Evaluating the Architecture Centric Traceability for Stakeholders (ACTS) Holistically

from the Economic, Social, and Technical Perspectives

Sherry Ramzy

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Computer Science and Software Engineering

University of Washington

2015

Committee:

Hazeline U. Asuncion

Munehiro Fukuda

Michael Stiber

Program Authorized to Offer Degree:

Computing & Software Systems

University of Washington

Abstract

Evaluating Architecture Centric Traceability for Stakeholders (ACTS) Holistically from the Economic, Social, and Technical Perspectives

Sherry Ramzy

Chair of the Supervisory Committee

Assistant Professor Hazeline Asuncion, Ph.D.

Computing & Software Systems

Software traceability has the capability of relating various artifacts produced throughout the development to assist with understanding the system from different viewpoints and abstraction levels. The Architecture Centric Traceability for Stakeholders (ACTS) technique was created to holistically address challenges from different perspectives that are associated with various stakeholders and artifacts. The challenges include time spent on, cost associated with, decreasing user's motivation, and the complexity in searching for related artifacts. ACTS is a stakeholder-centric technique, which is based on integration of hypermedia adapters of third-party software tools into a traceability framework. It uses rules to intelligently assign relationships and to filter noise in the captured traceability links. The purpose of this thesis is to evaluate ACTS from the economic, social, and technical perspectives with an emphasis on specific properties, which are capturability, accessibility, utilizability, affordability, and customizability. Furthermore, the research is focused on two specific stakeholders. The approach to evaluate ACTS is to design and implement two embedded single-case studies, where each case study focuses on one of the stakeholders. The first stakeholders are the users, who are divided into two groups based on their experience. The second stakeholders are the developers and are grouped as one set, but each developer focused on the development and integration of different third-part hypermedia

adapter. A comprehensive research methodology to design, administer, and analyze these case studies was developed. The users pointed out that ACTS facilitates the capturability, accessibility, and utilizabiliy of traceability links as well as it is time and cost efficient in finding related artifacts. It also enables interaction between users; however it needs improvements on the user friendliness and to reduce the noise in captured links when interactive rules are enabled. A conclusion based on paired sampled t-test statistical analysis found that there is no statistically significant difference between the feedback provided by users regardless of their traceability experience or the lack of it. It was also found that changing filtering rules functionality in ACTS yield no statistically significant difference in the users' perception of ACTS technique. From the other case study, a summative evaluation through comparing data between each developer was done. They reported that third-party tools APIs were easy to learn, enjoyed learning new technologies, and gained experience. Nevertheless, the developers indicated that integration of adapters in ACTS required significant time as well as better documentations is needed for ACTS. Based on the outcomes of the two case studies, it was found that ACTS positively affects the software development cycle from economic, social and technical perspectives with emphasis on affordability and customizability.

ACKNOWLEDGMENT

Foremost, I would like to express my sincere gratitude to my advisor Dr. Hazeline Asuncion for the continuous support of my master study and research, for her patience, motivation, enthusiasm, and immense knowledge. I could not have imagined having a better advisor and mentor for my Masters study.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Munehiro Fukuda and Dr. Micheal Stiber for their encouragement and insightful comments.

My sincere thanks go to Dr. George Youssef, Professor at California State University, Northridge for reviewing the statistics and for his useful comments and remarks through the learning process of this Thesis. His guidance helped me in all the time of research and writing of this thesis.

Many thanks go to Professor Laurie Anderson for teaching me technical writing. Her advices and support helped in expressing my ideas in this thesis.

A lot of thanks to Karen Potts for her help in installing and debugging ACTS. Numerous thanks to Hussein El-Saadi for his advices in fixing Java errors in ACTS. Also, I would like to thank Kanika Bhargava so much for her support and effort in finding participants with software traceability experience to volunteer in this study.

Thanks to all participants who volunteered in this study, this research would not have existed without your time, support, and effort.

Last but not the least; I would like to thank my parents Reda and Mary for supporting me throughout my life.

I would like to thank my loved ones, who have supported me throughout entire process, both by keeping me harmonious and helping me putting pieces together. I will be grateful forever for your love.

This work is based in part upon work supported by the US National Science Foundation under Grant No. CCF 1218266. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1: RESEARCH OVERVIEW

1.1 Thesis Sentence / Question

Architecture Centric Traceability for Stakeholders (ACTS) is a technique that is based on the integration of hypermedia adapters of third-party tools into a traceability framework by linking different artifacts, objects and functions and generating trace links. These links can later be used by users and developers alike. ACTS uses rules to intelligently assign traceability relationships and to filter noise in the captured traceability links. The purpose of this thesis is to investigate the following hypotheses:

Hypothesis 1: ACTS supports capturablity, accessibility, and utilizability.

Hypothesis 2: ACTS technique encourages developers to create affordable and customizable adapters to facilitate software traceability.

1.2 Introduction

Almost every new technological advancement is associated with software engineering, where the software development life cycle phases such as design, development, testing, maintenance, and evaluation are accomplished to produce applications [1]. The domain of software engineering extends over a wide range of engineering applications e.g., implementable biomedical devices, aerospace, web-based services, and mobile computing, etc. In general, software engineering is a multi-faceted discipline because it may include different programming languages, different requirements from perspectives of stakeholders or developers as various layers within the program may also require multiple programmers to collaborate simultaneously. The latter add complexity to the development process.

One of the most essential tasks in software engineering is efficient management of information, specifically in big-data processes as well as complex and large codes. Management Information Systems

(MIS) provide companies and organizations with a method to store and organize data into useful and well-structured databases. A suitable MIS can support management in the evaluation and assessment of formal performance as well as assets management and threat estimation [2 - 4]. Thus, MIS can be considered as a resource to improve business operations. The process of business operations starts with information gathering and subsequent analysis, which can be done electronically or manually. Data can be collected from internal sources, such as accounting, production, human resources; or from external sources such as the economic marketplace, competitors, national, or local organizations [2 - 4]. In addition to the improvement to the business operation, MIS is also important throughout the entire software development cycle because it provides a running log of all activities during the process, i.e., traceability.

Throughout the software development life cycle, various artifacts are produced that require software traceability. Traceability targets the identification of relationships between artifacts to make them more accessible and supportive for important tasks, like system comprehension, system analysis, system validation, and software maintenance [2]. In system comprehension, traceability helps stakeholders to understand the scope of the code [2]. It also help in connecting different artifacts, such as the steps of logic development, the bug reports, and fixes. In system analysis, traceability couples the artifacts, e.g., it connects bugs to its resolution. Traceability connects test cases with requirement documents and source code in system validation. Finally, it links communications between users and developers to the bug reports to ensure proper software maintenance; hence, reducing unexpected down-time [2].

Generally, software traceability is a way of recognizing related information distributed across heterogeneous artifacts created throughout the software development lifecycle. These artifacts could be related to information management, e-Science, software acquisition, or medical health records. A major drawback from tracing heterogeneous artifacts is that the associated work is generally laborious.

For example, manually checking candidate traceability links produced by a traceability tool can be a time-consuming process. Other drawbacks could be losing track of artifacts that are connected to each other due to the inability of the traceability tool in accurately capturing all links. Therefore, a careful and in depth evaluation of traceability tools and techniques are required [2, 5, 6].

The impact of traceability on the software development cycle can be assessed and evaluated by considering: 1) the feasibility of the traceability effort to ensure successful software development; and 2) the ability of the software to fulfill the customer's need as set forth in the requirements document. Thus, the traceability tool should be considered based on the interaction between all aspects of development. These aspects include definition of requirement document based on the need, selection of development architecture, development language, and algorithms, the process of software integration, quality assurance, and maintenance. Of course, as the development effort expands beyond the aforementioned aspects, the traceability tool must be able to evolve quickly and easily, especially when other third-party tools are added to the requirements. These aspects can be encompassed into three essential perspectives: economic, social, and technical. The perspectives are essential because they can facilitate the understanding of the traceability challenges from small to large-scale settings. These challenges affect interrelationships difficulty between artifacts, heterogeneity of artifacts, and different satisfaction factors of stakeholders. That is, the traceability impacts, i.e., economic, social, and technical, can be mapped directly into all aspects of development.

1.3 Solving Economic, Social, and Technical Perspective Issues

As mentioned above, a traceability technique must be evaluated based on its impacts on the software development cycle from economic, social, and technical perspectives. These perspectives are affected by several factors [7, 8]. The economic perspective is an important variable for accepting or refusing a traceability solution in an industrial software development process. The economic perspective is affected mainly by the cost and time factors. The cost is usually dependent on the funds associated with off-the-shelf (OTS) or developing a traceability tool, while the time is related, for instant, to the training hours spent. Generally, the software traceability is known for its expensive overhead costs [8].

The social perspective is concerned with the interaction of different stakeholders and the traceability tool. It is also associated with the method of integration of software traceability in an organization. Primarily, the human factor affects the quality of traces [9]. For example, the stakeholders are able to recognize whether the traceability links were accurate or not. Therefore, different roles in a company can agree or disagree on links traced [10]. This is the essence of the social challenge in implementing overall traceability tool that satisfies all stakeholders [8, 11, 12].

Finally, the technical perspective is concerned with the complexity of tracing, defining, creating, retrieving, and maintaining traceability links on different artifacts that are scattered in various levels of abstraction and types of relationships across numerous software granularity levels. Techniques to solve technical issues include natural language processing, open hypermedia adapters, data mining, and machine learning techniques [5, 8, 11 - 13].

Therefore, the decision on adoption of a software traceability tool can be based on evaluating the tool from the economic, social, and technical perspectives with an emphasis on specific properties. These properties are capturability, accessibility, utilizability, affordability, and customizability [11]. It is then the focus of this research to assist in evaluating a traceability technique based on aforementioned perspectives and its associated properties.

1.4 Motivation

The importance of software traceability is evident as it is required by many government agencies and professional standards. It also plays a vital factor for the success of large-scale software development [2, 5, 6]. Figure 1.1 shows that software traceability is an essential variable in the process of software systems development as it can act as a bridge between development and deployment. It is plotted between two consecutive points to indicate a relationship while connecting different artifacts in significant way to the software life cycle. Software traceability has the capability of relating various artifacts produced throughout software development to assist with understanding the system from different viewpoints and conceptual levels. It supports different activities from the time of development to deployment. The activities include analysis, design, refactoring, testing, and maintenance. Thus, the ability to reach software traceability depends on the establishment of navigable links between artifacts that are disconnected. The artifacts are created in testing, evaluation, and reporting phases during software development life cycle. Examples of artifacts include source code, design and requirement documents, test cases, and bug reports [2, 5-6, 14-15].

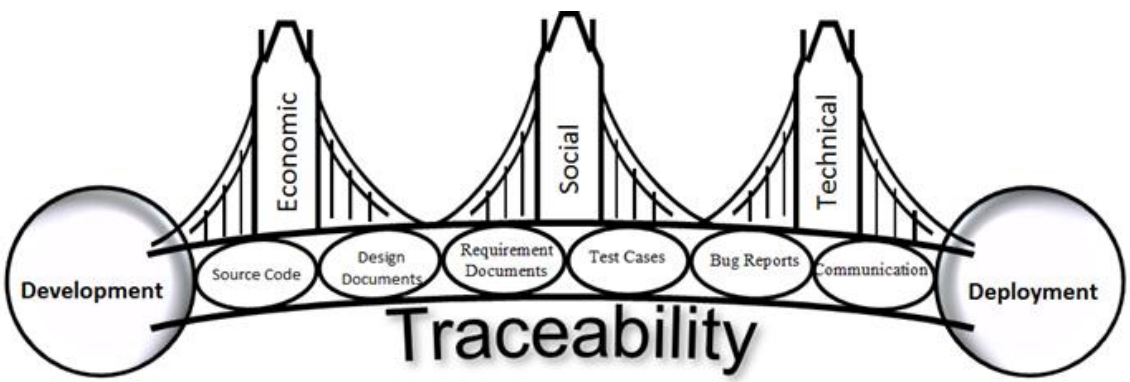


Figure 1.1: The traceability bridge between development and deployment.

Despite the importance of traceability and the extensive research documented in the literature thus far, there is need to formalize the evaluation process of traceability tools and the underlying technique. The need is based on three reasons. First, the cost and time associated with the development of software tools are on the rise as labor cost increases, i.e. economic. Second, the spread of globalization of software development – development team located in different countries – necessitate continuous interaction between different stakeholders at levels, i.e. social. Third, the number of artifacts in a given software tool can be very high especially in big-data projects or sophisticated software tools, i.e. technical. That is, the need rests on the economic, social and technical impacts, which requires a formal evaluation process to weigh the available traceability techniques and tools. Of specific interest to this research is ACTS technique as a traceability tool; the details of which will be discussed in Chapter 2. We selected ACTS because it is currently the only traceability tool that aims to address the traceability challenges from the economic, technical, and social perspectives.

1.5 Research Goal

The primary goal of this research is to develop two case studies to evaluate the hypotheses stated in Section 1.1. In essence, the research is focusing on determining the effectiveness of ACTS in addressing traceability challenges of capturing and categorizing links from the economic, social, and technical perspectives. Evaluating ACTS from these three perspectives helps in understanding the reason behind the difficulty of achieving software traceability in practice due to factors such as an overhead costs, heterogeneous artifacts and tools, and user's motivation as discussed in the previous section [5, 11].

The approach to accomplish this goal is to develop survey instruments to qualitatively and quantitatively measure the perception of different stakeholders. Specifically, participants in this research include users with no prior traceability experience, users with prior traceability experience, and developers of hypermedia adaptors. Each users group will be further divided into subgroups to evaluate different noise filtering schema, i.e., Background Rules vs. Interactive Rules.

In the first case study, end-users will be given a brief introduction to ACTS and its functionalities. Then, they will be required to provide information about their technical background, respond to quantitative questions, and answer open-ended qualitative questions [16]. Within each users group, the quantitative answer will be analyzed using statistical tools such as paired samples t-test, while qualitative questions will be reasoned, categorized, and reported. A comparative analysis is performed to evaluate the effect of users' experience on the feedback. The outcome of the first case study is directly mapped to Hypothesis 1 stated above in Section 1.1.

The second case study is developed to evaluate the facilitation of development and integration of hypermedia adapters for third-party tools that are not currently supported by ACTS. In this study, only developers were asked to respond to a survey instrument, which is different than the one used before in users' case study. It includes questions about the developers' software development background as well as quantitative and qualitative questions about their experience in developing and integrating adaptors for Google Chrome, Skype, Windows Media Player, Microsoft Visio, and updating Mozilla Firefox. The outcome of the second case study is directly mapped to Hypothesis 2 stated above in Section 1.1.

1.6 Research Contribution

The contribution of this research is to empirically determine whether a traceability technique that centers traceability links on the architecture and caters to different stakeholders addresses challenges from the economic, technical, and social perspectives. This research will also provide insights into how the technique can be used in software development settings.

The remaining chapters of the thesis are organized as follows. A comprehensive literature review is included in Chapter 2, in which different traceability techniques are discussed as well as their economic, social and, technical impact. A brief background about ACTS is also presented. In Chapter 3, the research methodology is discussed in details, which include the survey questions for both case studies, i.e., users and developers. The results are analyzed and discussed in Chapter 4. Thereafter, Chapter 5 is where conclusion and future work are presented.

CHAPTER 2: BACKGROUND AND RELATED WORK

Generally, software traceability case studies can be divided into two types: qualitative and quantitative. Some studies try to study the effect of traceability tools from all perspectives on the software development cycle. Others focus on a more refined look at investigating the economic, social, or technical perspectives.

This chapter includes a brief background about ACTS; then qualitative studies are presented and discussed from economic, social, and technical viewpoints. A similar literature review is provided on quantitative studies. Finally, the last section is about historic perspectives on case studies.

2.1 Industrial Tools

The importance of software traceability has motivated academic researchers as well as industrial companies to develop tools. From the industry side, IBM developed Rational RequisitePro, Rational Dynamic Object-Oriented Requirements System (DOORS), and DOORS Next Generation (DNG) [12]. These tools support industries with a focus on improving the software and systems lifecycle. Rational RequisitePro is existing customer base legacy requirements management software. IBM does not invest in adding new features to this tool because it is not aligned with a collaborative life cycle management strategy [11]. Another IBM tool is DOORS, which used to be a market leader in Telelogic acquisition; it is very feature-wealthy and sophisticated tool. It suits industries with tens of thousands of requirements with complex relations such as automobile, aerospace, etc. [12]. An additional IBM tool is DOORS Next Gen (DNG). DNG is a merger between Rational Requirements Composer (RRC), a tool that is built by IBM based on Jazz platform, and DOORS. DNG is a web-based tool part of the CLM portfolio (Collaborative Life Cycle Management) and is IBM's strategic requirements management web based tool for DevOps. IBM is investing in DNG as its primary requirements management tool [12]. Therefore, as the company business model change based in increase on trade activities, management must decide to upgrade the traceability system to accommodate the expansion. However, the features in DOORS are more advanced than the other two tools; the financial and social burden to replace a company-wide used tool may be a deterrent for change. In turn, this negatively affect the social and economic perspectives of traceability. Finally, it is unclear from the information available on these tools how flexible they are with third-party integration, i.e., potential technical shortcoming.

Another commercially available software traceability tool is dotTrace [17]. It offers various tracing modes to help in detecting bottleneck in a variety of .NET applications. In addition, it controls and collects data, navigates to source code, and compares traceability results [17]. dotTrace is limited to .NET applications and thus is useful for developers, but its use may be limited to users. That is, the tool can positively impact the technical perspective of traceability but it fails short in providing a feasible company-wide solution for all stakeholders.

Additionally, Team Foundation Server (TFS) is collaboration software for Microsoft application lifecycle management, which supports agile development [18]. It is a team-based collaborative tool; it is concerned with the social impact of traceability. Similar to dotTrace, this tool can be only used with Visual Studio and can only be used during the development. It helps the team to compare application development throughout the lifecycle to the company's best practice. This tool is not an open source and it does not easily support third-party tools.

In the academia, the focus was on development of traceability frameworks in addition to testing tools. As mentioned in Chapter 1, ACTS is investigated as part of this research, which was developed by Asuncion et al. [8].

2.2 Background on ACTS

In this study we evaluated ACTS which was created to holistically address challenges from the economic, social, and technical perspectives [2, 7, 8]. ACTS can be adapted to different software development lifecycles, including an iterative or agile lifecycles [2, 7, 8]. As shown in Figure 2.1, ACTS is built on top of ArchStudio 4, an architecture-centric development environment that was first created at University of California Irvine by the Institute for Software Research [19]. Archstudio 4 can create system architectures as well as model and visualize integrated tools. The ACTS technique supports architecture-centric development by enabling developers to connect related and heterogeneous artifacts to the architecture. The technique allows stakeholders to trace artifacts that are important to them, while performing their development work to capture traceability links [2].

