completion of my work is the fruit of their sacrifices, devotion, and determination.

Finally, I would like to extend my special thanks to the Tertiary Education Trust Fund (TETFUND) for granting me a scholarship, and my profound thanks to the School of Computer Sciences, Universiti Sains Malaysia (USM).

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF APPENDICES

SATU PENDEKATAN VISUAL UNTUK KEBOLEHKESANAN KEPERLUAN

ABSTRAK

Kebolehkesanan keperluan ialah kaedah penting untuk mengesan dan mengenal pasti hayat keperluan dalam arah kehadapan dan kebelakang semasa kitaran hayat pembangunan perisian. Ia digunakan untuk menyokong analisis impak, perubahan keperluan, penyelenggaraan, pengesahan dan pengesahan sistem perisian. Visualisasi ialah salah satu kaedah perwakilan visual yang paling sesuai bagi data kebolehkesanan keperluan kerana ia mempunyai begitu banyak aspek yang boleh diterokai. Ia menawarkan simbol demografi yang terperinci dan boleh dilihat untuk menunjukkan kebolehkesanan hubungan antara artifak artifak keperluan. Walau bagaimanapun, memaparkan pautan kebolehkesanan yang banyak dengan berkesan dan cekap merupakan satu cabaran besar, kerana sistem perisian mempunyai sejumlah besar artifak dan pautan kebolehkesanan akan menimbulkan masalah skalabiliti dan kekusutan visual. Oleh itu, rangka kerja visual telah direka dan dilaksanakan sebagai alatan untuk menggambarkan dan mengurus kebolehkesanan hubungan antara artifak artifak keperluan. Rangka kerja ini mengikut panduan antara muka pengguna grafik lanjutan bagi capaian maklumat bisual yang memfokuskan pada gambaran keseluruhan dahulu, zum dan tapisan, kemudian butiran lanjut atas permintaan. Alatan yang dibangunkan ini menggunakan teknik visualisasi sebagai simbol berkod warna pada rajah pautan nod untuk mempersembahkan data. Pengguna boleh merentasi graf bagi mengenalpasti kaedah analisis impak untuk memahami data dan membuat keputusan semasa kitaran hayat pembangunan perisian. Hasil penilaian menunjukkan tindak balas yang positif dari segi faktor kebergunaan dan kemudahan penggunaan.

Purata skor min untuk kegunaan ialah 4.33 (86%), manakala purata skor min untuk kemudahan penggunaan ialah 4.25 (86%). Keputusan ini menunjukkan bahawa rangka kerja yang dibangunkan ini berguna dalam mengesan pautan antara artifak keperluan serta mudah digunakan kerana berkesan untuk meningkatkan interaksi pengguna. Pembangun projek perisian boleh menggunakan rangka kerja ini dalam kitaran hayat pembangunan perisian mereka untuk pemahaman yang lebih baik tentang hubungan antara artifak keperluan mereka.

A VISUAL APPROACH FOR REQUIREMENT TRACEBILITY

ABSTRACT

Requirement traceability is a significant method of tracing and identifying the life of requirement in forward and backward directions during software development lifecycle. It is used to support impact analysis, requirement change, maintenance, verification, and validation of a software system. Visualization is one of the most suitable visual representations of requirement traceability data as it has so many aspects that can be explored. It offers detail and visible demographic symbols to show the traceability of requirements artefacts relationships. However, displaying many traceability links effectively and efficiently is a big challenge, because a software system with large numbers of artefacts and traceability links quickly gives rise to scalability and visual clutter issues. Therefore, a visual framework is designed and implemented as a tool to visualize and manage the traceability of requirements artefacts relationships. The framework follows an advance graphical user interface guide for Visual Information-Seeking which focus on overview first, zoom and filter, then details-on-demand. The tool used visualization techniques as colour-coded symbols on a node-link diagram to present data. Users can traverse the graph for an impact analysis method to understand data and make decisions during the software development life cycle. The evaluation results show a positive respond in terms of usefulness and ease of use factors. The average score mean for usefulness are 4.33 (86%), whereas the average score mean for ease of use are 4.25 (85%). This results show that the framework is useful in tracing links between requirements artefacts, easy to use as is highly effective to improve user interaction. Software project developers

can use this framework in their software development life cycle for a better understanding of the relationships between their requirements artefacts.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Software development has many processes that result in different artefacts such as requirements specification, design, documentation, source code, test cases for verification, and system validation (Kamalabalan et al., 2015). It involved new system development, prototypes, frameworks, modification, maintenance, reuse, re-engineering, and other software components. The software development artefacts are found in any of its single-phase during development or between any software requirement. A single modification of an artefact can affect so many other artefacts in one or many phases, causing more complexity of relationships and get lost easily (Rodrigues et al., 2017). Thus, proper traceability relates to software development artefacts to improve stakeholder and system analysts' understanding.

However, requirement engineering plays a significant role as the most severe aspect of other software development processes. It interrelates requirements to develop a complete documented software system identified as the foundation for the entire development process (Abbas et al., 2019). It helps analysts and stakeholders understand the development and manage requirements validation to conform to their quality measures (Elamin & Osman, 2018). Requirement engineering consists of several processes/activities such as elicitation, documentation, validation, negotiation, and management. Part of requirement management is requirement traceability, known as the essential quality of a well-engineered software system.

Requirement traceability is the process of following and identifying the life of requirement in forward or backward directions during software development in an explicit way (Gotel et al., 2012). Centre of Excellence for Software and Systems Traceability (CoEST) defined it as the ability to associate any individual recognize artefact to any other artefact and maintain requirements (Palihawadana et al., 2017). Requirement traceability helps develop a software system quickly, increase software quality, and reduce maintenance efforts (Kang et al., 2016). Therefore, it is essential to follow software requirements to identify the affected artefacts and relationships through requirement traceability during the software development life cycle or change of requirement.

Requirement engineering faces significant challenges due to software projects' inevitable failure, such as poorly defined requirements, poorly managed projects, changes in business organizations, environmental changes, improper use of tools and technologies, modifications in legal rules, and many more factors affecting requirements constraints (Abbas et al., 2019). However, as a result, it is important to have an effective solution for the project's failure to prioritize requirements rigorously and the changes during the software development life cycle (SDLC). Moreover, requirement prioritization emphasizes artefacts value such as time and cost, reliability and ease of use (Sasank et al., 2019). After developing a software system, it is subjected to change at different priority levels due to maintenance, such as; improving customer needs, reducing risk, documentation of requirements, estimation of benefits, and prioritizing investments. The prioritization of requirements helps develop a better-quality software system that fulfils its functional domain needs.

Furthermore, this research study has designed and developed a framework that can visualize requirement traceability data. The framework follows the graphical user interface guide for Visual Information-Seeking Mantra "Overview first, zoom and filter, then details-on-demand," proposed by Schneiderman in 1996. The framework presents requirement traceability data as colour-coded symbols on a node-link diagram. Moreover, the result of effectiveness of the framework are useful and ease of use to improve user interaction and better understand requirement traceability data.

1.2 Requirement Traceability Overview

The term traceability is defined as the ability to follow the traces (to trace) to and from requirements. In the domain of requirement engineering, requirements traceability is the process to capture and follow the traces left by requirements in the phases of the software development life cycle and the traces left by those elements on requirements (Kesserwan & Al-Jaroodi, 2021). And by (Gotel et al., 2012) is the ability to follow the life of a requirement in forward and backward directions (i.e., from inception, through specification and development, to subsequent deployment, in addition to on-going refinement and iterations in any of the phases).

During software development, requirements are the specific needs of a project that can be used to unveil an existing problem facing a software development project. Thus, for developers to detect those needs for requirements is very hardly. Therefore, they used a set process to identify those needs, known as traceability that comprises a series of trace tools and techniques used to capture and show requirement and artefact relationships (Gazzawe et al., 2020). As a result, requirement and traceability is formed together to link requirements and related artefacts during software development or modification (Gazzawe et al., 2020).

In the software development, the foremost purpose of requirements traceability is used to ensure that the developed software system meets the user and stakeholder expectations. According to (Murtazina & Avdeenko, 2019), the purpose of tracing requirements in each phase of development was to show the users that changes in the requirements have been taken into account and the software system meets the objectives of the project, ensuring that the use cases, source codes, test cases fulfil the requirements to avoid implementation of unnecessary features that were no requests for. An important benefit of traceability is change management and change impact analysis that can retrieve all artefact relationships and identify the effect of a modification between development phases, requirement traceability contributes to develop a high quality and reliable software system. Moreover, traceability helps to estimate the required cost, time and resources to propagate modifications of software artefacts (Elamin & Osman, 2018).

The figure 1.1 below shows an example of graph traceability visualization technique used to represent software artefacts and their trace links for requirement, use case, source code and test cases. Edges (trace links) show relationships between artefacts and edge labels (trace link types) define the meanings of the relationships. (Elamin & Osman, 2018).

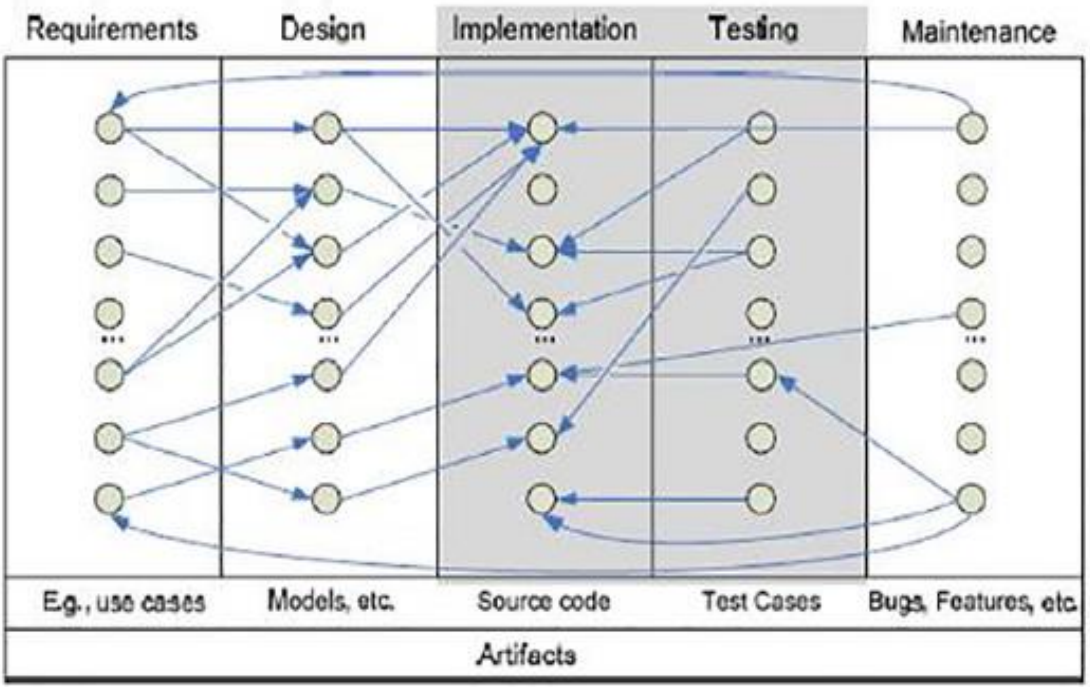


Figure 1.1 Example of graph traceability visualization (Swathine & Sumathi, 2021)

1.3 Motivation of Research

Requirements engineering frequently demands an intensive communication amongst analyst, stakeholders and end-users in order to discover and come to an agreement upon the needs for new system development or requirement change. Thus, visualization gives an excellent communicative value of visual representation to show the underlying evolution of software development projects. It converts natural language descriptions of each requirement into data to show all comprehension aspect of requirement life using colour-coded symbols for analytical understanding. This help explores all data attributes for its context that is rarely unrelated and positioned in multi-dimensional clusters.