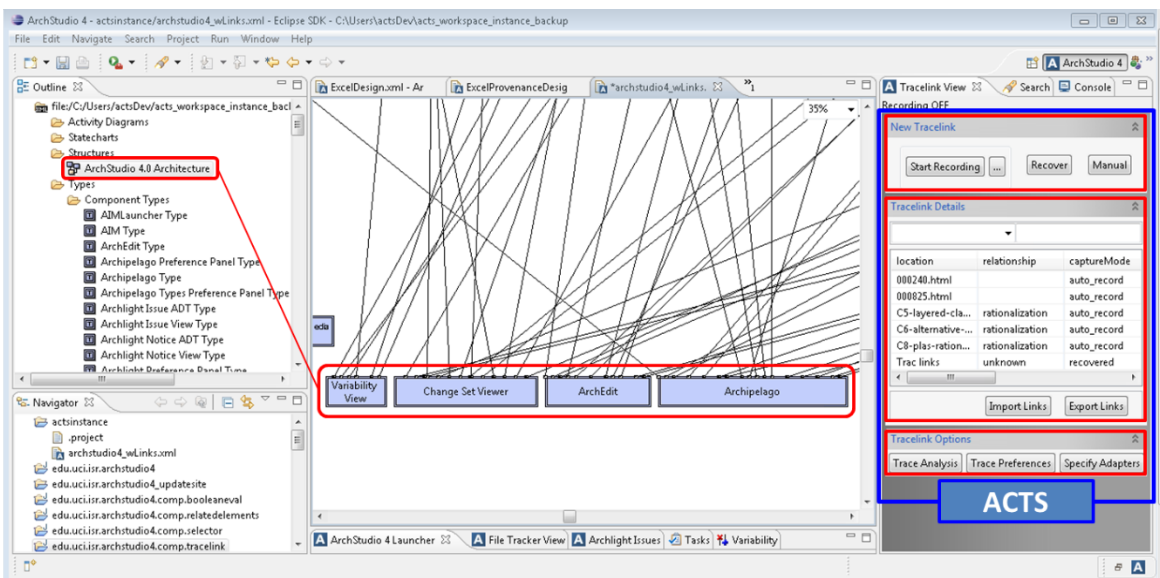


Figure 2.1: Snapshot of ACTS within Archstudio 4.

The goal of ACTS is to center links to its architecture to enable stakeholders to regulate the captured links and usage potentially. It is designed to concentrate all traceability links to one central artifact, known as the primary artifact [7]. That is, ACTS, as a traceability tool, can be viewed as a central storage facility connected to distributed open hypermedia adapters that are plugged in third-party software applications and that what makes it different than other software traceability tools mentioned in Section 2.1. Adapters are used to support capturing links from artifacts within an application. Thus far, several tools are supported by ACTS, which are Mozilla Firefox, MS Word, MS Excel, MS PowerPoint, and Adobe Acrobat [2]. In addition, ACTS enables the stakeholders to analyze, automate, and support captured links through external rules [2]. These include rules to “record,” “add relationship,” and “assign link quality”, the rules allocates relationships and filter captured traceability links [2]. These rules can either be applied in the background automatically, i.e., Background Rules, or manually in post-processing mode after data collection, i.e., Interactive Rules.

2.3 Case Studies

In the next section, qualitative and quantitative case studies are presented and discussed from economic, social, and technical perspectives. Table 2.1 summarizes the quantitative and qualitative case studies discussed in the following subsections.

Table 2.1: Qualitative and Quantitative Case Studies

Title	Study Type	Evaluation Method	Results
A Holistic Approach to Software Traceability [7]	Qualitative and Quantitative	Implemented case studies in different settings. Focused on Economic, social, and technical perspectives	ACTS covers the economic, social, and technical perspectives' interrelating challenges
A case study on value-based requirements tracing [20]	Qualitative and Quantitative	A team consisting of three project members (a project manager, quality manager, and project personnel) assessed the value of each requirement	The percentage of work needed to trace all requirements is 35%. Exhaustive tracing is needed for unstable requirements
An exploratory case study of the maintenance effectiveness of traceability models[21]	Qualitative and Quantitative	Quantitative model to evaluate the effectiveness, i.e., model dependency descriptors	The effectiveness of maintenance was increased by using a traceability tool to capture links

Title	Study Type	Evaluation Method	Results
Automating software traceability in very small companies: A case study and lessons learned [22]	Qualitative	developed a traceability environment for a small-size company, where traceability standards were not enforced	1) incremental introduction of traceability tool to convince engineers about the usefulness and effectiveness of the tool; 2) smooth integration with existing tools to capture the interest of developers; 3) customization is essential because different stakeholders have different traceability needs; and 3) trace utilization by providing search engine capabilities.
Identifying the starting impact set of a maintenance request: A case study [23]	Quantitative	Evaluated effectives and efficiency of two mathematical models (vector space and probabilistic) on data retrieval using two metrics (recall and precision)	Vector space approach supports components that initially thought important recovery process better than the probabilistic model.
Implementing requirements traceability: a case study [24]	Quantitative	Conducted various interviews with personnel in a weapon system technology (WST) organization. The personnel had different roles, which were in upper management, project management, system design, and test/audit	The source of fundamental traceability problems is human intervention
Recovering traceability links between code and documentation [13]	Quantitative	Evaluated the model's performance and effectiveness to assist a software engineer linked to standard tools. Implemented 2 case studies to apply "probabilistic and vector space	The highest recall was reached by probabilistic model. On the other side, vector space model demonstrated regular development in recall values after increasing the number of

Title	Study Type	Evaluation Method	Results
		Information Retrieval (IR) model". Eight participants of graduate and undergraduate students were included in this study.	retrieved documents. Also, it requires less effort in the preparation of the query and document representations"
Extended requirements traceability: results of an industrial case study [25]	Quantitative	The paper explained a previous case study's way of solving questions about a requirement's human source(s). They concentrated on vital issues attached to traceability insufficiency of requirements. The case study consisted of communication project with 25 participants from a small communication company and 33 participants from outside the company	Highlighted the most important problems regarding the approach use and the information use that the approach offers

2.4 Qualitative Case Studies

Qualitative case studies are an inquiry approach that concentrates on gathering mainly verbal data, not measurements. The hypothesis is in this kind of research is broad, with the emphasis of the research objective on explaining the outcomes based on data that was collected from one or a group of data collection methods. Examples of the methods include written feedback, user focus groups, interviews, and open ended discussion [16, 26]. Collected information is then analyzed in an interpretative, subjective, impressionistic or even diagnostic manner.

Qualitative research studies support various academic disciplines, typically in the social sciences, in market research, and in human-factor related studies, e.g., human-machine interface [16, 26].

Of specific interest to this thesis project, qualitative studies have been used in conducting research on evaluating software traceability tools. Some of these studies were quasi-experimental, but all studies concluded with an evaluation phase.

Quasi-experimental studies include testing a research variable from selected groups' variables that are in essence similar to traditional scientific experimental studies, but exclude any random pre-selection procedures. Generally, the random selection process guarantees the elimination of bias towards any experimental variables. Nonetheless, artifacts, in software traceability, are predefined and in some instances are prioritized depending on management decisions; thus, the quasi-experimental studies were considered suitable. That is, each artifact is assigned a specific value, depending on the mathematical model selected for analysis, and these values are traced and compared periodically. In fact, many software traceability researchers adopted these methods, such as Antoniol et al. [8, 9].

Commonly, evaluation is defined as the examination of the meritorious worthiness of a process, an object, or a tool in a systematic way [26]. The objective of the evaluation is then to provide useful feedback to different stakeholders from the collected data. System architects, software engineers, web developers, deployment managers, software testing, and evaluation engineers are just few examples of the stakeholders interested in the evaluation outcomes. However, since the stakeholders agree in the creation and usage of various artifacts related to each other, it is essential to integrate the various artifacts through traceability tool that is friendly use, reduces time, and cooperative. The interdependency of traceability and evaluation informs the stakeholders about the economic, social, and technical usefulness of the tool.

2.4.1 Economic Perspective

The economics of software traceability are related to the cost of implementation or development of the tool as well as the cost of supporting traceability process [27, 28]. The cost also includes the hours spent in training, data collection, data processing, and post-processing analysis and dissemination. In addition, the economics are related to benefit all the stakeholders. One specific benefit is the optimal use of the same traceability data to satisfy the needs of different stakeholders throughout a company or organization, as argued by Asuncion et al. [7, 29]. In short, the resolution of adopting software traceability, whether by in-house development or an OTS solution, is a basic cost-benefit decision [7]. Since ACTS relies on the integration of software adapters in distributed architecture framework, i.e., architecture centric traceability, to collect the activities from different artifacts, it sustainably reduces the cost of development and implementation [7]. Furthermore, ACTS processes the collected data in-situ, which means a significant reduction in human intervention and, therefore, in overall cost.

As the development cycle progresses and evolves, the probability of requirements becoming risky or volatile increases. This requires detailed tracing, which in turn implies a significant increase in the cost [7, 28, 29]. Heindl and Biffl proposed Value-Based Requirement Tracing (VBRT), which is a bottom-up traceability method. Heindl and Biffl methods focused on the requirements throughout the entire processes from definition to evaluation. The VBRT process included five steps: 1) requirements of definition; 2) requirements prioritization; 3) packaging of requirements; 4) linking of artifacts; and 5) evaluation [26]. The uniqueness of VBRT lies in the level of precision of each trace, i.e., prioritization. That is, the assignment of “priority” can change to accommodate the volatility of the requirements based on management decisions. Therefore, VBRT provides the benefit of traditional traceability tools, but, and more importantly, significantly reduces the time, and of course, cost of the tracing effort [20].

In some cases, the maintenance of software requires a change to the functionality or the underlying software structure. Obviously, these changes can be very costly, yet the effectiveness is hard to assess. The estimation of the effectiveness of any change can be measured by an impact analysis by analyzing the interdependency of various artifacts of the software. Such prediction, i.e., effectiveness based on impact analysis, is highly dependent on whether the analysis approach is well-structured based on documentation, which is usually preferred by management; or less-structure based on the code, which is preferred by the developers. Binachi et al. suggested that traceability models can offer a middle ground between management and developers [21]. Such that, a structured analysis approach based on traceability yields better results than the former approaches. The traceability-based approach offers greater degree of flexibility based on the degree of granularity of the software. The degree of granularity can be defined as the relative size of the smallest unit that is used to represent the software, i.e., artifacts. A fine-grained software model results in higher accuracy, while coarse-grained model is suitable for efficiency, Binachi et al. argued [21].

2.4.2 Social Perspective

The software development cycle is very dynamic, where the number of artifacts is continuously increasing. This creates complex traceability links, which are relevant to software engineering activities, such as maintenance [7, 8, 21]. Thus, the social perspective of traceability tool is important, as challenges arise from interactions between stakeholders [7]. Interactions include inter- or intra-departmental politics, confusion about artifacts ownership, and lack of proper communication between stakeholders [7, 21]. Additionally, traceability results may vary from one department to another and from one level in the corporate hierarchy to another.

Asuncion and Taylor integrated interactive or background rules to capture and analyze the traceability links after each recording session in ACTS [7]. In other words, they enabled stakeholders to record and use the links they captured, which they called “stakeholder-driven traceability” [7]. This concept is vital in the success of any traceability tool because it resolved the communication breakdown issues between the people who capture the links and those who use the links, by allowing the same user to capture and analyze links. In summary, a traceability tool must: 1) enable users to capture links based on their needs;

and 2) allow users to directly use the links captured. Earlier to this work, Asuncion et al. demonstrated that a process-oriented approach, where stakeholders are empowered by the tool to accomplish their traceability goals, encourages users to adopt the traceability tool [7, 8].

Instead of enabling the users to capture traceability links, Heindl and Biffl used a value-based system to prioritize the requirements and thus automating the traceability effort [20]. The value for each requirement was assigned in two steps. First, a team consisting of three project members (a project manager, quality manager, and project personnel) assessed the value of each requirement. The project manager assessed the value, risk, and effect of each requirement, i.e., impact analysis, while the other two members assessed the value only. The final step was done by the automated traceability system to finalize the prioritization of each requirement. However, this system sought to automate the traceability to reduce the time; it was successful in engaging the users in the decision-making process and thus gave them an ownership. This, in turn, motivated the users to adopt the tool [20].

2.4.3 Technical Perspective

There is a consensus in the software community, researchers, and practitioners on the importance of traceability during the development cycle [27]. Even a simple software application with few artifacts requires a comprehensive documentation and tracing effort. This effort is further complicated by the complexity associated with tracing a huge amount of independent artifacts, as the application size increases [27]. This is also complicated with the increase in the degree of heterogeneity of the artifacts. Therefore, the technical perspective of traceability tool specifically addresses the different kinds of artifacts for tracing relationships and their interdependency [7]

In Asuncion and collaborators' ACTS traceability tool, they joined spread- and varied-artifacts around concepts symbolized by the architecture [7, 28]. However, there are several techniques to overcome the technical challenges, Asuncion and Taylor employed hypermedia adapters to create traceability links across tool boundaries [7]. The results showed that the support for ACTS tool is extensible to integrate existing OTS tools. They investigated several case studies in different settings, such as research laboratory environment, privately-owned small company, and open-source project [7, 8]. Therefore, ACTS addresses the technical challenge by enabling users to define the rules and by providing flexibility in integrating in-house or OTS tools. ACTS is successful in linking heterogeneous artifacts and capturing custom link semantics, such as the purpose of the link [7]; this in turn reduced the overheads associated with customizing and using the tool [7].

Neumüller and Grünbacher developed a traceability environment for a small-size company, where traceability standards were not enforced [22]. The development was motivated based on the fact that: 1) small companies tend to capture links manually instead of adopting traceability tools; and 2) companies, in general, emphasize trace definitions over trace utilization. The objective of the new development environment was focused on delivering a solution, e.g., new traceability tool, with two specific roles: data warehouse and search engine. Neumüller and Grünbacher reported the following lessons learned which are focused mainly on overcoming the technical challenge poised to any traceability tool, automated or semi-automated [22]. Of the six reported lessons, only four are included here as they relate to the technical perspective of traceability: 1) incremental introduction of traceability tool to convince engineers about the usefulness and effectiveness of the tool; 2) smooth integration with existing tools to capture the interest of developers; 3) customization is essential because different stakeholders have different traceability needs; and 4) trace utilization by providing search engine capabilities.

2.5 Quantitative Case Studies

After examining qualitative studies, quantitative case studies are a typical experimental approach of most scientific disciplines studies that tackle the systematic empirical study of a phenomenon. These studies have been used in conducting research on evaluating software traceability tools [19]. The hypothesis of quantitative study is narrow, with the research objective focused on conclusive results. It uses statistical, mathematical, or numerical records, where the data are gathered quantifiably and analyzed. Examples of numerical data are closed-ended questions, questionnaires, surveys, and measurements. Generally, it concentrates in counting and categorizing types as well as assigning numerical values based specific and predefined scoring system [20]. Finally, researchers usually derive statistical models to clarify the observation [16, 26]. Traceability-focused quantitative case studies investigated both industry-based and academically-developed traceability tools [7, 23, 20]. Of the reviewed literature on quantitative research studies on software traceability, the main focus was on quantifying the effect of requirement definition and tracing either manually or automatically.

Also, Gotel and Finkelstein discussed an important shift in their methodology from artifacts-based requirements to personnel-based requirements [24]. They reported specific issues concerning their studies, which warrant additional research. For example, they reported issues about tool, artifact, or requirements ownership and their impact on the success of their approach to traceability [24]. Interestingly, many researchers have confirmed the impact of such issue on the successful development and adoption of traceability tools, such as Asuncion et al., Antoniol et al., and Ramesh et al. to name a few [7, 12, 26, 22].

Ramesh and coworkers reported on a model that described the practice of traceability in an organization as well as the perceived benefits. They stated that once an organization realized the importance of traceability on their long-term success, the organization is more prone to adopt a traceability practice [24]. The traceability model captures information from areas, such as design, allocation, and use of resources by different system components. Similarly, Egyed et al. reported on the automation of requirement traceability with emphasis not only on capturing, but also and more importantly, during the use of captured links. The main contribution of Egyed et al. approach is the creation of dependencies, which in turn informs the analysis, especially in non-standard situations [30].

2.5.1 Economic Perspective

The effort and cost associated with the adoption of a traceability tool is substantial, which has been acknowledged by many researchers [2, 7, 8, 16, 20, 21, 27 - 29]. The cost associated with traceability includes time spent in training, development, and use of the traceability tools and procedures, and the initial cost of traceability software. Ramesh et al. noted that the cost associated with traceability is almost twice as much as the cost required for the initial project [26]. However, the management realized that the cost is non-recurring and the benefits of traceability are far more important and long lasting beyond the initial implementation [26].

The high cost of traceability was confirmed by Asuncion et al. as a deterrent of implementation. Nonetheless, the benefits of implementing software traceability can be realized by the reduction of cost by automating or semi-automating the surrounding tasks linked to traceability. At any case, Asuncion et al. expressed the strong notion that manual trace links should be avoided because they are error-prone [8].

2.5.2 Social Perspective

The human factor in software development and traceability is crucial since it affects the entire process from definition of the requirements to deployment of the final product. The human factor is even more important during requirements change, and it has great implications on the success or the failure of traceability process.

Gotel et al. argued that the source of fundamental traceability problems is human intervention. Realizing the importance of this stakeholder, Gotel and Finkelstien proposed personnel-based requirements traceability rather than artifacts-based requirements, since people contributed to artifacts in the requirement engineering process. In a large quantitative case study that involved 85 people and 166 artifacts over 3 years (1992-1995), they studied the social role of people as they assumed responsibilities of artifacts development [24]. They divided the social role into four categories: true author, who completed most of the assignments on their own; nominal authors, and on a representative authors and ghost authors, who do not finish any assignments on their own, but contributes and help true authors. They found despite the initial buy-in from people, the long term success of traceability hinges on the availability of resources to complete the work and trace it. That is, as the person's responsibility increases, traceability starts taking less priority [24].

Automation can also be used to help a development team realize the importance of traceability, as in the case of Egyed and Grünbacher research report [30]. They generated trace information automatically to develop a tool-support for a technique to facilitate trace acquisition. They produced a video-on-demand (VOD) system case study through a trace analyzer technique to classify trace dependencies fully automatically. This automated approach created traces not easily anticipated by engineers; therefore, it excited them about the possibilities and potential time saving, in using traceability automated tools [30].

2.5.3 Technical Perspective

There have been many methods proposed to address the technical challenges of traceability tools. As mentioned in Section 2.2.3, the technical challenge is based on the complexity from the heterogeneity of artifacts. Techniques used to resolve this issue include natural language processing, model transformation, translation technique, shared repository, specialized code, and open hypermedia adapters [12, 22, 31, 16, 7]. The latter provides a unique solution since the software adapters can be developed independently of the traceability tool based on the present need or application. The idea, as discussed by Asuncion et al. is once a team has been trained on the traceability tool, e.g., ACTS, the addition of new adapters or new tracing rules will not require retraining of personnel [7, 8]. They tested this approach in an industry-based software development setting, where it was reported that identification of global and local artifacts and maintaining consistency with automation elevated the technical challenge [8].

Correspondingly, Bianchi et al. conducted an exploratory study to assess the effectiveness of traceability models on the maintenance of software system. They performed impact analysis on the implicit and explicit links generated by the traceability tools [21]. They used a quantitative model to evaluate the effectiveness, i.e., model dependency descriptors. The model provided a graph that represented the software system. Their findings included the following: automated impact analysis was insensitive to errors; as the complexity of the software increased, the difference between the estimated and actual set of links increased; and as the number of links increased, the secondary impact increased. In summary, the effectiveness of maintenance was increased by using a traceability tool to capture links [21].

Finally, Henidl et al. implemented a value-based requirement tracing, which systematically supported management in tracing precision, stakeholder effort, and requirement risk/volatility [20]. They found that for the VBRT tool, only 35% of the effort was required for full requirement tracing and detailed tracing is required for artifacts with higher probability of change [20].