Visualization is mainly used to support some parts of requirements engineering, more especially to show the structural organization of requirement traceability data. It depicts the hierarchical structure of requirement traceability data

by using demographic symbols representation to show requirements and artefacts relationships. Moreover, the visual representation of traceability data is more interactive rather than hand-drawn sketches on paper napkins via prototyping and scenario presentation tools. Therefore, visualization is used to analyse the traceability relationships of requirements during the software development process to improves stakeholders understanding and reach an agreement.

Furthermore, visualization increases general accessibility to explore and model data, then visualize the traceability of requirements and artefacts relationships in different levels of abstractions. It allows users to traverse the graph for impact analysis, make a change of procedure by ignoring non-relevant things and going in deep to get more information's (Ellis & Dix, 2007). Visualization is a great visual tool as it has more benefit over traditional techniques to visualize requirement traceability data for better understanding and communication.

1.4 Problem Statement

Software requirements change with timeline due to improvement of technology, change of environment, legal bodies rules and many more factors that may affect the change of software requirements (I. Rubasinghe et al., 2018). When a requirement being developed gets changed, the related artefacts are rapidly becoming inconsistent with one another, vulnerable to drift or erosion and lose their value for development (Meedeniya et al., 2019). These artefacts need to be appropriately related to their missing software requirements to show their relationships and propagate inconsistencies as traceability relates to requirements and artefacts relationships. Therefore, managing the software requirements change is preferred over building a new software system because it consumes resources, time, cost and effort (Jin, 2019).

Traceability links between artefacts are captured by a traceability recovery technique, a remaining key issue is how to represent the retrieved links to assist software engineers to understand, browse, and maintain them effectively and efficiently. Adopting software visualization techniques (e.g. tree-based, graph-based, or 3D-based approaches) is a common way to display retrieved links (Chen et al., 2018). However, displaying many traceability links effectively and efficiently is a big challenge, because a software system with large numbers of artefacts, and thus very large numbers of traceability links between artefacts, quickly gives rise to scalability and visual clutter issues (Aljawabrah, 2020). Moreover, the efficient visualization of both the structure of the traced system and the enormous number of links between artefacts is far from a trivial problem. Many traceability visualization techniques have been designed and developed to represent traceability links (Aljawabrah & Qusef, 2019). But have a limitation to explore multiple relationships between artefacts interactively to comprehend the overall structure of a system. Therefore, it is time-consuming and resource-intensive to explore and interpret the system (Aung et al., 2019). To date, no recent traceability visualization techniques can visualize many traceability links effectively and efficiently without scalability and visual clutter issues (Aung et al., 2020).

1.5 Research Questions

The aim of these research questions was upraised to identify traceability issues that the proposed framework will visualize to fulfil the research objectives which includes some questions.

1. How can traceability be properly visualized to help software developers in tracing the requirements? This question is used in order to design and

develop a framework that can visualize traceability data, and help monitor development progress, manage the traceability of requirements and artefacts.

2. What technique is suitable to be implemented in the visual approach that can reduce visual clutter? This question is used in order to study what filtering technique will be added to the framework to reduce visual clutter and enhanced the graph view.
3. How usefulness and ease of use of the framework help users understand requirement traceability data? This question is used to help derive out the result of effectiveness of the framework.

1.6 Research Objectives

The research aim is to enhance and propose a framework for visualizing requirement traceability data in order to achieve the following objectives.

1. To design and develop a framework that can visualize software requirement traceability data and highlight visual clutter.
2. To conduct a perception study in order to identify ease of use and usefulness of the framework.

1.7 Research Contributions

Data visualization and requirement traceability have been contributed and mainly been used within requirement engineering to support the latter phases of the development lifecycle. Requirement engineering is part of software engineering that regularly demands in-depth communication amongst stakeholders to discover and agree upon a new software development or change of requirement. Visualization

supports requirements engineering aspects to convey the structure of relations between an evolving set of software requirements and other artefacts to support the organization of software requirements and manage changes. It assists with requirements elicitation sessions and related analysis activities to model subsets of the software requirements or single properties for analytical purposes. Visualization offers a communicative value of good visual representation to show multi-dimensional requirement traceability data for a rapid comprehension of requirements artefacts relationships during the software development life cycle. It enables exploration and mapping of data into graphical notations to show the traceability relationships of requirements artefacts relationships.

As the potential need for traceability visualization approach, a visual framework for software requirement traceability is designed to help support the visualization of requirement artefacts relationships. The framework follows the graphical user interface guide for visual information-seeking mantra "Overview first, zoom and filter, then details-on-demand" proposed by Schneiderman in 1996. This framework will deliver maximum automation with a solution that can frequently handle visualization of software requirements traceability during software development life cycle. It also enables exploration requirement traceability data to visualize overview of requirements artefacts relationships. It is possible to help minimize ambiguities, improve understanding, develop product rapidly, increase software quality and reduce maintenance efforts.

Furthermore, the framework is developed by joining some visualization techniques and layouts as a node-link diagram to visualize the traceability of requirements artefacts relationships. The nodes represent requirements and artefacts as colour-coded circles, while the edges represent relationships as edges colour-coded

arrows. The framework also helps end users reduce visual clutter on the network to enhance the graph interactive and scalable at a different level of abstractions. It also enables change of procedure to preserve consistency between all requirements artefacts relationships on the graph. Users can traverse the graph for impact analysis method to understand data and further make decisions.