2.6 Historic Perspectives on Case Study Designs

This section discusses the theory and propositions of some qualitative and quantitative case studies, which includes the research questions and the research methodology utilized to answer them. It also includes information about the participants contributing to each case study.

Antoniol et al. designed two quantitative case studies to recover traceability links through Information Retrieval (IR) methods. The first study implemented a probabilistic model, while the second implemented a vector space model. The case studies aimed to perform traceability between

C++ and Java source codes with manual pages and functional requirements, respectively. The study compared results based on the previously mentioned models, and then they discussed the benefits, restrictions, and directions for enhancements. Specifically, Antoniol et al. evaluated the performance and effectiveness of the model to assist a software engineer linked to standard tools. They focused on the amount of work saving to each document's space granularity as well as comparing methods to retrieve number of documents. Each of the case studies included four students from different academic levels. The study divided the students into two groups; the first group consisted of one post graduate students and three undergraduate students. Dissimilarly, the second group consisted of one undergraduate student and three post graduate students. The students had different level of Java programming skills, but they all knew about object oriented programming patterns. Undergraduate students were introduced to Java six months before they started, while the post graduate students already had experience. Each student was provided with requirements documents and source code, but the first group was provided with more resources than the second group. The additional resources included ranked requirements list, which was provided through applying traceability links to probabilistic IR model. The results of the first group were better than the second group's results in regards to the probabilistic IR model performance [13].

Ramesh et al. implemented a qualitative case study to present a comprehensive traceability vision of an organization that practice system development. The goal of the study goal was to deliver a comprehensive model of information that could be used in traceability view. Ramesh et al. conducted various interviews with personnel in a weapon system technology (WST) organization. The personnel had different roles, which were in upper management, project management, system design, and test/audit. The study defined the traceability view and examined the usage and profits of traceability through day-to-day operations from different stakeholder's perspectives. The conclusion was requirement traceability is vital for WST quality management applications and from different stakeholders' perspectives regardless their roles. The product's quality, system development and maintenance methods are related to the costs accompanied to developing a comprehensive traceability scheme. This study supports the economic and social impact of traceability on software development [24].

Bianchi et al. conducted a quantitative exploratory case study to assess the efficiency and accuracy of traceability models. The study evaluated an adopted granularity of traceability model in connection to the efficiency of maintenance method. Specifically, the goal of the study was to evaluate the way various traceability links models support object-oriented environment analysis; and the impact of these links on accuracy of maintenance method. The case study consisted of ten students from University of Bari, Italy. They were divided into two equal groups to work on Model Dependency Descriptor (MDD), which is a framework to characterize a software system. They both applied three maintenance requests, which are analyzing primary impact, secondary impact, and specification of changes. Analysis of primary impact is 'subset' of the software systems that qualifies as contenders for change. The analysis of secondary impact contains another 'subset' of artifacts that belongs to the primary contenders. The specification of the changes is a subset of software system entities that would actually be changed. They concluded that the degree of granularity is dependent on software management, which implies social impact of traceability is more prominent in such case [21].

Gotel et al. explored a qualitative case study in an industrial project environment, in which they studied the contribution of structures modeling and usage [25]. The paper explained a previous case study's way of solving questions about a requirement's human source(s). They concentrated on vital issues attached to traceability insufficiency of requirements. The case study consisted of communication project with 25 participants from a small communication company and 33 participants from outside the company. The company focused on solving communications problems through software and practical resolutions. The data were collected through witnessing meetings, taking notes, recording sessions, and taking photocopies of artifacts. The project created 166 artifacts; 23 related to customer-specific service, 65 related to generic service development of baseline, 39 related to basic generic service development, and 39 related to generic service extended development. Therefore, this study reported on the social perspective from the accessibility, and utilizability standpoint [25].

While the previous techniques examined the economic, social, or technical perspectives, this thesis aimed to holistically evaluate ACTS from all three perspectives. Specifically, this research investigation is a mixed method of quantitative and qualitative evaluation of the ACTS technique. The purpose of this study is to assess the capturability, accessibility, utilizability, affordability, and customizability of ACTS. It provides researchers with more insights into traceability techniques and their impacts and drawbacks. The details of design and implementation of the case studies used in this investigation are discussed in the next chapter.

CHAPTER 3 METHODOLOGY: STUDY CASE DESIGN

An embedded single-case study was designed and implemented in this research to evaluate the research hypotheses discussed earlier: Hypothesis 1 and 2. The rational for selecting single-case here is the objective to record the feedback of users and developers. That is, the single-case study is justified because it is a representative study of the feedback about ACTS, i.e., common situation. The outcome of this case study is expected to inform further improvement and developments of ACTS based on the users and developers experience. In this research the users are divided into two groups. Thus, this study is focused on analyzing multiple units, i.e., embedded single-case study. To avoid common pitfalls with design and implementation with single-case study, the following units have been defined within the users groups: users without any traceability experience, users with traceability experience, and different filtering rules [32].

This chapter explains the methodology process used to evaluate the ACTS technique through two case study designs: 1) for users and 1 for developers. The chapter includes: 1) Pilot Study; 2) User Study design; 3) Developer Study Design; 4) Survey Questions Classifications; 5) Users' Questions; and 6) Developers' Questions.

3.1 Pilot Study

The pilot study's goal was to test the process of evaluating ACTS technique, i.e., making sure that the survey questions were written clearly, and the process of using the tool is understandable. The participants involved in this study had to go through the same process that would be doing in the users' official study. This study included four participants from Professor Hazeline Asuncion's research group from University of Washington Bothell, and they were: a) a graduate research assistant; b) an undergraduate research assistant; c) a research analyst; and d) an undergrad student. The study facilitated the process of the user study design and trained the researcher on how to explain steps that the user needs to follow.

3.2 Users Case Study Design

The steps for conducting successfully case study are: 1) preparing the research hypothesis and questions to evaluate the ACTS based on feedback provided by the users and developer; 2) create research protocol; 3) define clear data collection procedure; 4) define methods to recruit and screen candidates which was done thought many channels based on the investigator's professional network; 5) create a demonstration and survey instrument to evaluate the tools; 6) design data analysis and interpretation methodology; and 6) evaluate the feedback and report on the results. The user's research protocol consisted of 7-10 minutes demonstration of ACTS, allowing the user to use ACTS for few minutes to collect trace links from various artifacts from a third-party application, and finally respond to a survey instrument according to the users' usage of the tool. If the user wishes to spend more time to learn ACTS, in addition to the 7-10 minutes of demonstration, they were allowed to do so. The questions on the survey included background, quantitative and qualitative questions, which allowed the collection of a richer dataset to support the objective of the case study.

The objective of this study is to measure the capturability, accessibility, and utilizability of ACTS technique from the economic, social, and technical perspective based if the user had prior traceability or no prior traceability experience. Therefore, there were two types of users involved in that study design, users with no prior traceability experience, and users with prior traceability experience. They were recruited to assess ACTS tool from the economic, social, and technical perspective with emphasis in capturability, accessibility, and utilizability.

The users recruited were from the industry; some of them were researchers, and most of the ones from industry were software engineers. Most of the users with no prior traceability experience were working at Microsoft Inc. and Google Inc., and most of the users with prior traceability experience were from Boeing Company.

In order to assess ACTS tool, the study had to evaluate two techniques that were developed inside ACTS: 1) Interactive Rules; and 2) Background Rules. In the Interactive Rules Group, users had to apply rules interactively (manually) to a recording session, in that case the user can choose rules to apply; while in Background Rules Group, rules will be applied in the background (automatically). Users with no prior traceability experience were divided into two equal groups: a group to evaluate the Interactive Rules, and a group to evaluate the Background Rules. A similar division was applied to users with traceability experience.

To facilitate the process of evaluating ACTS tool, it was installed on a virtual remote machine in University of Washington Bothell in order for the participants to use it without the need to install it on their personal machines. Users were given a brief tutorial on how to use the tool preceding evaluating the tool. The steps for using the tool are described in Figure 3.1 and screenshots were added in the appendices. Filtering a component was stated as optional in Figure 3.1 as a group of user applied filters and a group did not apply filters.

The steps that user followed were:

1) Logon to a remote machine located in University of Washington Bothell, the ACTS tool will be opened. (The user was given the machine name, user name, and password to access the virtual remote machine).
2) Click on “Start Recording” button.

3) Interact with the architecture by choosing a component, and then interact third-party tools, such as Mozilla Firefox, Microsoft Word Document, Microsoft PowerPoint Document, or Microsoft Excel Document.

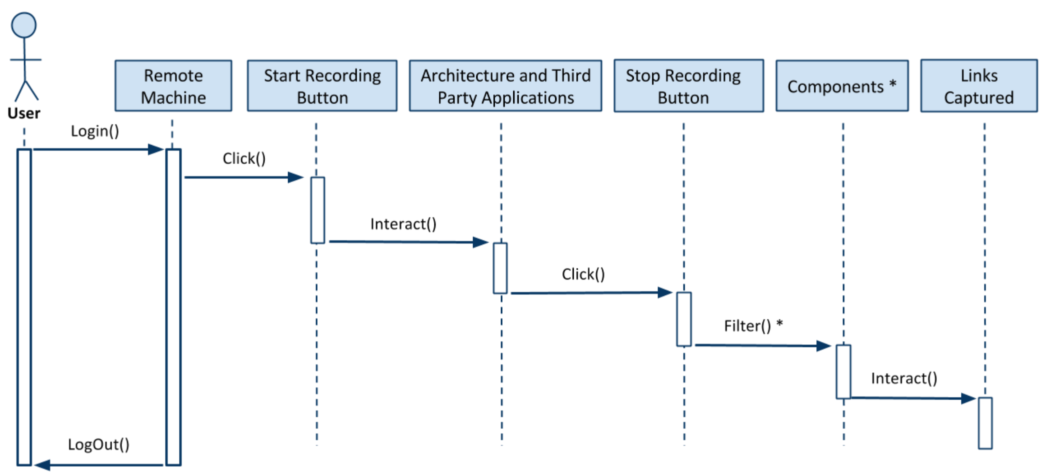
4) After finishing the recording session, the user would close all the applications, and then click on “Stop Recording” button.

5) Then filter the components if filters were interactively applied. For this study, 15 users out of 30 of the non-traceability users applied filters interactively; the other 15 users applied filters in the background. Similarly, four users out of eight users applied filters interactively; the other four users applied filters in the background.

6) Interact with recorded links through reviewing, retrieving and deleting links

7) Final step would be logging out of the machine.

8) Answer a set of survey questions described in the appendices.



*** Optional:**
1- A group of Non-traceability users (n= 15) interactively applied filters, the other group (n=15) did not apply filters interactively as it was applied in the background.
2- Another group of traceability users (n=4) applied filters, the other group (n=4) didn’t apply filters interactively as it was applied in the background.

Figure 3.1: ACTS user’s sequence diagram.

The survey questions were designed to evaluate the capturability, accessibility, and utilizability, which have an impact on economic, social, and technical perspective. The survey questions were hosted online on University of Washington catalyst to be accessible from anywhere for all the users.

Since the economic, social, and technical perspective would differ from a user standpoint than from a developer's standpoint, another case study design for developers was created.

3.3 Developers Study Design

The objective of this study is to measure the affordability and customizability of ACTS technique from economic, social, and technical perspective. Five developers were involved in this study; two of them were former students at University of Washington Bothell. Three of them were current student involved in an undergrad program at University of Washington at the time the study was done. The developers were recruited to work on developing different hypermedia adapters that would be integrated into ACTS.

The following process was followed:

1) The researcher had to search about third-party tools that weren't integrated yet into ACTS, then decide whether those tools were able to be traced or not through Software Development Kit (SDK) libraries.
2) The developer had to learn about ACTS tool and software traceability through the researcher.
3) The developer was given previous codes for software adapters that has been already developed.
4) The developer had to learn about third-party tool SDK libraries, and then develop hypermedia adapters as a standalone application.

5) The developer had to learn about ACTS codebase and then integrate the hypermedia adapter into ACTS.

6) Answer a set of survey questions (Appendix B).

The survey questions were hosted on University of Washington catalyst to evaluate affordability and customizability of ACTS technique from the economic, social, and technical perspective.

The main concern was about the affordability, and customizability of developing hypermedia adapter and integrating it into ACTS.

3.4 Survey Questions Classifications

The survey questions targeted three perspectives i.e., the economic, social, and technical perspectives with the emphasis in capturability, accessibility, utilizability, affordability, and customizability. The following are the definitions of the previous classifications; they were derived from prior research by Asuncion and Taylor [11]:

Capturability: concerned with captured traceability links and also the kind of information captured;

Accessibility: concerned with capability to navigate and to find traceability links;

Utilizability: is concerned with the usage of the traceability links e.g., using links to support a software development task;

Affordability: concerned with effort needed to train users;

Customizability: concerned with the ease to customize traceability according to different set of users.

From the previous definitions, if we tackled the user's viewpoint, the economic perspective can be mapped directly to capturability as it is concerned with time spent to record capture links. At the same time, capturability is concerned with social and technical perspective because it can evaluate ACTS techniques' effectiveness, accuracy, and the loss of captured links. The social perspective is also related to accessibility as it is connected to how ACTS will facilitate the software development process. Likewise, accessibility is linked to economic perspective through evaluating the acceptability of time to find traceability links and time to retrieve, and review captured links. The utilizability is associated to the social and technical perspective as could be used to assess if ACTS could be used to trace future projects or not. It also could be used as a measure to rate the usefulness, the accuracy, and the loss of the links captured.

From the developer's viewpoint, the economic perspective could be mapped to affordability as it is concerned with the hours a developer spent in creating an adapter, learning ACTS codebase, learning a programming language used in the development process. The affordability was connected to both the economic and technical perspective by evaluating the easiness to integrate the adapter with ACTS as it is measured as an effort to train developers. Similarly customizability was related to both economic and technical perspective as it is concerned with the easiness of developing a software adapter, and integrating it into ACTS. That was considered to be the easiness of customizing ACTS based on various set of users.

The classification of each question based on the perspectives of software traceability and the properties pertained to each perspective is shown in Table 3.1. The classification was done based on the definitions discussed thereafter. It is important to note that: 1) a question may fit under multiple perspectives and may satisfy multiple properties; and 2) the properties were derived from prior research by Asuncion and Taylor as well as the wording of the evaluation instrument questions [10]. Table 3.1 contains four columns: 1) classifications mentioned previously as the emphasis; 2) kind of participant that could include users or developers; 3) questions number that relates to each classification and type of participant; and 4) perspectives affected by that classification.

Table 3.1: Classification table for users and developers

Classification	Kind of Participant	Question	Perspective
Capturability	**Users**	• Is ACTS more effective than your current technique of tracing? • How accurate was the traceability links that were captured? • Were there missing links?	Economic, Social, and Technical
Accessibility	**Users**	• Do you think the tool would facilitate software development by making relevant artifacts accessible to all members of the development team? How? • On a scale of 1 to 5, rate the acceptability of time spent on finding relevant traceability links. • Rate the acceptability for referring time to back link retrieving and reviewing.	Social
Utilizability	**Users**	• Would you use this tool for future projects? Why or Why not? • On a scale of 1 to 5, rate the usefulness of the links created with the ACTS tool. • How accurate was the traceability links that were captured? Use your best estimate. • Were there missing links? Use your best estimate.	Social, and Technical
Affordability	**Developers**	• Using your best estimate, how many hours did you spend in creating the software adapter? • What percentage of this time was spent in learning ACTS? • What percentage of this time was spent in learning the programming language and the API of the third-party tool? • Was your software adapter able to integrate with the ACTS tool? If not, why not? • On a scale of 1 to 5, rate ease of integrating the third-party tool into ACTS using the software adapter you developed.	Economic

Classification	Kind of Participant	Question	Perspective
Customizability	**Developers**	• On a scale of 1 to 5, rate ease of developing a software adapter for a third-party tool. • Was your software adapter able to integrate with the ACTS tool? • On a scale of 1 to 5, rate ease of integrating the third-party tool into ACTS using the software adapter you developed.	Economic, and Technical

In connection to the survey questions, there were background and perspective questions related to both users and developers. The following sections explain the background and perspectives questions that are economic, social, and technical included in both study designs, i.e., users and developers.

3.5 Users' Questions

The survey instrument used in evaluating the users' feedback included background and perspectives questions. The background questions were a short list that contained basic questions, while perspectives' questions were a long list that tackled the economic, social, and technical perspective.

3.5.1 Users' Background Questions

The background questions listed below were necessary for finding out about the participant background to connect the participant's experience with the results:

Q1: Primary Role or Occupation
Q2: How long have you held this role?
Q3: Secondary Role or Occupation
Q4: How long have you held this role?
Q5: What type of software does your organization develop?
Q6: Please specify the operating that you use: (Windows, Linux, UNIX, MAC OS)
Q7: Have you used eclipse before?
Q8: If yes, then for how long? (0-1 months, 1-3 months, 3-6 months, 6 or more months)

The background questions targeted the role or occupation and the length of time the participant held that role, which helped in revealing the participant's background and knowledge about software architecture and development. It also targeted the type of software that they develop with their organization, which helped in finding out if they use/used software traceability tools or not. Lastly, the operating system that the participants used and if used Eclipse for a sometime or not helped in minimizing the time spent on training as the tool was developed in Eclipse and on Windows environment.

3.5.2 Users' Perspective Questions

The user's perspective questions were designed to map the economic, social, and technical perspectives. They were a mix of quantitative and qualitative questions. The following are some of the questions tackling economic and social, and technical perspectives. The entire list of questions with explanation is included in Appendix A.

- **Economic Perspectives Questions:** focused on the time spent in recording traceability links, finding relevant traceability links, and the referring time to retrieve, and review links.
- **Social Perspectives Questions:** focused on finding the users' ideal traceability tool, comparing ACTS with other existing traceability tools, and if the user wants to use the tool in the future. Other social perspectives questions asked about if the user would recommend the tool to their peers, if ACTS is more effective than user's current technique of tracing, if ACTS would facilitate the software development process by making relevant artifacts accessible to all members of the development team, and if the links captured were useful or not. Some questions focused on how user-friendly is the tool, challenges encountered, and what the user liked and not liked about ACTS.

- **Technical Perspectives Questions:** focused on if the tool would facilitate the software development process by making relevant artifacts accessible to all members of the development team, rating the usefulness of the links created, accuracy of the traceability links captured, and the percentage of the missing links. Other questions asked about features needed to be added to ACTS, challenges encountered, and what the user liked and not liked about ACTS.

3.6 Developers' Questions

Similarly, the developers' questions included background questions and perspectives questions. The questions followed the same structure as those in the users' case study.

3.6.1 Developers' Background Questions

The survey questions include the background questions, which were necessary for finding out about the participant background. The role, or occupation and the length of time the participant held that role helped in reveal the participant's background and knowledge about software architecture and development. Below is the list of background questions:

Q1: Primary Role or Occupation
Q2: How long have you held this role?
Q3: Secondary Role or Occupation
Q4: How long have you held this role?
Q10: In what context did you develop the software adapters?

3.6.2 Developers' Perspective Questions

The developers' perspective questions were designed to map the economic, social, and technical perspectives. The following are some of the questions concerning economic and social, and technical perspectives. The entire list of questions in the developers' survey is included in Appendix B.

- **Economic Perspectives Questions:** focused on if the developer worked on ACTS prior to creating the software adapter, years of experience in using programming language used in the development process and in other programming languages, hours spent in creating the software adapter, percentage of time spent in learning ACTS, and learning the programming language to develop the software adapter. Other questions asked about the easiness of developing a software adapter and integrating within ACTS, if the software adapter was integrated inside ACTS, and the easiest and the most difficult part in developing the software adapter.

- **Social Perspectives Questions:** focused on if the developer worked on ACTS prior to creating the software adapter, the reason behind choosing the programming language, years of experience in using programming language used in the development process and in other programming languages, and the easiest and most difficult part in developing the software adapter.
- **Technical Perspectives Questions:** focused on the programming language used in developing the software adapter to which third-party tool. Other questions asked about the easiness of developing a software adapter and integrating within ACTS, the ability to integrate the newly developed software adapter within ACTS, and the easiest and the most difficult part in developing the software adapter.

In summary, two case studies were designed and administered to users and developer of ACTS. In Chapter 4, the results of the survey questions were analyzed based on different groups involved in the study.

CHAPTER 4: RESULTS AND DISCUSSION

This chapter includes the analysis of the results of the survey questions from users and developers. As mention in the previous chapter, there were two types of surveys corresponding to the two stakeholder of interest to this research. One survey that was administered to users of traceability tools and the other for developers of hypermedia adapters. The questions in each survey were divided into two subsets. The first subset collected information about the participants, which included the primary and secondary occupations and the duration at each occupation, respectively. The second subset contained qualitative and quantitative questions to assess the perspectives of software traceability and its properties. Figure 4.1 provides an illustration of the breakdown of participants and the data collection arrangement, which is discussed in detail in the subsequent sections. All participants were recruited and engaged in the study according to approved Internal Review Board Authorization (IRB), which was approved by the University of Washington Bothell. A list of the questions for the users and the developers are presented in Appendix A and B, respectively.

4.1 Users Survey

The same survey instrument was administered to four different independent groups of users, where two of the groups had no prior traceability experience with a total of 30 participants and the other two had prior experience with total of eight users. In addition to the division based on traceability experience, the users were further divided into two subgroups to evaluate the same functionality of ACTS tool. Table 4.1 shows the number of participants in each group, the first group, no prior experience, (number of users (n) = 15) used ACTS with interactive filters enabled, which means they needed to filter their links manually (Interactive Rules Group). The second group of users, no prior experience, (n = 15) used ACTS without interactive filters enabled; that is the filtering rules were applied automatically in the background (Background Rules Group).

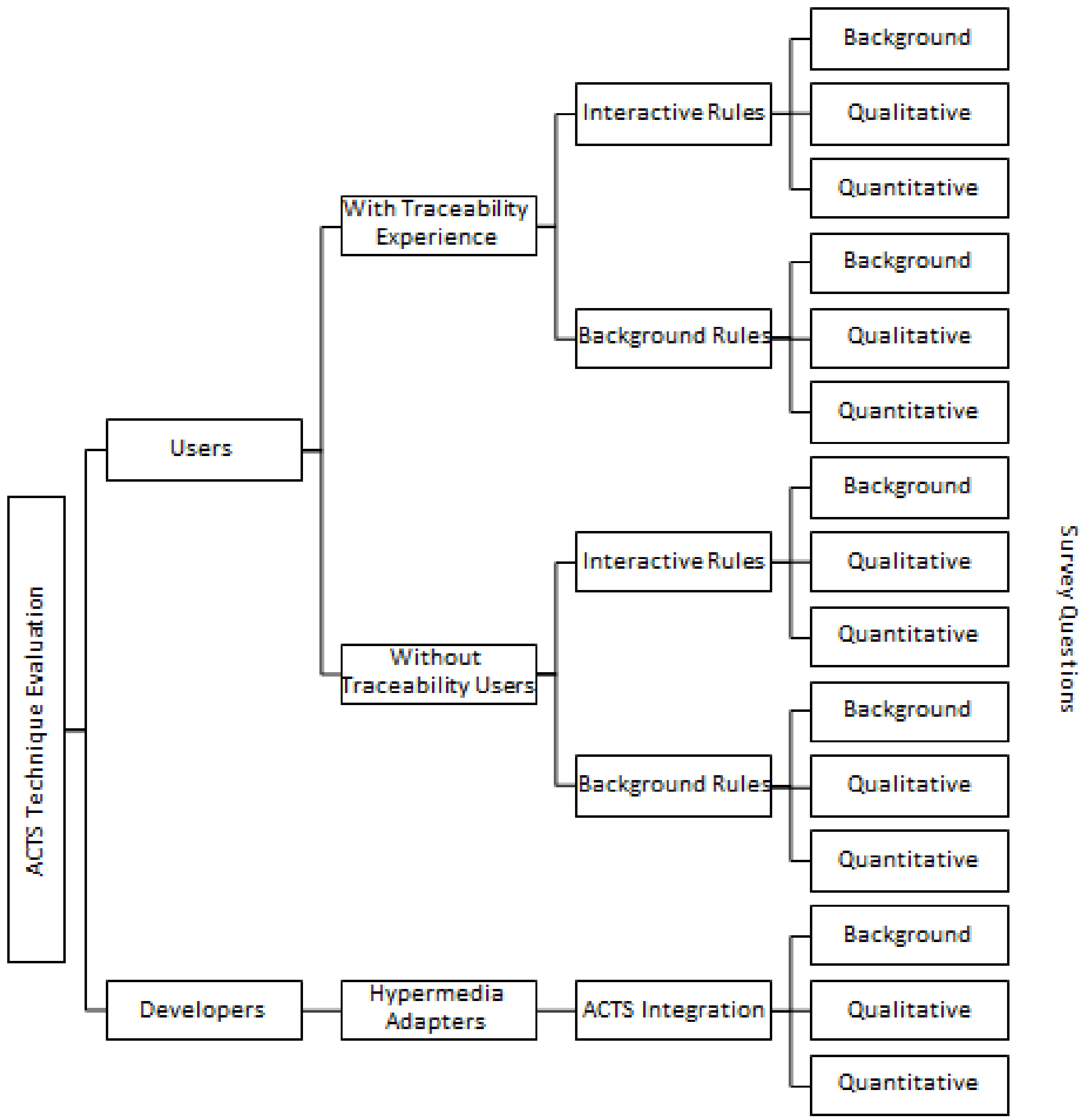


Figure 4.1: Breakdown of participants and the data collection arrangement.

The other two groups mirrored the aforementioned division, but all users had prior traceability experience. In total, 38 different users took part of this study. Figure 4.1 shows the users breakdown based on their traceability experience and ACTS settings. The size of samples for users with prior experience was significantly smaller than the users with no experience due to lack of availability of such users. It is worth noting that the research team tried multiple approaches –such as social media, professional contacts, and industry collaborators – to recruit users with prior experience.

Table 4.1: Number of participants in each group

Users Experience	ACTS (Interactive Rule)	ACTS (Background Rule)
Non-Traceability	15	15
Traceability	4	4

The users were randomly recruited based on their willingness to participate in the study and received no compensation for their participation. The survey was reviewed and approved by human subject committee at UW (IRB#45324). The researcher met individually with each user in a face-to-face style meeting, either in-person or on-line. The session with each user lasted approximately from 30 to 45 minutes, where an introduction to the study, a quick tutorial to ACTS, and instructions about how to complete the survey were given to each user. The tutorial lasted about seven minutes on average.
The results of the users' survey have been analyzed using one-tailed paired-samples *t*-test with 95% confidence interval. The overarching null hypothesis of all the quantitative questions is: *there is no statistically significant difference between the feedbacks provided by the two user groups at 95% confidence interval*. Or simply put, the null hypothesis is the mean of the two data sets is the same, i.e., there is no difference between the Interactive Rules Group and Background Rules Group. An alternative hypothesis may be considered as the mutually exclusive and opposite to the null. The paired *t*-test is appropriate because the users were randomly recruited, i.e., random sample, and those answered were paired between the user groups, i.e., participants answered the same questions using the same scale. In what follows, we have divided the presentation of results and discussion based on the users' traceability experience or the lack there of. In each subsection, a statistically-based comparison is done between Interactive and Back ground groups.

4.1.1 Non Traceability Users

As stated, two groups of users participated in this part of the study: Interactive Rules and Background Rules groups. Each group consisted of 15 users. All 30 participants had no prior traceability experience. The Interactive Rules group consisted primarily of software engineers, where 12 participants indicated as their primary role. Additionally, six users in this group indicated they held a secondary occupation in computer science. The group duration of experience ranged from 0.2 to 14 years. The Background Rules group consisted of primarily technical managers in software-based companies with experience duration that ranged between 0.5 year and 8 years. Nine users within this group indicated they held a secondary role.

Tables 4.2-4.8 show the results of quantitative questions, where the number of responses for each question corresponds to each category as it was counted and tabulated. Table 4.2 represents the results of the recommendations of the users of ACTS to their peers, which was evaluated on four-point scale. Table 4.4-4.5 are the responses of both groups on a three-point scale to the questions evaluating the effectiveness of ACTS and whether the tool would facilitate software development or not. Tables 4.6-4.9 are the responses of both groups for questions on five-point scale, where the questions were assessing the usefulness, and time to record, find, and retrieve links, respectively.

Most notably, the numerical count of each of the categories from each of the groups was almost the same. For example, the number of users that would “Recommend” ACTS to their peers was six and eight from the Interactive and Background groups, respectively. None of the survey participants from either groups rated “Not useful at all” for usefulness of the links, time spent in recording and retraining the links, and time spent in finding the links. In other words, there is a consensus between all users without prior traceability experience on the effectiveness of ACTS as a traceability tool. After all, 60% of participants in this group would “Recommend” or “Strongly Recommend” ACTS to their peers. The users from the Interactive Rules Group mentioned that ACTS technique is a professional way for organizing information from multiple artifacts, and connect the information to each component in the design, that would save time in the development process. Also, the users from Background Rules Group mentioned that it saves time, and it is useful for complex projects. Four users from Background Rules Group mentioned the tool is friendly use.

The p-value calculated from one-tailed paired-samples *t*-test is consistently greater than the 5% significance level ($\alpha = 0.05$), which suggests that there is no reason to reject the null hypothesis. That is, the mean difference between the feedback provided by the Interactive Rules and the Background Rules groups was not statistically significant (p-value = 0.5) and thus the data has randomly drawn from the same population. Table 4.3 shows that there is no statistically significant difference (p-value = 0.5) between the willingness of both groups on suggesting ACTS as a traceability tool to their peers. Interestingly, despite the lack of formal software traceability experience, there was no difference (p-value = 0.5) between the indications of both groups on the effectiveness of ACTS with respect to the methods of tracing that they are currently using.

Table 4.2: Would you recommend this tool to you peers? (p-value = 0.5)

Category	Interactive	Background
Strongly Recommend	1	3
Recommend	6	8
Indifferent	8	3
Do Not Recommend	0	1

Table 4.3: Is ACTS more effective than your current technique of tracing? (p-value = 0.5)

Category	Interactive	Background
Yes	5	5
Maybe	7	8
No	3	2

Table 4.4: Do you think the tool would facilitate software development by making relevant artifacts accessible to all members of the development team? (p-value =0 .5)

Category	Interactive	Background
Yes	10	12
Maybe	4	2
No	1	1

Table 4.5: Rate the usefulness of the links (p-value = 0.5)

Category	Interactive	Background
Most Useful	2	5
Useful	3	6
Somewhat useful	8	2
Not Useful	2	2
Not Useful At All	0	0

Table 4.6: Rate the acceptability of time spent to do the recording traceability links (p-value = 0.5)

Category	Interactive	Background
Most Acceptable	5	6
Acceptable	3	5
Somewhat Acceptable	3	3
Not Acceptable	4	1
Not Acceptable At All	0	0

Table 4.7: Rate the acceptability of time spent on finding relevant traceability links (p-value = 0.5)

Category	Interactive	Background
Most Acceptable	2	7
Acceptable	2	3
Somewhat Acceptable	7	4
Not Acceptable	4	1
Not Acceptable at All	0	0

Table 4.8: Rate the acceptability for referring time to retrieve and review links (p-value = 0.5)

Category	Interactive	Background
Most Acceptable	7	8
Acceptable	1	5
Somewhat Acceptable	7	1
Not Acceptable	0	1
Not Acceptable at All	0	0

Table 4.9 shows that 83% of the participants in both groups rated ACTS above 70% accurate in capturing traceability links with no statistically significant difference in the responses (p-value = 0.5). It seems, however not significant, that ACTS is less accurate when users applied manual filters, where two users indicated that ACTS was less than 50% accurate. Remarkably, users unanimously (29 users) agreed that

less than 10% of the links were missing and one user didn't answer that question. That is to say, ACTS has a positive technical impact on software traceability as it applies to capturability, utilizability, and effectiveness.

Table 4.9: How accurate was the traceability links that were captured? (p-value = 0.5)

Category	Interactive	Background
91%-100% Accurate	7	7
81-90% Accurate	5	4
71%-80% Accurate	0	2
60%-70% Accurate	1	2
50- 60% Accurate	0	0
Less than 50%	2	0

Table 4.10 shows the users prioritization of the important aspects of a traceability tool. The data collected from users with no prior traceability experience point to no statistically significant difference (p-value = 0.5) between interactive or background rule. It is important to note that approximately 57% of all users selected that timeliness is the most important aspect of the traceability tool as highlighted in Table 4.10.

Table 4.10: Prioritize the friendliness, timeliness and helpfulness of traceability tool (p-value = 0.5)

Category	Interactive	Background
1) Friendliness, 2) Timeliness, 3) Helpfulness	1	3
1) Friendliness, 2) Helpfulness, 3) Timeliness	1	0
1) Timeliness, 2) Friendliness, 3) Helpfulness	4	4
1) Timeliness, 2) Helpfulness, 3) Friendliness	6	3
1) Helpfulness, 2) Friendliness, 3) Timeliness	0	1
1) Helpfulness, 2) Timeliness, 3) Friendliness	3	4

The users also provided valuable feedback in open-ended questions. When asked about features that are not currently available in ACTS, the users in the Interactive Rules group suggested: 1) developing more hypermedia adapters to support more software application, e.g., browsers and Microsoft Visio; 2) ability to define, use, sort, and control multiple filters; 3) external access to software applications, i.e., open from outside ACTS; 4) introducing machine learning capabilities; and 5) developing more software adapters such as supporting more browsers and other modeling tools like Visio. While users from Background Rules group suggested elimination of transition links when using Firefox adaptor and agreed with the other group on improving filters.

The Interactive Rules users listed some challenges in using ACTS. These challenges include: the interface is not user-friendly, functionality not effective, filters, and noise in recorded links. The users suggested the addition of functions to: 1) refresh to update links after recording and deleting; and 2) detecting active components. While, the rest of the interactive group indicated that the tool was easy to use and logical. In contrast, 40% of Background Rules Group reported that they did not encounter any challenges while using ACTS, and the rest of the users from this group conveyed the same challenges reported by the Interactive Rules group, which are the difficulty in using the interface, ineffectiveness of some of the functionality, applying filter, and noise in links. In general, this is consistent with quantitative results, where the results from the two considered groups show no statistically significant difference. It is worth noting that users from both groups reiterated these challenges when asked about aspects they did not like in ACTS. On the other hand, users appreciated the capabilities of ACTS to capture, store, and retrieve links from external applications.

4.1.2 Traceability Users

As mentioned in Section 4.1, two groups of users participated in this part of the study similar to non-traceability users: Interactive Rules and Background Rules groups. Unlike non-traceability users, each group consisted of only four users. The Interactive Rules group consisted of two software engineers, one system analyst, and one avionics design engineer as stated by their primary role. Also, two of the previous users indicated their secondary role: one as researcher and the other as software developer. The group duration of experience was from four to 14 years. On the other side, the Background Rules group consisted of a computer scientist, project manager, system engineer, and a software engineer;

one of them indicated the secondary role as a computer engineer. The group duration of experience was from 3.5 to 7 years.

Tables 4.11 - 4.18 show the results of quantitative questions, where the number of responses for each question corresponds to each category as it was counted and tabulated. Similar to Section 4.1.1 ,Table 4.11 represents the results of users recommendation of ACTS to their peers, which was evaluated on 4-point scale, while Table 4.12-4.13 are the responses of both groups on a 3-point evaluating scale the effectiveness of ACTS and whether the tool would facilitate software development or not. Tables 4.14-4.16 are the responses of both groups for questions with 5-point scale, where the questions were assessing the usefulness and time to record, find, and retrieve links, respectively.

In response to whether or not the participant would recommend ACTS to peers as a traceability tool, it was found that none of the users from either groups marked “Do Not Recommend” the tool to their peers. Three out of four of the Interactive Rules group rated indifferent, while two out of four of the Background Rule group rated indifferent, that means that they developed neither a negative nor a positive perception about ACTS. Even though, two out of three the Interactive Rules who selected indifferent for recommending tool to peers, selected “Yes” for the tool facilitating the software development process while one user in the Interactive Rule Group selected “Maybe”. It was fascinating that the four users for Background Rule group selected “Yes” for the tool facilitating the software development process.

None of the survey participants from either groups selected “Not useful” nor “Not useful at all” for usefulness of the links. Similarly, none of the users rated “Not acceptable” nor “Not acceptable at all” for the acceptability of time spent in recording and retraining the links, and time spent in finding the links. It was notable that, there was an agreement between all users prior traceability experience on the effectiveness of ACTS as a traceability tool.

Table 4.18 shows that three out four users of the Interactive Group rated the accuracy of the links as more than 71% accurate. Likewise, three out four users of the Background Group rated the accuracy of the links as more than 71% accurate. Therefore, the mean difference between the feedback provided by the Interactive Rules and the Background Rules groups was not statistically significant (p-value = 0.5).

Table 4.11: Would you recommend this tool to you peers? (p-value = 0.5)

Category	Interactive	Background
Strongly Recommend	0	1
Recommend	1	1
Indifferent	3	2
Do Not Recommend	0	0

Table 4.12: Is ACTS more effective than your current technique of tracing? (p-value = 0.5)

Category	Interactive	Background
Yes	0	2
Maybe	3	2
No	1	0

Table 4.13: Do you think the tool would facilitate software development by making relevant artifacts accessible to all members of the development team? (p-value = 0.5)

Category	Interactive	Background
Yes	3	4
Maybe	1	0
No	0	0

Table 4.14: Rate the usefulness of the links (p-value = 0.5)

Category	Interactive	Background
Most Useful	0	1
Useful	2	2
Somewhat	2	1
Not useful	0	0
Not useful at all	0	0

Table 4.15: Rate the acceptability of time spent to do the recording traceability links (p-value = 0.5)

Category	Interactive	Background
Most Acceptable	1	1
Acceptable	3	2
Somewhat	0	1
Not Acceptable	0	0
Not Acceptable at All	0	0

Table 4.16: Rate the acceptability of time spent on finding relevant traceability links (p-value = 0.5)

Category	Interactive	Background
Most Acceptable	2	1
Acceptable	1	1
Somewhat	1	2
Not Acceptable	0	0
Not Acceptable at All	0	0

Table 4.17: Rate the acceptability for referring time to retrieve and review links (p-value = 0.5)

Category	Interactive	Background
Most Acceptable	2	2
Acceptable	1	1
Somewhat	1	0
Not Acceptable	0	1
Not Acceptable at All	0	0

Table 4.18: How accurate was the traceability links that were captured? (p-value = 0.5)

Category	Interactive	Background
91%-100%	1	2
81-90% Accurate	1	0
71%-80% Accurate	1	1
60%-70% Accurate	1	1
50- 60% Accurate	0	0
Less than 50%	0	0

It was distinguishable that only one user from the Background Rule group mentioned that 11%-%50% of the links were missing; the other seven users from both groups mentioned that only 10% or less of the links were missing.