The framework was used in the evaluation as a tool to derive out the result of effectiveness and ideas provided by respondents with additional help to visualize and understand the traceability of requirements artefacts relationships better. Therefore, based on the evaluation result, this research has developed a framework that is ease of use for interaction and useful in real-time for manipulation to better understand the traceability of requirements artefacts relationships. The framework also used statistical analysis method to understand traceability data at a different level of abstraction explicitly. However, other similar domains that use tree visualization can apply many of these techniques.

1.8 Research Scope

This research study is used to generally understand requirement traceability and existing visualization techniques available to explore data visually. Visualization has been used to trace and relate requirements artefacts relationships, which has been chosen and appears to be suitable as a tool for presenting requirement traceability data. The datasets are inputted through the visual framework to visualize the requirement traceability data effectively and efficiently to the end-users.

1.9 Thesis Outline

This thesis contains six (6) chapters that are structured in consecutive order and organized as follows:

CHAPTER 2 described the relevant literature work on an overview of requirement engineering, overview of requirement traceability and types, and an overview of data visualization. Then a summary and comparison of existing requirement traceability visualization tools and techniques along with a discussion that suggests furthering this research study based on the gaps left behind.

CHAPTER 3 identified and discussed the research methodology used to conduct this research in three phases which provides a precise method to commence from the beginning to an end of the research study. This chapter illustrates and explains all the research procedures used to conduct this research study.

CHAPTER 4 explained about the framework of this research which is called VizTraceArtefacts. This chapter contains a full explanation about the design of the framework and descriptions of techniques and procedures used in each stage of design and development.

CHAPTER 5 presents the procedure and techniques used to evaluate the framework with the results of the effectiveness. It explained the experiments used to conduct the evaluation procedures that include the case study, sample selection and questionnaire design. This research study evaluated the effectiveness of VizTraceArtefacts based on ease of use and usefulness. The results of the analysis and findings of the evaluation are also derived out.

CHAPTER 6 presents the overall conclusion of this research study based on the findings. At last, the chapter provides future research work and recommendations for this research study.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Visualization is a significant feature that supports the exploration of complex and large datasets. This chapter discusses relevant literature. It then summarized and compared the existing visualization tools and techniques to reveal scalability and visual clutter issues. It consists of eight sections; Section 2.1 introduced the overall chapter work literature review. Section 2.2 gives a detailed overview of requirement engineering and its phases. Section 2.3 discusses requirement traceability in a forward direction, backward direction, and bi-directional traceability. Section 2.4 presents an overview of data visualization, where graph visualization and types were being discussed. Section 2.5 explained requirement traceability visualization. Section 2.6 discussed about tools and techniques to visualize requirement traceability data. Section 2.7 contains a discussion of the overall findings in the chapter. Moreover, the last section 2.8 concluded the research studies in this chapter.

2.2 Requirement Engineering Overview

Requirements engineering has been identified important aspects of newly developed or changed software requirements in the real world (Abbas et al., 2019). Requirements engineering is a software engineering sub-division that apprehensively link software requirements from specification to its development through relationships (Bano et al., 2018). Requirements engineering focused on identifying customer and stakeholder's needs for the system that will be developed or software maintenance. It has stages for each phase of development: elicitation, analysis, specification,

validation, and management (Bano et al., 2018). Each phase has an organized procedure that must be performed to collect information from users and perform requirement modelling, negotiation, validation, documentation, and maintain the software requirements (Sasank et al., 2019).

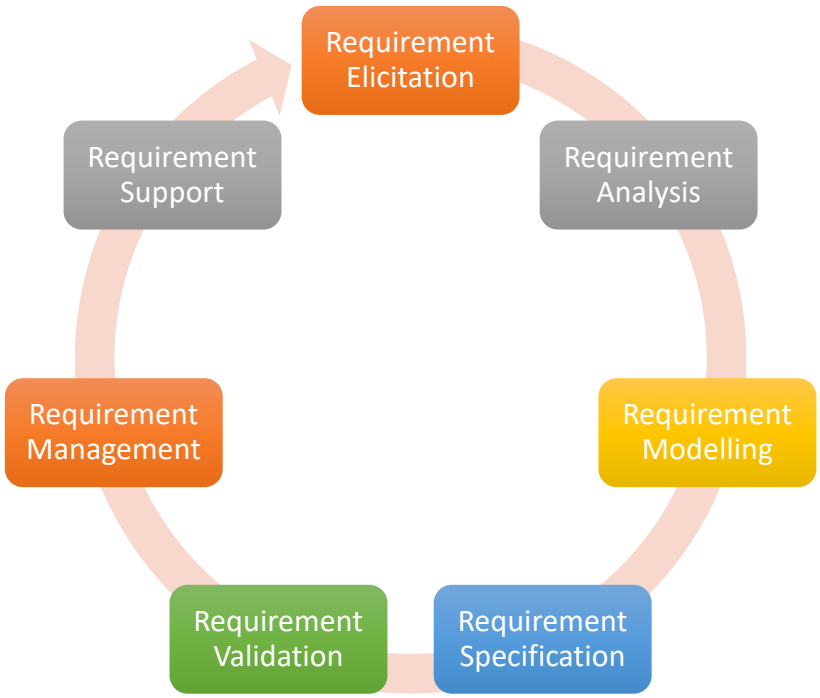


Figure 2.1 Requirement Engineering Phases (Abbas et al., 2019).

The process of requirement elicitation is also called requirement discovery sometimes, which means identifying, reviewing, documentation, and understanding the needs of users and stakeholders with limitations of system requirements (Yaseen & Farooq, 2018). Requirements specification is a process used to document the needs of stakeholders and some of the specific limitations (da Silva et al., 2018). It is needed to identify software application tasks and implementation processes in detail. Requirement validation is used to confirm completeness, correctness, consistency, cleanliness, feasibility, necessity, prioritization, unambiguous, and verify the system requirements (Nuzzo et al., 2018). The process used some procedures, such as

reviewing requirements, prototyping, and generating test cases. Requirements management manages and modifies requirements during software development processes (Wang et al., 2019). During system development and deployment, new requirements are developed and documented. Due to evolvement of time, some of the requirements missed or harm one another. Therefore, requirement management used three connected processes of; change management, requirement attributes, and requirement traceability to trace, monitor, and manage their advancement over time (Wang et al., 2019).