Table 4.19 shows how the users prioritized the most remarkable aspects of a traceability tool. Friendliness is concerned with how friendly use is the tool, timeliness is concerned with how much time spent in using a feature, and helpfulness is how helpful is the tool with the software development. The data collected noted no statistically significant difference (p-value = 0.5) between Interactive or Background Rules groups. Nonetheless, 50% from both groups selected helpfulness and 37% selected timeliness as the most important aspects in traceability tool.

Table 4.19: Prioritize the friendliness, timeliness and helpfulness of traceability tool (p-value = 0.5)

Category	Interactive	Background
1) Friendliness, 2)Timeliness, 3) Helpfulness	0	1
1) Friendliness, 2)Helpfulness, 3) Timeliness	0	0
1) Timeliness, 2) Friendliness, 3) Helpfulness	1	0
1) Timeliness, 2) Helpfulness, 3) Friendliness	2	0
1) Helpfulness, 2)Friendliness, 3) Timeliness	0	1
1) Helpfulness, 2) Timeliness, 3) Friendliness	1	2

The open ended questions that were asked had valuable answers about improving the tool. The users suggested the following when asked about features: 1) the tool need to incorporate with Unified Modeling Language (UML), complex hierarchal diagrams, and Internet Explorer; 2) functionality to sort links and remove the first recorded link; and 3) a status to indicate which developer is working on a certain component.

The users mentioned the following challenges that they faced: 1) clicking on a component after starting to record; 2) a user indicated that the tool is intuitive but face some difficulties due to some functionality of a hypermedia adapter; 3) Adding links through recording only and not any other functionality; and 4) recording extra noise links.

In regards to the questions that asked about the most they liked in ACTS, the users liked the following: 1) saving time as the recorded links would open on a specific slide in PowerPoint; 2) capturability was simple and linking to a specific part of a file; 3) it was developed using Eclipse; 4) Incorporating with various applications; and 5) associating links directly to a component.

Although one user in the Background Rule Group mentioned that the tool was friendly to use and navigable, another user from the Interactive Rule Group stated that the UI was not friendly to use unnavigable. The Interactive Rule Group did not like the following: 1) capturing all cursor movements and actions, which caused a lot of noise; 2) controlling the navigation of recorded links; and 3) supporting more applications, such as Microsoft OneNote.

Regarding comparing the tool with other existing tools that the users had already used, for the Interactive Rules Group, one user mentioned that ACTS is better than existing tools because ACTS able to trace inside documents and web browsing. Other users indicated that the tool they are using links requirements with Black Box testing, the design document with the component level testing, and the code with the static analysis and the unit test document plan. Additionally, one of the users indicated that ACTS compares favorably with other traceability tools.

For the Background Rules Group, one user mentioned that IBM Jazz is a mature tool that is quite user-friendly and easy to install compare to ACTS. In terms of tool features, ACTS has the advantage of being able to connect to specific slides or paragraphs in PowerPoint and in Word documents, respectively. Another user mentioned that ACTS has more flexibility in mapping to actual project documentation, which is very useful compared to IBM Rational Rose. From the improvement side, ACTS need some enhancements in the filtering. Another user mentioned that ACTS seems fairly intuitive compared to Visual Paradigm based in UML. Lastly, one user mentioned that ACTS allows linking to specific pages of Word documents, cells and worksheets in Excel workbook, and slides on PowerPoint presentation. It is also nice that the links are related to objects in architecture model. The idea of the recording functionality is good since it just keeps track of what the user is doing. However, there is a bit cumbersome since it does not show traced links until clicking on stop recording. Deleting the unnecessary links was an unnecessary overhead.

4.2 Comparison between Non-Traceability and Traceability Users

Previous sections in this chapter independently analyzed the data of non-traceability users and traceability users comparing between the Interactive Rules group and the Background Rules group. In this section, a comparison is done between non-traceability users (n = 30) and traceability users (n = 8). In this section, the null hypothesis is: *there is no statistically significant difference between the feedback provided by users with and without traceability experience at 95% confidence level*. The two-sample t-test is applicable for unequal size and unequal variances and testing the null hypothesis can be easily done without changing the test.

In order to perform the t-test to reject the null hypothesis of unequal size samples that have unequal variances, the standard error of the combined samples is calculated as:

$$S_e = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} \qquad (1)$$

where, S_e is the standard error or what is known as the Satterthwaite Approximation, s_i and n_i are the variance and size of i^{th} sample (i = 1 and 2) [33]. It is important to note that the variance of each sample is normalized with respect to its size, i.e., the terms under the radical in Eq.1. Thereafter, the t-test statistics (t_{stat}) is calculated as:

$$t_{stat} = \frac{x_1' - x_2'}{S_e} \qquad (2)$$

where, x_1' and x_2' are the means of the first and second samples, respectively. Additionally, the t-test statistics (t_{crit}) based on the degree of freedom on the combined sample and significance level ($\alpha = 0.05$) is looked up in any standard students t-distribution table. The Welsh-Satterthwaite degree of freedom (ν) is calculated as shown in Eq.3 [34].

$$\nu = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\left[\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2 - 1}\right]} \qquad (3)$$

If $t_{stat} < t_{crit}$, we fail to reject the null hypothesis based on the available data, while if $t_{stat} \geq t_{crit}$, we can reject the null hypothesis.

Table 4.20 shows the summary of the t-test statistics for comparing the feedback of non-traceability and traceability users for quantitative analysis. A detail analysis of each of the questions listed in the table is included in Appendix D. It is evident from the results that we cannot reject the null hypothesis, which means that there is no statistically significant difference between the feedback from traceability and non-traceability users. In what follows, a summative evaluation is presented to compare the numerical results between the two user groups.

Table 4.20: Summary of t-test statistics for comparing traceability and non-traceability users' feedback

Question	t_{stat}	t_{crit}	Reject Ho?
Would you recommend this tool to your peers?	0.666	1.699	No
Is ACTS more effective than your current technique of tracing?	0.160	1.701	No
Do you think the tool would facilitate software development by making relevant artifacts accessible to all members of the development team?	1.248	1.699	No
On a scale of 1 to 5, rate the usefulness of the links	0.377	1.699	No
On a scale of 1 to 5, rate the acceptability of time spent to do the recording traceability links	0.496	1.699	No
On a scale of 1 to 5, rate the acceptability of time spent on finding relevant traceability links	1.196	1.701	No
Rate the acceptability for referring time to back link retrieving and reviewing	0.096	1.699	No
How accurate was the traceability links that were captured? Use your best estimate	0.582	1.701	No
If you would like to prioritize the friendliness, timeliness and helpfulness of this tool, which of the flowing sequence best describes your experience (1 was most remarkable, 3 was least remarkable)	0.881	6.314	No

In addition to the t-test results discussed beforehand, Tables 4.21-4.27 show summative evaluation of the feedback from the users for the quantitative questions. Generally, the summative evaluation supports the findings from t-test, which indicate that there is no statistically significant difference between the two data sets. The tables show the percentage of responses for each question as each group. Table 4.21 shows the percent of users that would recommend the tool from both groups. For example, 60% of the non-traceability users "Strongly Recommended" or "Recommended" the tool which means that they developed a positive perception about ACTS. On the contrary, 63% of the traceability users selected "Indifferent", that is, they developed neither a positive nor a negative opinion about ACTS.

The users with prior traceability experience mentioned “Indifferent” for recommending the tool to other peers due to some UI issues and the navigation between recorded links. Other users did not like that ACTS captures all the cursor moves in Word document, and clicking on a component after clicking on “Start Recording” button.

Table 4.21: Would you recommend this tool to your peers?

Category	Non-Traceability Users	Traceability Users
Strongly Recommend	13.3%	12.5%
Recommend	46.7%	25.0%
Indifferent	36.7%	62.5%
Do Not Recommend	3.3%	0.0%

Regarding the effectiveness of ACTS technique, Table 4.22 shows the results for the question about the effectiveness of ACTS technique in comparison to the current traceability technique of the user. It is reported that 63% of the users with prior traceability experience and 50% of the users with no prior experience selected “Maybe.” That is due to the same reasons mentioned in the previous section chose “Indifferent” to recommend ACTS to other peers.

Table 4.22: Is ACTS more effective than your current technique of tracing?

Category	Non-Traceability Users	Traceability Users
Yes	33.3%	25.0%
Maybe	50.0%	62.5%
No	16.7%	12.5%

Facilitating the software development process by making relevant artifacts accessible to all members of the development team was a question that had remarkable results, which are shown in Table 4.23. Specifically, 73.3% of the non-traceability users and 87.5% of the traceability users stated “Yes.”

This means that most of the participants in both groups agreed that ACTS tool would facilitate development.

Table 4.23: Do you think the tool would facilitate software development by making relevant artifacts accessible to all members of the development team?

Category	Non-Traceability Users	Traceability Users
Yes	73.3%	87.5%
Maybe	20.0%	12.5%
No	6.7%	0.0%

Table 4.24 - 4.25 show the results for the questions that asked about the usefulness of the links created through the recording process and the acceptability for referring time to retrieve and review links, respectively. Specifically, 100% of the traceability users rated “Most Useful”, “Useful”, or “Somewhat Useful” for the links created through the recording process; on the other side, 86.2% of the non-traceability users rated “Most Useful”, “Useful”, or “Somewhat useful” for the links created with ACTS tool. For the question referring time to retrieve and review, the traceability users rated 87.5% for “Most acceptable”, “Acceptable”, and “Somewhat acceptable” while the non-traceability users rated 96% for “Most acceptable”, “Acceptable”, and “Somewhat acceptable.”

Table 4.24: Rate the usefulness of the links created with ACTS tool

Category	Non-Traceability Users	Traceability Users
Most Useful	20.7%	12.5%
Useful	31.0%	50.0%
Somewhat useful	34.5%	37.5%
not useful	13.8%	0.0%
Not useful at all	0.0%	0.0%

Table 4.24: Rate the acceptability for referring time to retrieve and review links

Category	Non-Traceability Users	Traceability Users
Most Acceptable	50.0%	50.0%
Acceptable	20.0%	25.0%
Somewhat Acceptable	26.7%	12.5%
Not Acceptable	3.3%	12.5%
Not Acceptable at All	0.0%	0.0%

Most notably, Table 4.25 shows the difference between the feedback provided by non-traceability users and traceability users for rating the accuracy of the traceability links that were captured. 76.7% of the non-traceability users rated that 81% or more of the links captured were accurate, while 50% of the traceability users rated that 81% or more of the links captured were accurate. Also, Table 4.27 displays the feedback provided by the two groups for prioritizing the friendliness, timeliness, and helpfulness of the traceability tool. 60.83% of the non-traceability users rated timeliness as most remarkable, while 50% of the traceability users rated helpfulness as most remarkable and 25% rated timeliness as most remarkable as highlighted in Table 4.27.

Table 4.26: How accurate was the traceability links that were captured?

Category	Non-Traceability Users	Traceability Users
91%-100% Accurate	46.7%	37.5%
81-90% Accurate	30.0%	12.5%
71%-80% Accurate	6.7%	25.0%
60%-70% Accurate	10.0%	25.0%
50- 60% Accurate	0.0%	0.0%
Less than 50%	6.7%	0.0%

Table 4.27: Prioritize the friendliness, timeliness and helpfulness of traceability tool

Category	Non-Traceability Users	Traceability Users
1) Friendliness, 2) Timeliness, 3) Helpfulness	22.22%	12.50%
1) Friendliness, 2) Helpfulness, 3) Timeliness	5.56%	0.00%
1) Timeliness, 2) Friendliness, 3) Helpfulness	27.78%	12.50%
1) Timeliness, 2) Helpfulness, 3) Friendliness	33.33%	25.00%
1) Helpfulness, 2) Friendliness, 3) Timeliness	0.00%	12.50%
1) Helpfulness, 2) Timeliness, 3) Friendliness	11.11%	37.50%

The graphs shown in Figure 4.2-4.3 indicate that non-traceability users and traceability users did not have the same behavior. That is, 46% of the non-traceability users rated “Acceptable” and “Most acceptable,” while 75% of the traceability users rated “Acceptable” and “Most acceptable” for the acceptability of the time spent to record traceability links. Likewise, 46.7% of the non-traceability users selected “Acceptable” and “Most Acceptable,” and 75% of the traceability users selected “Acceptable” and “Most acceptable” for the acceptability of the time spent on finding relevant traceability links.

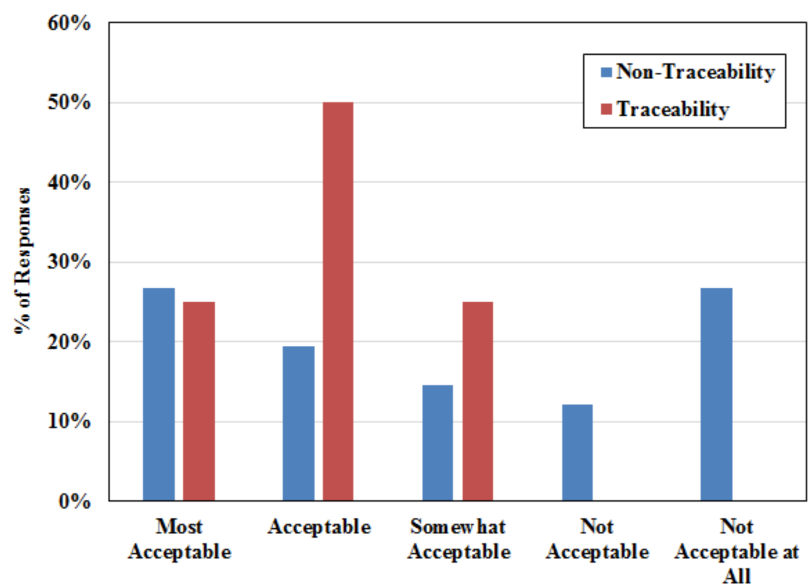


Figure 4.2 Rate the acceptability of the time spent to do recording traceability links.

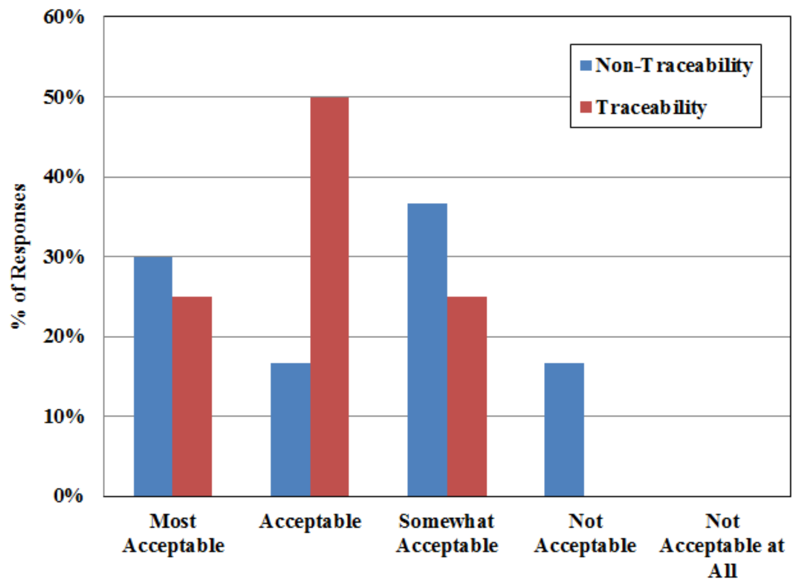


Figure 4.3 Rate the acceptability of the time spent on finding relevant traceability links.

4.3 Developers Survey

This section reports the analysis of the developers data (n = 5). The developers were randomly recruited from University of Washington based on their interest to learn about ACTS and third-party tools API libraries. The survey instrument was reviewed and approved by the human subject committee at UW (IRB#48362). Three out of the five developers stated their primary role as students, while two indicated that they are software engineers. One stated their secondary role as solution architect. Four students developed the software adapter for a no-credit research project, but one other student developed it for a graduate capstone project. The years of experience for the candidates ranged from 3 weeks to 11 years.

All developers did not perform any software development on ACTS prior to creating the software adapters. The third-party tools software adapter that were created using Java language were Google Chrome and Mozilla Firefox. On the other Skype, and Microsoft Visio were created using C#. Lastly, Windows Media Player was created using Visual Basic. Choosing the programming language was dependent on the easiness of integration with ACTS existing code or with the third-party available code.

Despite, one user mentioned that the reason was behind the familiarity with that programming language and easiness of writing functions. Java developers had 0-3 years of experience, while C# developers had 3-4 years of experience, and Visual basic developer had 2 years of experience. All developers had experience in developing software with other programming languages ranged from 2-12 years.

Table 4.28: Quantitative developers' questions

Question	Illustration
Using your best estimate, how many hours did you spend in creating the software adapter?	The hours spent for developing the adapters for no credit research developers ranged from 6-20 hours, for graduate project was 100 hours.
What percentage of this time was spent in learning ACTS?	20% - 90%
What percentage of this time was spent in learning the programming language and the API of the third-party tool?	10% - 100%

Table 4.29 shows the easiness of developing a software adapter for Google Chrome, Skype, and Windows Media Player. The developer of Microsoft Visio adapter rated "Middle," which means that the developer neither developed a positive or a negative idea about the easiness of developing the software adapter. Finally, the developer of Mozilla Firefox adapter rated "Difficult" because facing difficulties in understanding Firefox API's.

Table 4.29: rate ease of developing a software adapter for a third-party tool

Adapter	**Easiness**
Google Chrome	Very easy
Skype	Very easy
Windows Media Player	Very easy

Adapter	Easiness
Microsoft Visio	Middle
Mozilla Firefox	Difficult

Three out of the five developers were able to integrate their software into ACTS, two of which rated the ease of integrating the adapter into ACTS as “Very easy’ and one rated it as fair. Additionally, two developers were not able to integrate it due to the time limitation and the experience of the developers with programming and they rated ease of integrating as “Very difficult” and “difficult.”

It is interesting to note that the developers were split in deciding on the most difficult part in the development of software adapters. Two developers, who were responsible for developing and integrating Chrome and Media Player adaptor, mentioned that integration was challenging. One developer, who was responsible for Skype adaptor, mentioned that learning the API for the tool was difficult due to loss of functionality. The final two developers stated that learning the APIs as well as the integration with ACTS was problematic. On the contrary, only one developer reported that the integration between the tool and ACTS was the easiest part of the development process, while another stated that the support provided by the research team facilitated the adaptor development. Two developers specified that writing the code for API was relatively easy. The final developer, who was involved with Firefox, indicated that the entire development and integration process was challenging. That to say, the perception of the developers about the degree of difficulty or the degree of easiness is highly dependent on their understanding of the API, tool of interest, and ACTS. Table 4.30 summaries the results discussed in this paragraph.

Table 4.30: rate ease of developing a software adapter for a third-party tool

3rd party Tool	Difficult part in the development	Easiest part of the development
Chrome	Integration	Coding Adaptor
Media Player	Integration	Support
Skype	API	Integration
Firefox	API + Integration	None
MS Visio	API + Integration	Coding Adaptor + ACTS

Developers enjoyed learning about a third-party tool, integrating it into ACTS as it was a new technology for them. All developers agreed to develop another adaptor if given the opportunity, while only one stated a condition for further development. The condition was the availability of support for ACTS architecture and documentation.