2.3 Requirement Traceability

Requirement traceability is the process of following and identifying the life of software requirements either in forward or backward directions to overcome the inconsistencies during software development lifecycle (Altaf et al., 2018). Changed between requirements occurs and needs to be traced accordingly to validate and verify the requirement sources and destinations. Therefore, requirement traceability provides a logical connection between software development artefacts to help maintain the traceability between unrelated and related artefacts in each phase of software development life cycle and gather requirements, design, development, testing, maintenance, and deployment. Requirements traceability is an important method considered as backbone support for projects development to guarantee and deliver a quality reliable product that meet stakeholder's expectations (Altaf et al., 2018). It helps to;

1. Verify and validate consistency, analyse and determine requirement conflicts.

2. Maintain a change request and trace back design.

3. Document activities for discovery of information easily and risks.

4. Discover software development process and review check.

2.3.1 Forward Traceability

This is the process of tracing requirement from original source to its specific product requirement in order to confirm the completeness of requirements and specification (Jin, 2019). The tracing process of a requirement starts from its specification development i.e. from requirement design, source code, test cases, to validation and maintenance. This process aim to confirm the implemented requirements to be tested and trace to their sources (Gotel et al., 2012). Forward traceability is sometimes called as Pre-requirement specification that happens before starting of requirement specification that is usually revealed with the activities of a requirements life earlier to its merging with product requirements to collect interrelated information of elicitation process (Nair et al., 2013).

2.3.2 Backward Traceability

This is the process of tracing each requirement or artefact that implement the requirement such as design, source code and test cases back to their related requirement or its original source of development phase (Jin, 2019). Backward traceability is sometimes called as Post-requirements specification, but if compared with pre-requirement it is interrelated to life of requirement activities and will end up with an insertion of requirement deployment (Kamalabalan et al., 2015). When its result to a poor requirements traceability, the problem is caused due to insufficient of full pre-requirement specification activities to collect full information that is link to its use after the process of elicitation has performed (Gotel et al., 2012).

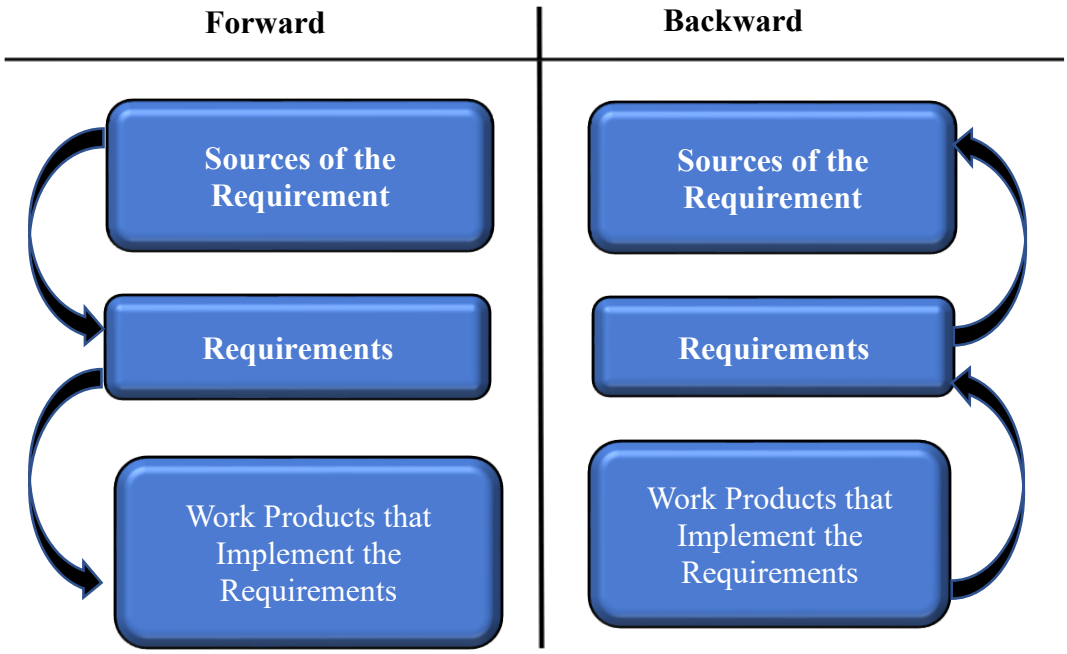


Figure 2.2 Requirement Traceability Processes of Forward and Backward Directions (Laghouaouta et al., 2013).

2.3.3 Bi-Directional Traceability

Bi-directional requirement traceability is the process that combines both the two directions at once i.e. (forward and backward directions) is popularly known as bi-directional traceability used to investigate influence of a modification to guarantee effects changed of requirements (Jin, 2019). The requirements affected by modification or any deficiency in a work product, there status is also considered as existing requirements and evidence of identification of misplaced requirements (Gotel et al., 2012).