4.2 Discussion

To this end, the results from feedback provided from two user groups, i.e., non-traceability and traceability experience, and developers of hypermedia adapters for third-party tools were listed and briefly discussed. Important to recall that each users group was further divided into users who evaluated ACTS with filtering rules enabled, i.e., Background Rules, and those with filtering rules disabled, i.e., Interactive Rules and. It was found that there was no statistically significant difference between users with prior traceability experience and users with no traceability experience as well as there was no effect of enabling or disabling the filtering rules on the users' feedback. From the developers' perspective, they enjoyed learning and developing new technologies such third-party tools' hypermedia adapters and integrating it into ACTS. They wanted to work on developing more adapters in the future.

Referring to the Table 3.1 in Chapter 3, from the capturability classification, there were four users' questions that fall under this category. These questions were concerned with capturing traced links and the kind of information captured, specifically for evaluating the effectiveness of ACTS technique, the acceptability of the time spent to record traceability links, the accuracy of the captured traceability links, and the percentage of the missing links captured. That is to say, the capturability is a measure of the technical, social, and economic impact of traceability. For example, measuring the time spent on recording the data can be considered an economic impact since the time resource is limited and generally associated with cost. Also, the accuracy of traces is a measure of the technical functionality of the tool and the social acceptance of such a tool by the users. If a tool effectively and accurately captures all the links during a recording session, as it is the case with ACTS since majority of users irrespective of their experience noted that ACTS capture more than 90% of the links, then the tool will be accepted by users and more likely to be adopted in future projects. A similar argument can be considered for the percentage of missing links, where it is a measure of both the technical and social impacts. Since ACTS was found to be accurate in capturing the link and percentage of missing links were less than 10%, this in turn has economic impact due to the reduction in the time spent in validating and searching traceability links, respectively.

From the accessibility classification, there were four users' questions that fall under this category. These questions were concerned with how the capability of the user to navigate to find traceability links. Two questions asked about how ACTS would facilitate the software development process. Therefore, the accessibility measures the social and economic impact. For example, facilitating the software development process can be considered as an economic and a social factor, as it affects the time spent in developing a task and with how stakeholders interact with each other. It also affects accessibility of the artifacts for the team members in a project. The other two questions asked about the acceptability of the time spent to do recording traceability links, and the accessibility of the time to retrieve and review links. Those questions affect the economic perspective, since the time factor was used as a measure. The accessibility factor is affected by the time spent in recording, retrieving, and reviewing links as well as by the user's navigation to find traceability links.

Also, for the utilizability, there were four users' questions that fall under that category. Two questions were concerned about the ability of the tool to facilitate the software development process by making relevant artifacts accessible to all members of the development team. It was found that, there was no statistically significant difference between the feedbacks provided by both kinds of users, i.e. non-traceability and traceability, regardless of the filtering rules as it pertains to utilizability

From the affordability side, there were five developer's questions that were focusing on the number of hours needed for training the developers and the easiness of integrating the software adapter inside ACTS. It was obvious that the developer who worked on Microsoft Visio adapter needed much more time in the development process than the rest of the developer because it was graduate project. Since the questions mentioned the time spent, therefore, the affordability affects the economic perspective. Two of the affordability questions were also classified under the customizability. These questions were about the ability to integrate the software adapter inside ACTS, which also affected the technical perspective due to the complexity of the software development process. Overall, the easiness of integrating an adapter into ACTS is not dependent on the developer's experience rather on the programming language used to develop the adapter. This was obvious based on the feedback reported by the developer, where those used C# and Visual Basic developers rated "Very easy", while developers used Java rated "Very difficult" and "difficult". Additionally, the integration of third-party tools is dependent on the type software adapter. For example, Google Chrome and Mozilla Firefox were difficult to integrate within ACTS, which gives an impression that an improvement need to be done regarding how ACTS interact with browsers. From the customizability side, rating the easiness of developing a software adapter for a third-party tool relied on the availability of online resources about APIs.

In the next section of the thesis, the conclusion is included and followed by suggestions on future work as it relates to the work presented herein.

CHAPTER 5: CONCLUSION AND FUTURE WORK

This chapter includes the conclusion based on the research presented in the previous chapters. The research seeks to answer the thesis questions in Section 1.1. The approach was to design and implement two case studies for traceability stakeholders: users and developers. The hypothesis for case study to evaluate the feedback of the users is: ACTS supports capturablity, accessibility, and utilizability from economic, social, and technical perspectives. Similarly, the hypothesis for the developers is: ACTS technique encourages developers to create affordable and customizable adapters to facilitate software traceability from economic, social, and technical perspectives. The short responses to both research questions were “Yes”. Detailed conclusion is provided below. This chapter also includes suggestions for future work based on the feedback provided by the users and the developers.

5.1 Conclusion

This thesis evaluated ACTS technique through the design and administration of case studies to qualitatively and quantitatively measure the perception of different stakeholders, i.e., users and developers. The case study for the users consisted of one-on-one meeting with each participant, at the end of which the participants completed a survey. On the other hand, the case stud for the developers involved the development and integration of third-party tool adapters into ACTS and completion of a survey. All survey instruments were reviewed and approved by Human Subject Committee at University of Washington Bothell. The survey questions tackled three specific perspectives: economic, social, and technical of software traceability with the emphasis on capturability, accessibility, and utilizability from the users’ viewpoint; and the emphasis in affordability, and customizability from the developers’ viewpoint.

From the standpoint of the users, the research compared the feedback from four different user groups about ACTS technique. The pool of users was first divide into two groups: one group had no prior traceability experience with a total of 30 users and the other had prior traceability experience with a total of eight users. Each of these groups was further divided into two subgroups to evaluate the filtering functionality in ACTS. The latter were the Interactive Rules and Background Rules subgroups. From the viewpoint of the developers, the research recruited five developers that worked on developing different hypermedia adapters for third-party tools. These adapters were also integrated into ACTS. The users and the developers responded to different surveys as indicated above. One-tailed paired-samples t-test was used to evaluate the null hypothesis of no statistically significant difference between the feedbacks provided by the two user groups at 95% confidence interval. It was found that there is no statistically significant difference between feedbacks provided by all the users regardless of their prior experience as well as the enabling or disabling of filtering rules.

Overall, the users pointed out that ACTS facilitated capturability, accessibility, and utilizability of traceability links. From the economic perspective, it saves time and cost in finding related artifacts. From the social perspectives, it facilitate interaction between users, but it needs improvements form the user friendliness side. From the technical perspective, the accuracy of captured links is high, but it still needs improvements to remove the noise for the interactive rules (manual filtering). On the other hand, developers pointed out that ACTS facilitates affordability and customizability. From the economic perspective third-party tools are easy and fast to learn, but it needs improvements in integration as it required time. From the social perspective, developers enjoyed learning new technologies and wanted to work on ACTS in the future. From the technical perspective, developers pointed out that there need to be architectural design and documentation for ACTS code to facilitate the integration process. Therefore, ACTS is considered customizable and affordable from the technical more than customizable from the economic perspective.

5.2 Future Work

As a result of the users and developers case studies, it was found that ACTS tool needs some improvements. The users indicated that some development could be done to make the tool more user-friendly use side. The developers reported that documentation could be improved to facilitate development of hypermedia adapters.

The followings summarize the suggested future work that could be introduced to ACTS:

1) The links captured could be refreshed automatically after each recording session or creating a refresh button.
2) Creating user-friendly UI for manual filtering and navigating recorded links.
3) Adding functions that will not require the user to click on the component in the architecture after clicking on “Start Recording” button.
4) The third-party tools should be activated automatically after opened by ACTS tool.
5) Adding a machine learning technique that would delete unnecessary links captured.
6) Adding more software adapters to ACTS such as supporting more browsers other than Firefox, and also supporting more applications like Microsoft OneNote, and Google Home and Office applications.
7) Updating Firefox adapters.
8) Updating ACTS to work with the latest Archstudio version.
9) ACTS could incorporate with Unified Modeling Language (UML), complex hierarchal diagrams.
10) Creating documentation for ACTS code on how to develop adapters for third-party tools and integrate them into ACTS.
11) Refactoring some of the backend code.

Finally, new case studies for the users are need to be designed and implemented to evaluate other features in ACTS such as evaluating the functionality of the following buttons: “Recover”, “Manual”, “Import Links”, “Export Links”, “Trace Analysis”, “Trace Preference”, and “Specify adapters”. Also, case studies for developers could be designed to evaluate developing more adapters and refactoring ACTS code.

APPENDICES

A) Users' Survey Instruments Questions

A1. Users Survey Questions:

Questions1:
Primary Role or Occupation

Question 2
How long have you held this role?

Question 3
Secondary Role or Occupation

Question 4
How long have you held this role?

Question 5
What type of software does your organization develop?

Question 6
Please specify the operating system that you use.

- ○Windows
- ○Linux
- ○Unix
- ○Mac OS

Question 7
Have you used Eclipse before?

- ○Yes
- ○No

Question 8
If yes, then for how long?

- ○0-1 months
- ○1-3 months
- ○3-6 months
- ○6 or more months

Question 9
Describe your ideal traceability tool?

Question 10
How do you compare the tool with other existing tools? If you haven't used any other traceability tool, then please write your impression about ACTS tool.

Question 11
Would you use this tool for future projects? Why or Why not?

Question 12
Would you recommend this tool to you peers?

- ◯ Strongly Recommend
- ◯ Recommend
- ◯ Indifferent
- ◯ Do Not Recommend

Question 13
Is ACTS more effective than your current technique of tracing?

- ◯ Yes
- ◯ Maybe
- ◯ No

Question 14
Do you think the tool would facilitate software development by making relevant artifacts accessible to all members of the development team?

- ◯ Yes
- ◯ Maybe
- ◯ No

Question 15
How?

Question 16
On a scale of 1 to 5, rate the usefulness of the links created with the ACTS tool.

- ◯ (1)Most useful
- ◯ (2)
- ◯ (3)somewhat useful
- ◯ (4)
- ◯ (5)Not useful at all

Question 17
On a scale of 1 to 5, rate the acceptability of time spent to do the recording traceability links.

- (1)Most acceptable
- (2)
- (3)Somewhat acceptable
- (4)
- (5)Not acceptable at all

Question 18
On a scale of 1 to 5, rate the acceptability of time spent on finding relevant traceability links.

- (1)Most acceptable
- (2)
- (3)somewhat acceptable
- (4)
- (5)Not acceptable at all

Question 19
Rate the acceptability for referring time to back link retrieving and reviewing.

- (1)Most acceptable
- (2)
- (3)Somewhat acceptable
- (4)
- (5)Not acceptable at all

Question 20
How accurate was the traceability links that were captured? Use your best estimate.

- 91%-100% accurate
- 81-90% accurate
- 71%-80% accurate
- 60%-70% accurate
- 50%-60%
- Less than 50%

Question 21
Were there missing links? Use your best estimate.

- 91%-100% of the links were missing
- 81%-90% of the links were missing
- 71%-80% of the links were missing

- 61%-70% of the links were missing
- 51%-60% of the links were missing
- 11%-50% of the links were missing
- 10% of the links or less were missing

Question 22
Did you use your captured traceability links while you were performing development task(s)? If so, please specify the task(s).

Question 23
If you would like to prioritize the friendliness, timeliness and helpfulness of this tool, which of the flowing sequence best describes your experience (1 was most remarkable, 3 was least remarkable).

- 1)Tool is user-friendly. 2)Tool minimizes the time spent to do traceability tasks. 3)Tool helps with developing the tasks.
- 1)Tool is user-friendly. 2)Tool helps with developing the tasks. 3)Tool minimizes the time spent to do traceability tasks.
- 1)Tool minimizes the time spent to do traceability tasks. 2)Tool is user-friendly. 3)Tool helps with developing the tasks
- 1)Tool minimizes the time spent to do traceability tasks. 2)Tool helps with developing the tasks. 3) Tool is user-friendly
- 1)Tool helps with developing the tasks. 2)Tool is user-friendly. 3)Tool minimizes the time spent to do traceability tasks.
- 1)Tool helps with developing the tasks. 2)Tool minimizes the time spent to do traceability tasks. 3)Tool is user-friendly.

Question 24
Are there features in a traceability tool that were not present in ACTS? If so, what are they?

Question 25
Did you encounter any challenges when using the tool? If so what are their problems?

Question 26
What functionality (ies) in the ACTS tool did you use?

Question 27
What do you like most about the ACTS tool? Why?

Question 28
What do you not like about the ACTS tool? Why?

The following table includes the question number which is the order in the survey questions, question word that showed exactly to the user, question type to clarify if it is qualitative or quantitative, perspective to illuminate if it is economic, social, or technical. Finally, an explanation that which has two bullets which are describing the reason behind question's type, and the reason behind the perspective's type.

A2. Users Perspective Questions

Table A.1 includes the question numbers in the survey instrument, question wording that showed exactly to the user, question type to clarify if it is qualitative or quantitative, and perspective to illuminate if it is economic, social, or technical. Finally, explanations are describing the reason behind question's type, and the reason behind the perspective's type.

TableA.1: Economic, Social, and Technical Questions

Question Number	Question Word	Question Type	Perspective	Explanation
9	Describe your ideal traceability tool?	Qualitative	Social	Describing the ideal traceability tool intended to solicit user's thoughts about a tool that would impact their usage and that will help to improve the quality of ACTS tool, so it affected the social impact
10	How do you compare the tool with other existing tools? If you haven't used any other traceability tool, then please write your impression about ACTS tool.	Qualitative	Social	Comparing the tool with other competitive tools or writing the impression about the tool provided different interactions of competitive tools and different expectations of the user, so it affected the social impact
11	Would you use this tool for future projects? Why or Why not?	Qualitative	Social	Using ACTS for future project gives an impression on whether it improved the process it will improve the behavior of the work or not, so it affected the social impact
12	Would you recommend this tool to your peers? • Strongly Recommend • Recommend	Quantitative	Social	Recommending the tool to other peers will encourage other users to use the tool, that allows more interaction between users, so the

Question Number	Question Word	Question Type	Perspective	Explanation
	• Indifferent • Do Not Recommend			question affected the social impact
13	Is ACTS more effective than your current technique of tracing? • Yes • Maybe • No	Quantitative	Social	Asked about comparing the efficiency of the tool with other personal techniques that the user follows. The question allowed the user to think about the most familiar technique of tracing, so it affected the social impact
14	Do you think the tool would facilitate software development by making relevant artifacts accessible to all members of the development team? • Yes • Maybe • No	Quantitative	Social, Technical	Asked about the easiness of the software development process affecting all the team members by making relevant artifacts accessible, so it affected the social impact. At the same time it affected the technical perspective because it asked if the tool would facilitate the software development process.
15	How?	Qualitative	Social, Technical	The user had to explain how ACTS facilitated the software development process or not. The user will brainstorm about scenarios to use ACTS tool affecting human interaction and efficiency, so this question affected the social and technical question
16	On a scale of 1 to 5, rate the usefulness of the links created with the ACTS tool. 1 Most useful 2 3 Somewhat useful 4 5 Not useful at all	Quantitative	Social, Technical	The user rated how useful the captured links that connects all relevant artifacts to all members of team to access it easily. It could improve tasks accomplished every day and the behavior of work, so the question affected the social perspective. At the same time, it affected the technical perspective because it would affect simplicity of development task through the usefulness of traced links.

Question Number	Question Word	Question Type	Perspective	Explanation
17	On a scale of 1 to 5, rate the acceptability of time spent to do the recording traceability links. 1 Most acceptable 2 3 somewhat acceptable 4 5 Not acceptable at all	Quantitative	Economic	The question had the time factor spent in recording traceability links, so it affected the economic perspective
18	On a scale of 1 to 5, rate the acceptability of time spent on finding relevant traceability links. 1 Most acceptable 2 3 somewhat acceptable 4 5 Not acceptable at all	Quantitative	Economic	The question had the time factor in finding relevant traceability links, so it affected the economic perspective
19	Rate the acceptability for referring time to back link retrieving and reviewing. 1 Most acceptable 2 3 somewhat acceptable 4 5 Not acceptable at all	Quantitative	Economic	The question had the time factor for referring traced links and the ability of time spent in retrieving and reviewing traced links, so it affected the economic perspective
20	How accurate was the traceability links that were captured? Use your best estimate. 91%-100% accurate 81%-90% accurate 71%-80% accurate 60%-70%accurate 50%-60%accurate Less than 50%	Quantitative	Technical	The question evaluated the accuracy of the links captured which allowed the user to check the correctness of the traced links. So this questions affected the technical perspective

Question Number	Question Word	Question Type	Perspective	Explanation
21	Were there missing links? Use your best estimate. 91%-100% of the links were missing 81%-90% of the links were missing 71%-80% of the links were missing 60%-70% of the links were missing 50%-60% of the links were missing 10% of the links were missing	Quantitative	Technical	The question evaluated the estimated missing links which allowed the user to check if all the visited links were captured. The less the missing links, the easier it makes the development process. So this questions affected the technical perspective
23	If you would like to prioritize the friendliness, timeliness and helpfulness of this tool, which of the flowing sequences best describes your experience (1 was most remarkable, 3 was least remarkable). 1) Tool is user-friendly. 2)Tool minimizes the time spent to do traceability tasks. 3)Tool helps with developing the tasks. 1)Tool is user-friendly. 2)Tool helps with developing the tasks. 3)Tool minimizes the time spent to do traceability tasks. 1)Tool minimizes the time spent to do traceability tasks. 2)Tool is user-friendly. 3)Tool helps with developing the tasks 1)Tool minimizes the time spent to do traceability	Quantitative	Economic, Social, Technical	Asked about user-friendliness affected the social perspective; asked about timeliness affected the economic perspective; asked about the helpfulness affected the technical perspective

Question Number	Question Word	Question Type	Perspective	Explanation
	tasks. 2)Tool helps with developing the tasks. 3) Tool is user-friendly 1)Tool helps with developing the tasks. 2)Tool is user-friendly. 3)Tool minimizes the time spent to do traceability tasks. 1)Tool helps with developing the tasks. 2)Tool minimizes the time spent to do traceability tasks. 3)Tool is user-friendly.			
24	Are there features in a traceability tool that were not present in ACTS? If so, what are they?	Qualitative	Technical	Asked the user if there were features need to be added to ACTS. This question will give further information about features that need to be added to ACTS, so it affected the technical perspective
25	Did you encounter any challenges when using the tool? If so, what are their problems?	Qualitative	Social and Technical	The challenges and problems that the user faced while using the tool could affect the software development life cycle efficiency and the collaboration between the team members, so it affected the social impact. At the same time, challenges refer to the complexity of using the tool, so it affected the technical perspective

Question Number	Question Word	Question Type	Perspective	Explanation
27	What do you like most about the ACTS tool? Why?	Qualitative	Social, Technical	Allowed the user to solicit thoughts about features liked, and that will be a motivation to use ACTS, so it affected the social impact. At the same time, the user could mention functionalities that affect the complexity of the software development process
28	What do you not like about the ACTS tool? Why?	Qualitative	Social and Technical	Allowed the user to solicit thoughts about what wasn't motivating in the tool and about the features he/she didn't like, so it affected the social perspective. At the same time, the user could mention functionalities that affect the complexity of the software development process

B) Developers' Survey Instruments Questions

B1. Developers Questions

Question 1
Primary Roe or Occupation:

Question 2
How long have you held this role?

Question 3
Secondary Role or Occupation:

Question 4
How long have you held this role?

Question 5
Did you perform any development on the ACTS traceability tool prior to creating the software adapters?
○Yes
○No

Question 6
What programming language did you use for your software adapter?

Question 7
Why did you choose this language?

Question 8
How many years of experience do you have in developing software with the above programming language?

Question 9
How many years of experience do you have in developing software with another programming language?

Question 10
In what context did you develop the software adapters?
○Undergraduate research (CSS499)
○Undergraduate capstone project (CSS497)
○Graduate project (CSS595/596)
○Other (please specify):

Question 11
For which third-party tool did you create the software adapter?

Question 12
Using your best estimate, how many hours did you spend in creating the software adapter?

Question 13
What percentage of this time was spent in learning ACTS?

- ◯100% of the time
- ◯90% of the time
- ◯80% of the time
- ◯70% of the time
- ◯60% of the time
- ◯50% of the time
- ◯40% of the time
- ◯30% of the time
- ◯20% of the time
- ◯10% of the time

Question 14
What percentage of this time was spent in learning the programming language and the API of the third-party tool?

- ◯100% of the time
- ◯90% of the time
- ◯80% of the time
- ◯70% of the time
- ◯60% of the time
- ◯50% of the time
- ◯40% of the time
- ◯30% of the time
- ◯20% of the time
- ◯10% of the time

Question 15
On a scale of 1 to 5, rate ease of developing a software adapter for a third-party tool.

- ◯1 Very easy
- ◯2
- ◯3
- ◯4
- ◯5 Very difficult

Question 16
Was your software adapter able to integrate with the ACTS tool?

○ Yes

○ No

Question 17
If not, why not?

Question 18
On a scale of 1 to 5, rate ease of integrating the third-party tool into ACTS using the software adapter you developed.

○ 1 Very easy

○ 2

○ 3

○ 4

○ 5 Very difficult

Question 19
Please explain your documentation or notes (if available)

Question 20
Did you use any references (books, tutorials, etc.) in creating your documentation (if available)?

Question 21
What is the most difficult part of developing your software adapter?

Question 22
What is the easiest part of developing your software adapter?

Question 23
What do you like about developing the software adapter? Why or why not?

Question 24
What do you not like about developing the software adapter? Why or why not?

Question 25
Would you develop another software adapter? Why or why not?

Question 26
Any other suggestions/comments?

B2. Developers Perspectives Questions

Table B.1 shows the question number which is the order in the survey questions, the question word that showed exactly to the user, the question type if it is qualitative or quantitative, and the perspective if it is economic, social, or technical. Lastly an explanation that includes two bullets which are describing the reason behind question's type, and the reason behind the perspective's type

Table B.1: Developers' Economic, Social, and Technical Questions

Question Number	Question Word	Question Type	Perspective	Explanation
5	Did you perform any development on the ACTS traceability tool prior to creating the software adapters? • Yes • No	Quantitative	Economic, Social	The developer's familiarity with the ACTS tool would affect the time consumed in development, so the question affected the economic perspective. At the same time, it affected the social perspective because of the familiarity of the developer with ACTS tool would affect the motivation of working on the tool
6	What programming language did you use for your software adapter?	Qualitative	Technical	Asked about technology used in the development affected the technical perspective
7	Why did you choose this language?	Qualitative	Social	Asked about the reason behind choosing the programming language used affected how easy and familiar the developer is with the programming language, so it affected the social perspective
8	How many years of experience do you have in developing software with the above programming language?	Qualitative	Economic, Social	Asked about the number of years of experience in the programming language used affected the consumed time in developing the adapters, so it affected the economic perspective. At the same time, it affected the familiarity of the developer with the programming language, so it affected the social perspective
9	How many years of experience do you have in developing software with another programming language?	Qualitative	Economic, Social	Asked about the number of years of experience in the programming language used affected the consumed time in developing adapters, so it affected the

Question Number	Question Word	Question Type	Perspective	Explanation
				economic perspective. At the same time it affected the familiarity of the developer with the programming language, so it affected the social perspective
11	For which third-party tool did you create the software adapter?	Qualitative	Technical	Asked about which third-party technology used affected the technical perspective
12	Using your best estimate, how many hours did you spend in creating the software adapter?	Qualitative	Economic	Including time factor spent in creating the software adapter affected the economic perspective
13	What percentage of this time was spent in learning ACTS? 1) 100% of the time 2) 90% of the time 3) 80% of the time 4) 70% of the time 5) 60% of the time 6) 50% of the time 7) 40% of the time 8) 30% of the time 9) 20% of the time 10) 10% of the time	Quantitative	Economic	Including the time factor for learning ACTS affected the economic perspective
14	What percentage of this time was spent in learning the programming language and the API of the third- party tool? 1) 100% of the time 2) 90% of the time 3) 80% of the time 4) 70% of the time 5) 60% of the time 6) 50% of the time 7) 40% of the time 8) 30% of the time 9) 20% of the time 10) 10% of the time	Quantitative	Economic	Including the time factor for learning the programming language and the API affected the economic perspective
15	On a scale of 1 to 5, rate ease of developing a software adapter for a third-party tool 1 Very Easy	Quantitative	Technical, Economic	The easiness of development affected the complexity which tackles technical perspective, and also affected the hours spent in development, so it tackled the

Question Number	Question Word	Question Type	Perspective	Explanation
	2 3 4 5 Very difficult			economic perspective too
16	Was your software adapter able to integrate with the ACTS tool? • Yes • No	Quantitative	Technical, Economic	Integration of the software adapter affected the technical perspective, and also affected the number of hours spent to integrate the adapter into ACTS, so it tackled the economic perspective too
17	If not, why not?	Qualitative	Technical	Explaining why wasn't the software integrated with ACTS affected the technical and economic perspective as the user would explain complexity and the also, the time limit could be a factor
18	On a scale of 1 to 5, rate ease of integrating the third-party tool into ACTS using the software adapter you developed. 1 Very Easy 2 3 4 5 Very difficult	Quantitative	Technical, Economic	Rated the easiness of integration of the third-party tool into ACTS affected the complexity of integration which tackled technical perspective. At the same time, it affected the economic perspective because the easiness of integrating third-party tool could affect the number of hours spent in integration
21	What is the most difficult part of developing your software adapter?	Qualitative	Economic, Technical	Asked about the most difficult part of development which affected the complexity, so the question affected the technical perspective. At the same time, the difficulty would affect the amount of time consumed to develop a software adapter, so it affected the economic perspective

Question Number	Question Word	Question Type	Perspective	Explanation
22	What is the easiest part of developing your software adapter?	Qualitative	Economic, Technical	Asked about easiest part of development affected the complexity of the development process, so it affected the technical perspective. At the same time the easiness would affect the amount of time consumed to develop a software adapter, so it affected the economic perspective
23	What do you like about developing the software adapter? Why or why not?	Qualitative	Social, Technical	Asked about what the developer liked allowed the user to write about the motivating, encouraging and enjoyable parts in the development process, so this question affected the social perspective. At the same time the developer will mention technical experience gained, so this question affected the technical perspective
24	What do you not like about developing the software adapter? Why or why not?	Qualitative	Social, Technical	Asked about what the user didn't like which allowed the user to write about the demotivating, discouraging, and unenjoyably parts in the development process, so it affected the social perspective. At the same time, it affected the technical perspective because the user could mentions about the complexity of the development process
25	Would you develop another software adapter? Why or why not?	Qualitative	Social, Technical	Asked about the developer's motivation in developing other adapters in the future affected the social perspective. At the same time, the user would could mention about the requirement conditions needed to develop another software adapter, so it affected the technical perspective

Question Number	Question Word	Question Type	Perspective	Explanation
26	Any other suggestions/comments?	Qualitative	Economic or/and Social or /and Technical	Asked about the suggestions/comments allowed the developer to write in comments that could affect any of the economic, social, or technical perspective

C) How to Use ACTS Tool

The researcher met individually with each user in a face-to-face style meeting, either in-person or on-line. The session with each user lasted approximately from 30 to 45 minutes, where an introduction to the study, a quick tutorial to ACTS, and instructions about how to complete the survey were given to each user. The following are the steps of the study.

1. The user had to be given a username and password to access the remote machine.
2. Once the user accessed the remote machine, the user would find Archstudio 4 opened and a XADL file (Architecture) opened as shown in Figure C.1. It is a huge architecture of Archstudio 4 system; it's used for this research for testing ACTS to show its purpose. As for this system, there would be probably a long Word document, PowerPoint presentation, Excel Sheets and a lot of Mozilla Firefox links related to that system.

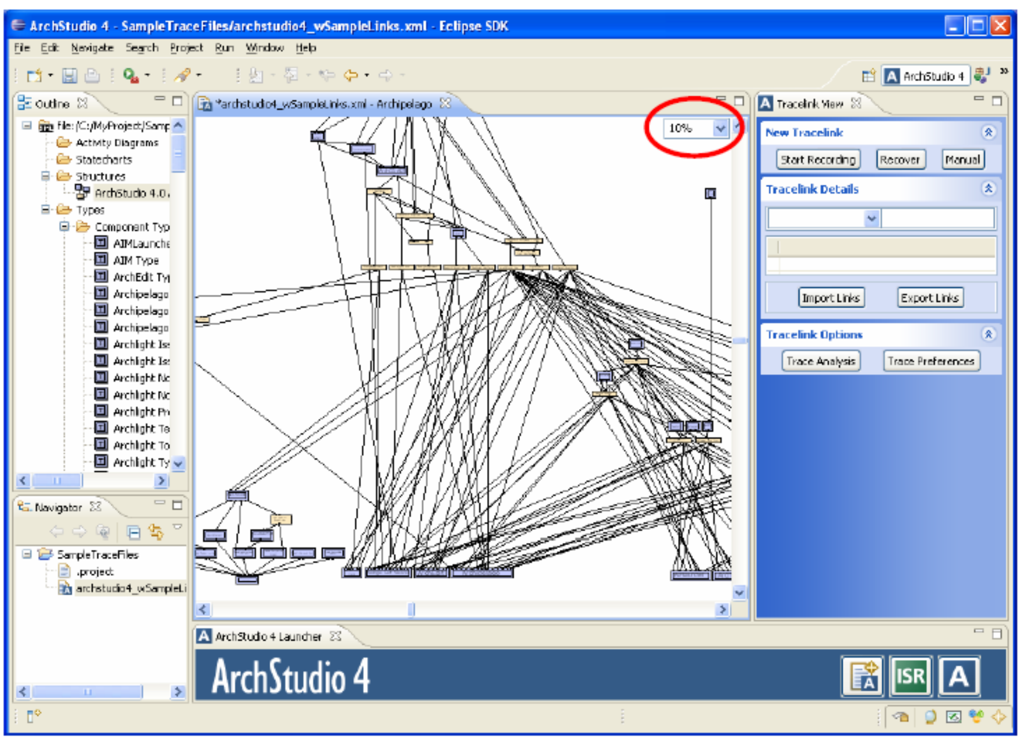


Figure C.1: Screenshot Step 2

3. The user viewed the trace links associated with the component, then clicked on any of the components, represented by a blue box. The link table would populate with links as shown in Figure C.2.

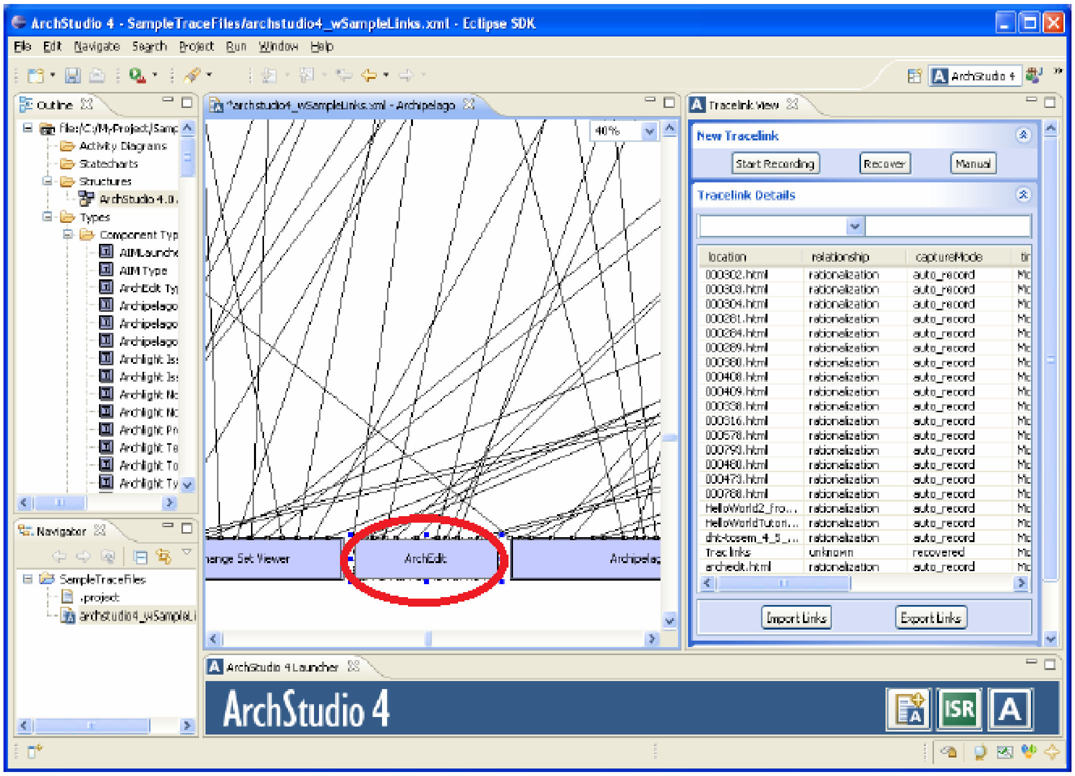


Figure C.2: Screenshot Step 3

4. The user double-clicked on a link on the table. The resource would be displayed within Firefox browser as shown in Figure C.3.

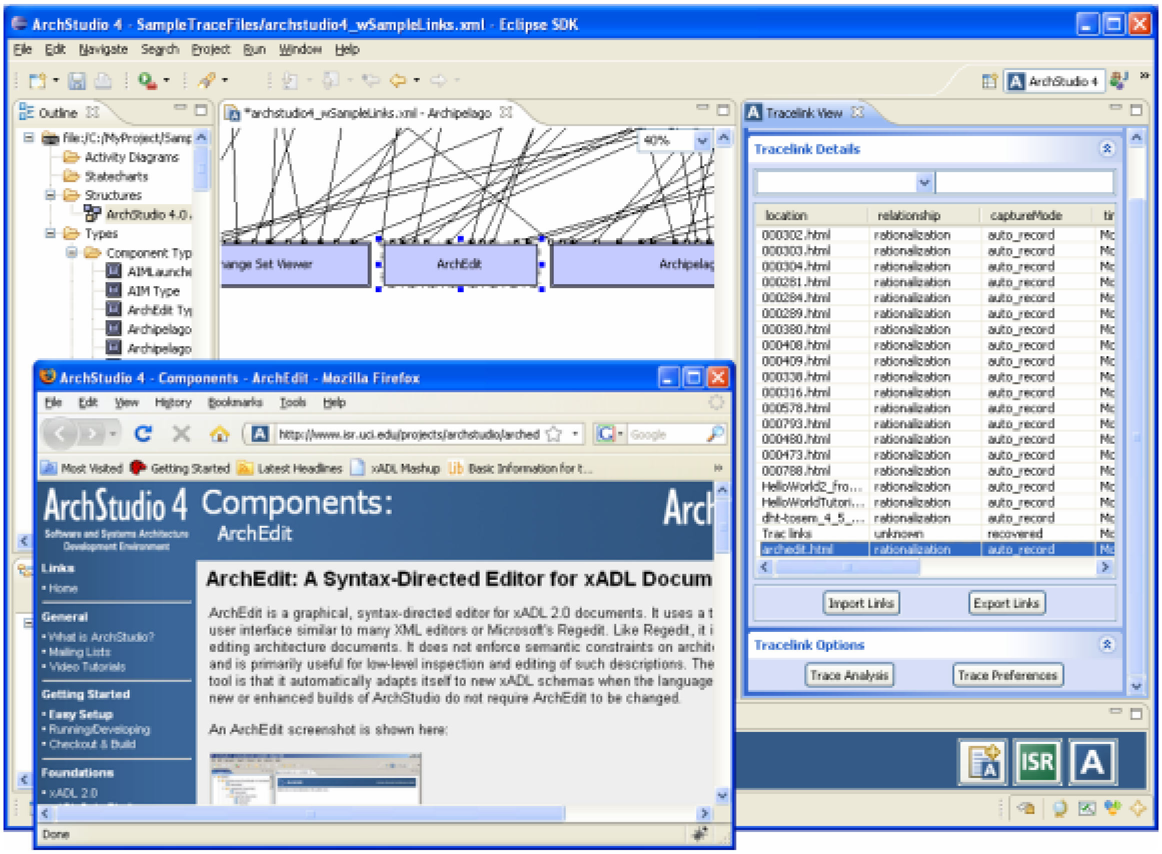


Figure C.3: Screenshot Step 4

5. To associate links to a component, the user clicked on “Start Recording” as shown in Figure C.4.

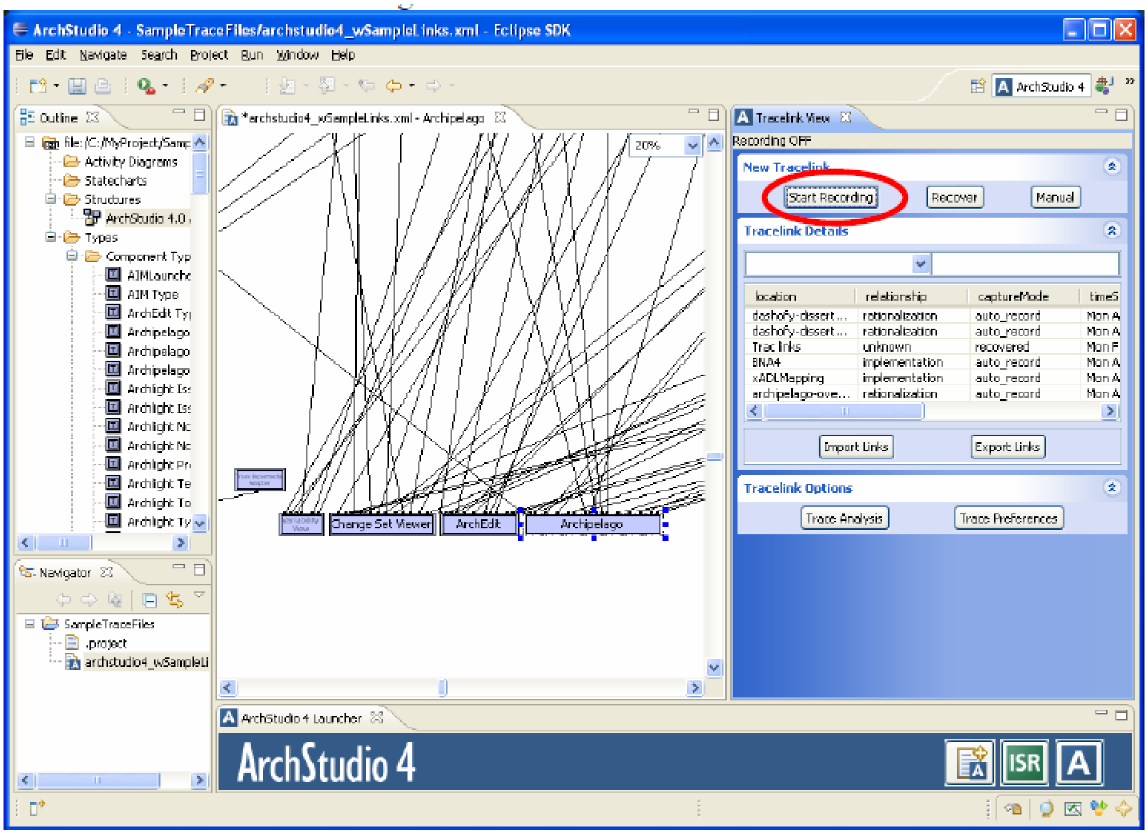


Figure C.4: Screenshot Step 5

6. The user double-clicked or clicked on a component, such as Archipelago, as shown in Figure C.5.

   The user had to make sure it's clicked.