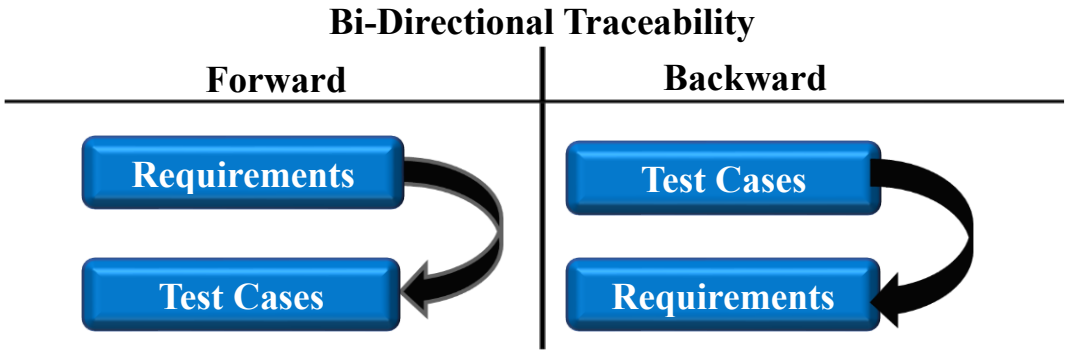


Figure 2.3 Bi-Directional Process of Requirement Traceability (Laghouaouta et al., 2013).

2.4 Introduction to Data Visualization

Visualization is defined as the "act of forming a mental vision, image, or picture of something not visible to the sight, or abstraction to make visible to the mind" (Ware, 2019). Visualization is a computing technique that aims to produce insight of a data. It converts complex data into graphical representations for interaction to maximize users understanding and sense its insight for communication (Cardno et al., 2018). Therefore, visualization is concerned with mechanisms within humans and computers to restructure numerical and symbolic data into visually represented forms and use the sensory information for a deeper understanding of physical representations.

Visualization process has a traditional data flow model steps that include the collection of data, data processing, data modelling, analyzing and interpretation of data (Ware, 2019). Moreover, each step in the visualization requires a design and transform it to a model as it is a process of "changing information into a visual representation and allow users to see the design patterns for communication" (Healy, 2018). The below figure 2.4 shows the process of converting computer representation into an image.

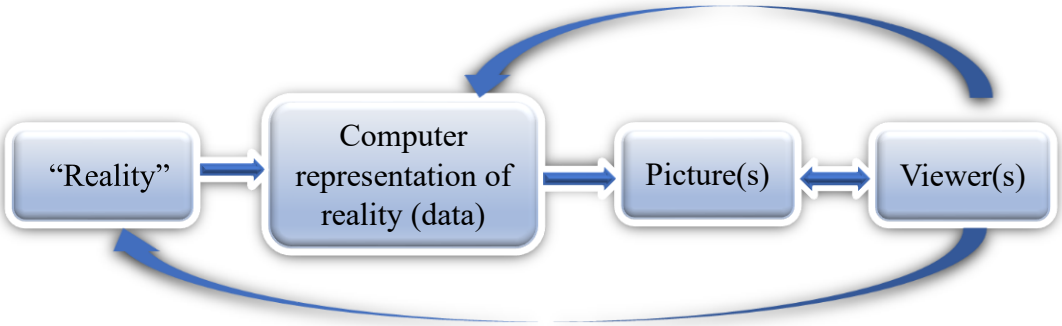


Figure 2.4 Process of Visualization Converting Reality Data using Computer Representation to an Image (Domik and Gatkauf, 1994).

2.5 Graph Visualization

Graph visualization is a type of that works with data to discover relationships between the data features (Chrapko & Chan, 2019). It has some regular entities called nodes and links between the entities called edges (Chrapko & Chan, 2019). In the world-wide-web domain, the nodes are representing classes, while the links are representing relationships (Nie et al., 2018). In the real-world application, a graph is popularly known as multiple class because it has a node with relationships between entities. The graph allows organization and discovery of a specific single task (Nie et al., 2018).

Moreover, multiple class graphs consider problems due to the complex number of nodes and edges on the network (Chrapko & Chan, 2019). This results in a visual clutter issue and affects the performance or the viewing platform. Graph visualization has some types of representation, such as social network and tree structure visualizations.

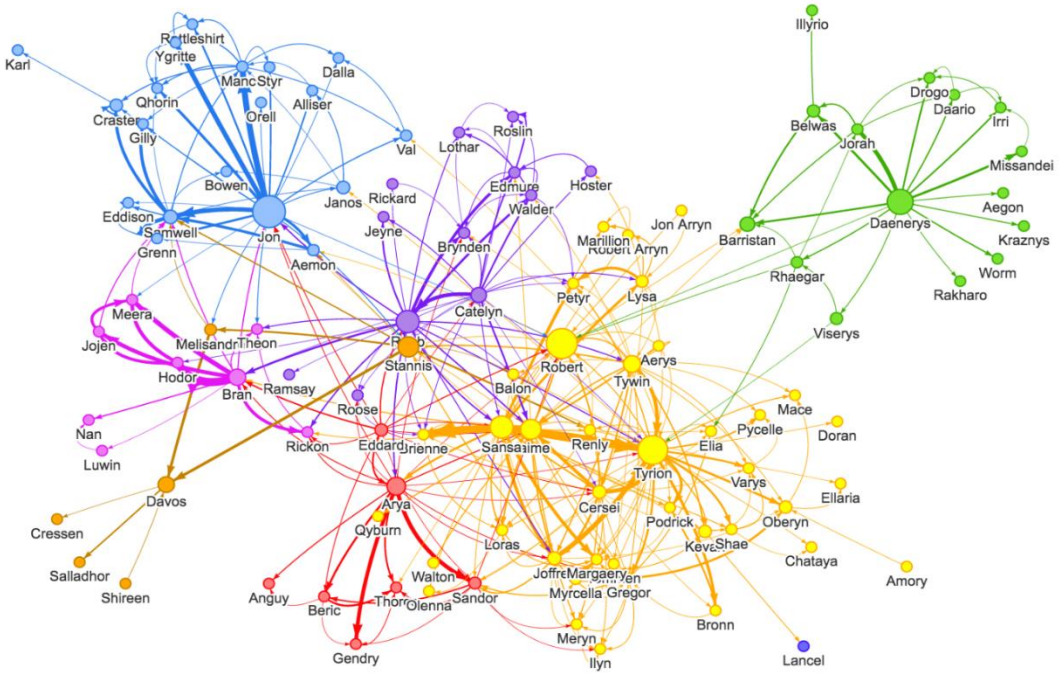


Figure 2.5 Graph Visualization Process (Viégas & Donath, 2004).

2.5.1 Tree Structure Visualization

Tree structure visualization is a demonstrative diagram that visually represents a hierarchy in a tree-structure form (Sevastjanova et al., 2018). Usually, a tree structure diagram has some elements such as a root node that has no parent or superior and is a member of the tree. Then the nodes that are linked and connected to the lines are called branches; they represent the relationships and connections between the tree members (Nie et al., 2018). In the end, the leaf nodes or ended nodes are also members that do not have child nodes or children. Examples of tree structure graph visualization include treemap, tree plus, and hyperbolic tree are the primary techniques appropriately for exploration of data and consider as contributors to the field of visualization.

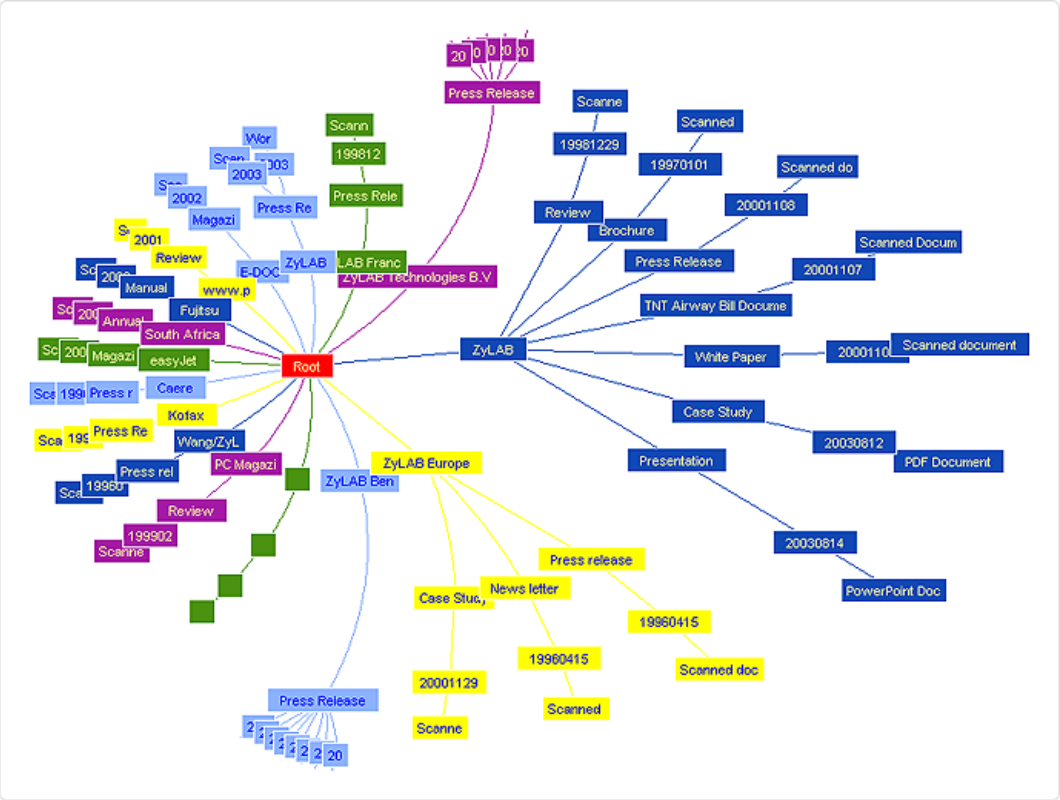


Figure 2.6 Example of Hyperbolic Graph Visualization (Zhao et al., 2018)

2.5.2 Social Network Visualization

Social network visualization is used to support a visual representation for a network organization and comparison (De Nooy et al., 2018). It is visualized in the form of a graph with entities as nodes and relationships as edges. A social network is formally represented as a tool by graph theory to compute structures in the form of a network (Farooq et al., 2018). Moreover, (De Nooy et al., 2018) states that social network analysis has been added as an essential method for identifying the significance of connections in a network. Social network analysis has some powerful applications that are developed to support an extensive network with complicated relationships. The social network analysis application types include Krackplots, Pajek and SocialAction are the foremost contributors to the field of social network visualization.

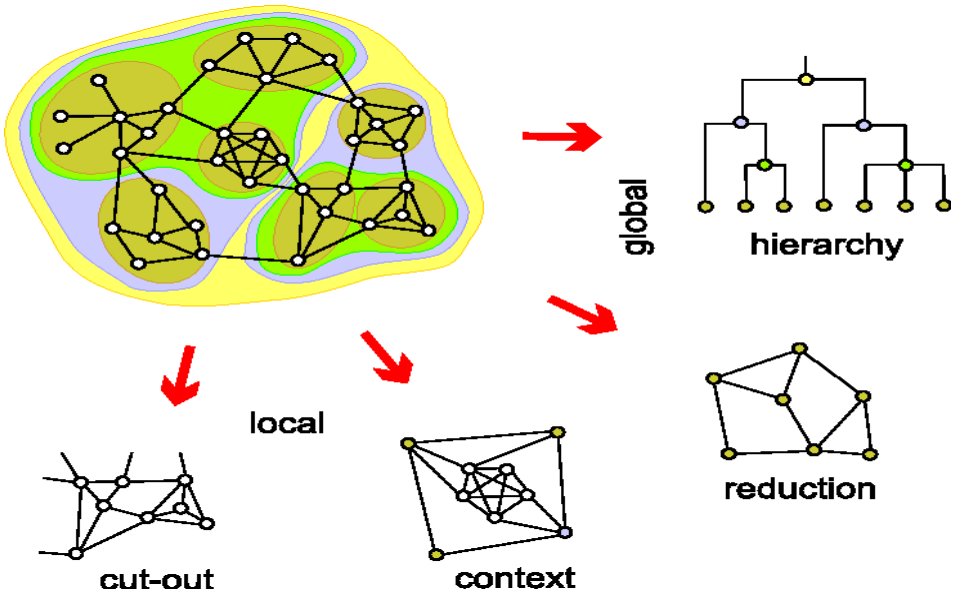


Figure 2.7 Process of Treating Large Networks (Borgatti, 2005).

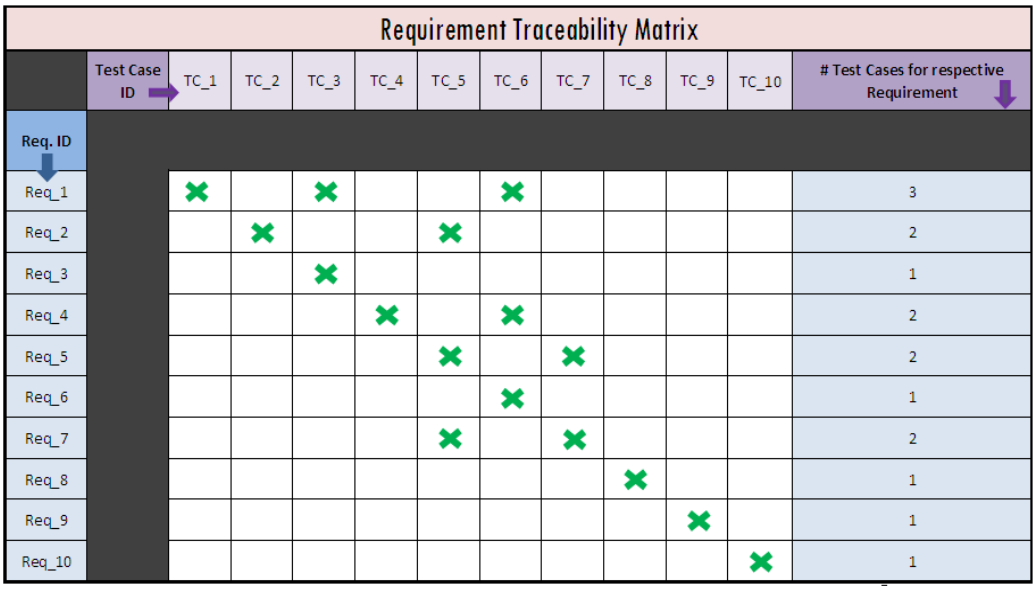
2.6 Requirement Traceability Visualization

Natural language descriptions of requirements are written and positioned in multi-dimensional clusters of data (Altaf et al., 2018). Therefore, stakeholders may need to understand the requirements during software development or those that needs to go for modification. However, the use of requirement traceability visualization helps system analyst and stakeholders to trace missing requirements and relate the relationships between requirements and artefacts. Thus, the visualization of many related artefacts and directions has some challenges in real-time that comprise of scalability and visual clutter issues (Beier & Müller, 2017).

Accordingly, data visualization has delivered requirement traceability with some tools and techniques for analyzing large datasets. Some of the techniques use the tabular form to show relationships, while graph and hierarchical tree techniques used diagrammatic representation for users to interact and understand the requirements during software development or change of requirement. The next section has discussed the techniques for requirement traceability visualization in detail.

2.6.1 Traceability Matrix

Traceability matrix is a traditional technique that is used to show requirements and artefacts relationships in a tabular form using two dimensions (Putro & Wibowo, 2018). It has some set of columns that show requirements while the rows show other artefacts and relationships. Traceability matrix is used in to trace the relationships captured between requirements and artefacts during the software development life cycle (Mezghani et al., 2019). However, using this technique can probably add up creation and maintenance cost of traceability as even the row-column arrangement is easy (Mezghani et al., 2019). Stakeholders understand this representation, but in terms of storage, it only has one repository which is not enough to trace forward and backward directions (Putro & Wibowo, 2018). Moreover, this technique is practically less due to its complexity of inaccessibility, search and update requirements and artefacts relationships.



Requirement Traceability Matrix

Req. ID \ Test Case ID	TC_1	TC_2	TC_3	TC_4	TC_5	TC_6	TC_7	TC_8	TC_9	TC_10	# Test Cases for respective Requirement
Req_1	✖		✖			✖					3
Req_2		✖			✖						2
Req_3			✖								1
Req_4				✖		✖					2
Req_5					✖		✖				2
Req_6						✖					1
Req_7					✖		✖				2
Req_8								✖			1
Req_9									✖		1
Req_10										✖	1

Figure 2.8 Example of Traceability Matrix (http://testingclub.blogspot.com/)

2.6.2 List Traceability

List traceability is a traditional technique used to show requirements and artefacts relationships in a single-entry tabular form (Putro & Wibowo, 2018). This technique capture information of requirement origin, destination and relationship with other requirements or artefacts during the software development life cycle. List traceability technique is suitable whenever a set of requirements or artefacts needs to be trace (Putro & Wibowo, 2018).

Requirement	Depends-on
R1	R3,R4
R2	R5,R6
R3	R4,R5
R4	R2
R5	R6

Figure 2.9 Example of Traceability List (Chen, 2010).

2.6.3 Cross-Reference Traceability

Cross-reference traceability is a traditional technique use to show each individual requirement with a list of artefacts relationships related to the requirement in a tubular form (Putro & Wibowo, 2018). This technique representation is straightforward for the user to understand requirement and artefact relationships during the software development life cycle. Though, this technique has a weakness to show a single relationship in the table if a particular requirement is missing and has a scalability issue to present a large number of requirements and artefacts relationships (Elamin & Osman, 2018).