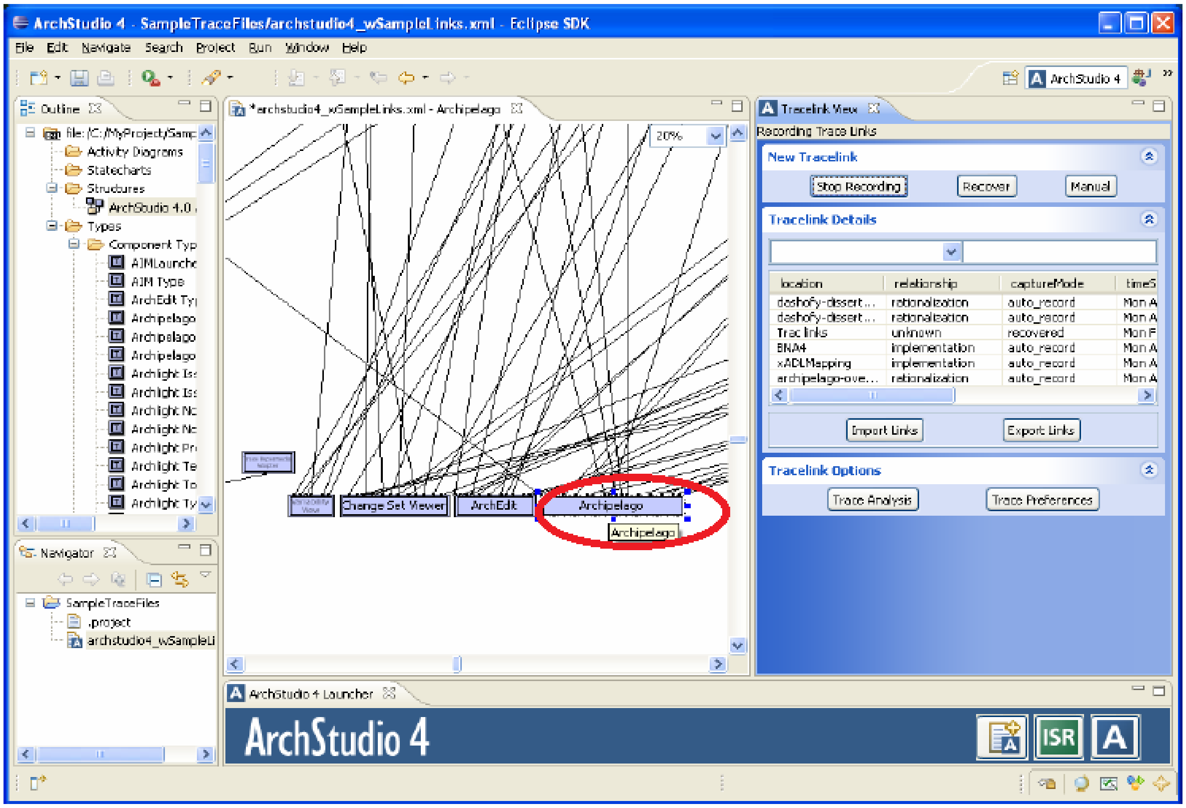


Figure C.5: Screenshot Step 6

7. To record internet links, the user followed these steps:

    a) The user opened the Firefox browser from the start menu in windows. Then opened any website for example: Google (www.google.com) search engine, then search for "archstudio-dev archipelago" or access any website as shown in Figure C.6. The Firefox adapters captured any link visited.

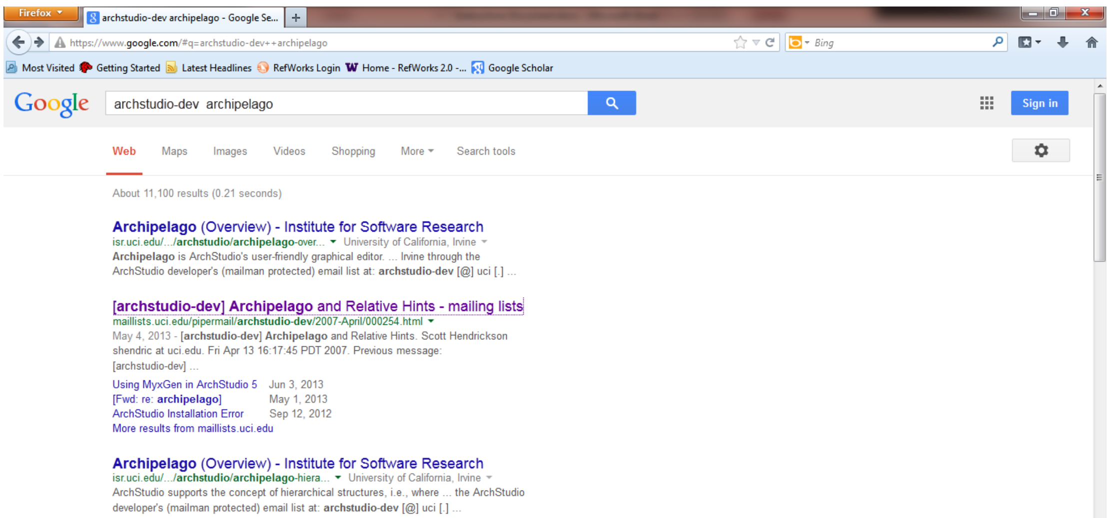


Figure C.6: Screenshot Step 7 (a)

b) The user clicked on a link as shown in Figure C.7

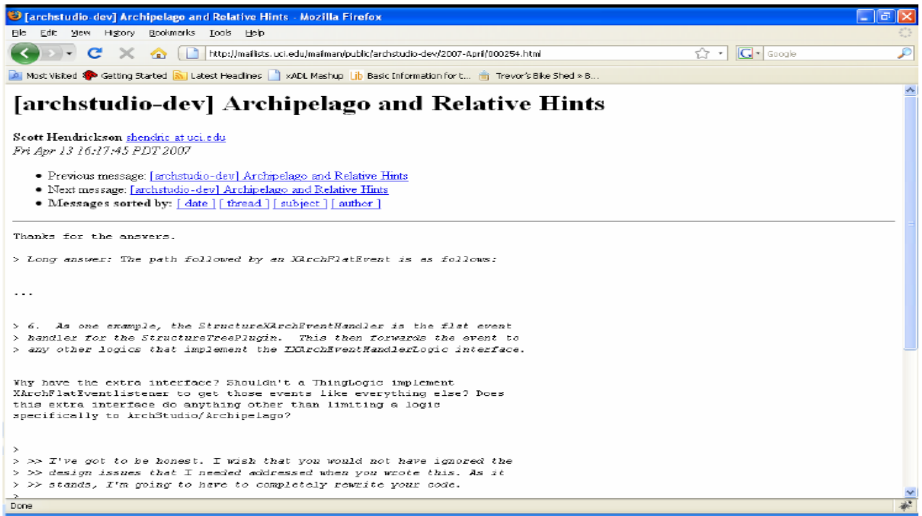


Figure C.7: Screenshot Step 7 (b)

c) The user closed the Firefox browser, if the user was done recording, then they jumped to Step 9 to stop recording. If the user wanted to record links within Word, Excel, or PowerPoint then they proceeded to Step 8.

8. To record links for Word, Excel, PowePoint, the user followed these steps:

   a. The user clicked on the dotted button near "Stop Recording" as shown in Figure C.8.

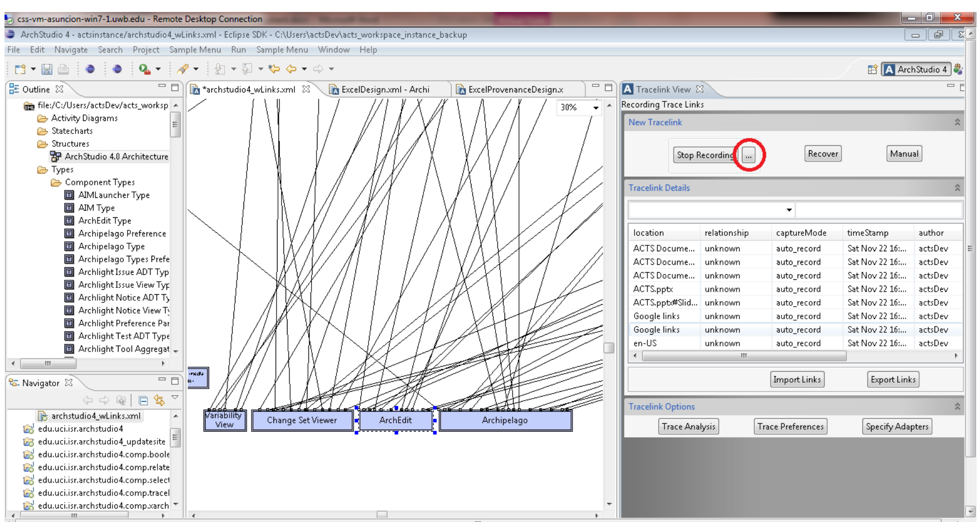


Figure C.8: Screenshot Step 8 (a)

b. A window appeared saying “Enter a URL or select a file to open:” The user entered a URL (path to the file) or clicked on the dotted button on that form as shown in Figure C.9.

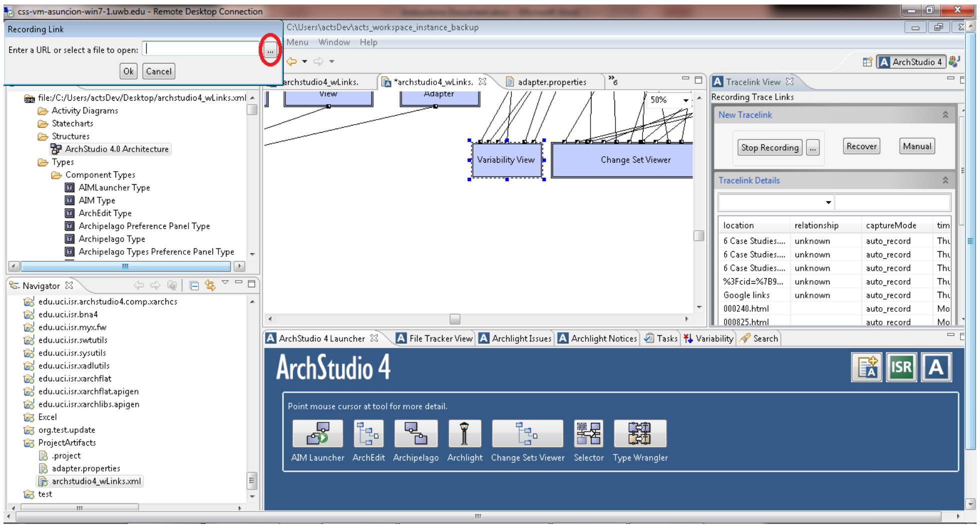


Figure C.9: Screenshot Step 8 (b)

c. The user could browse to the file that they wanted to use. The user could browse to desktop and choose a document for example: ‘ACTS documentation’ as shown in Figure C.10, then they clicked ‘ok’ as shown in Figure C.11.

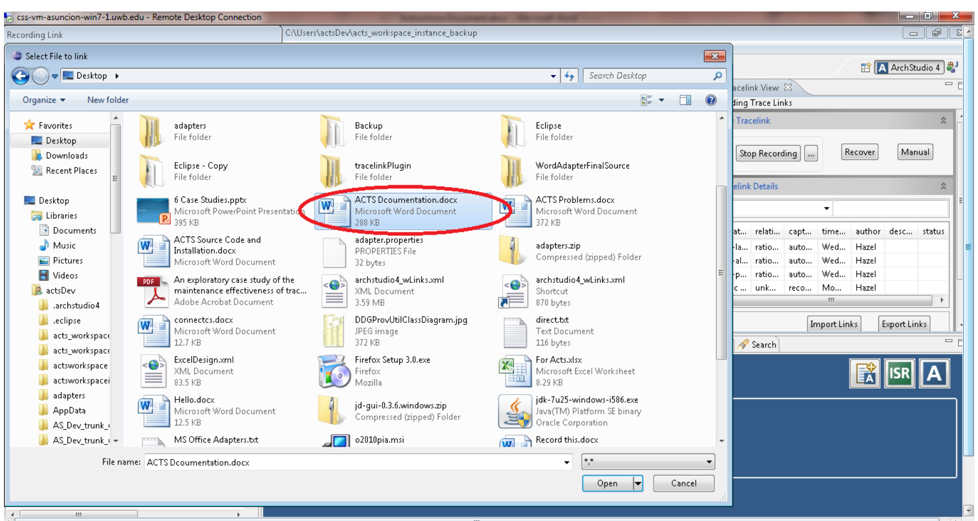


Figure C.10: Screenshot Step 8 (c)

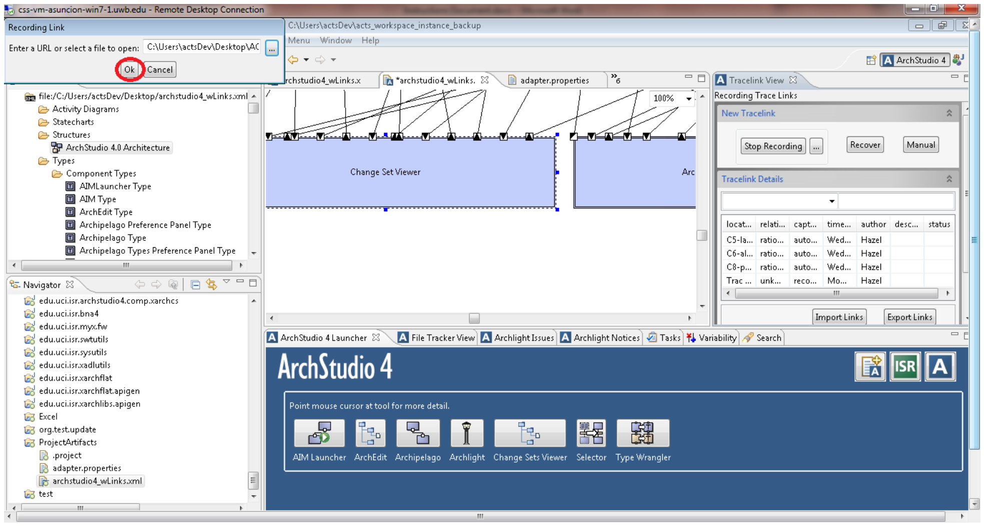


Figure C.11: Screenshot Step 8 (c) "ok"

d. Word is opened. The user needed to maximize the Word document to be able to change anything in the content or to click the cursor on the specific parts they wanted to be linked to the doc. After making changed, the user saved and closed the document as shown in Figure C.12. If the user was done recording, then they moved to Step 9. If they wanted to record links for PowerPoint, or Excel, then they repeated the Steps a-d, and selected a PowerPoint, or Excel in Step c instead of a Word document.

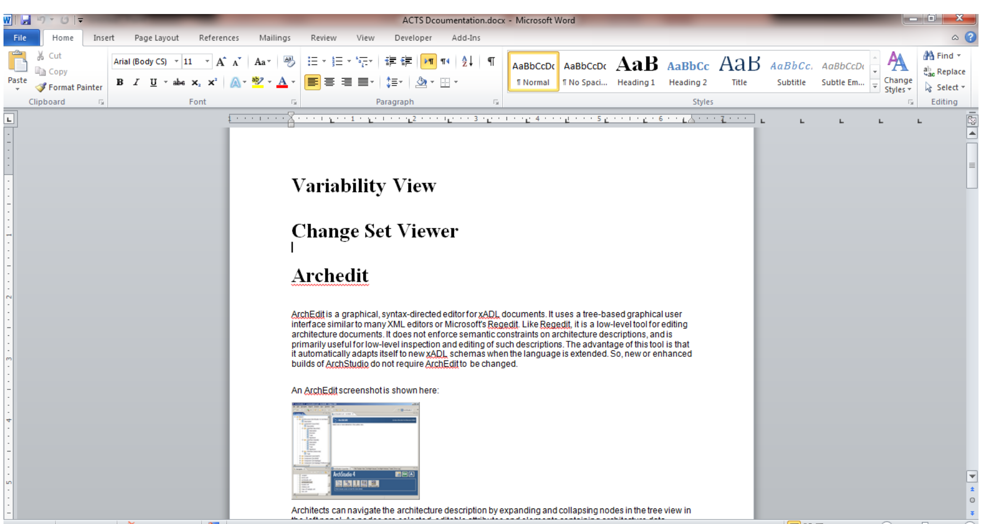


Figure C.12: Screenshot Step 8 (d)

9. To stop recording, the user clicked on Stop Recording as shown in Figure C.13. Then follow the steps in Figures C.14, C.15, and C.16. If the user were in the Background Rules Group (Automatic Filtering), then the user had to do only the steps in Figure C.13, C.15, and C.16 then review the traced links.

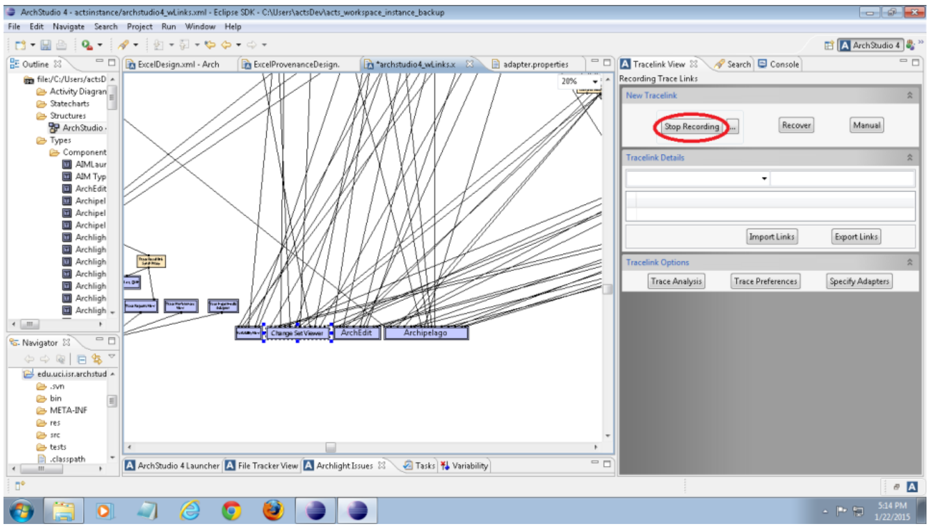


Figure C.13: Screenshot Step 9

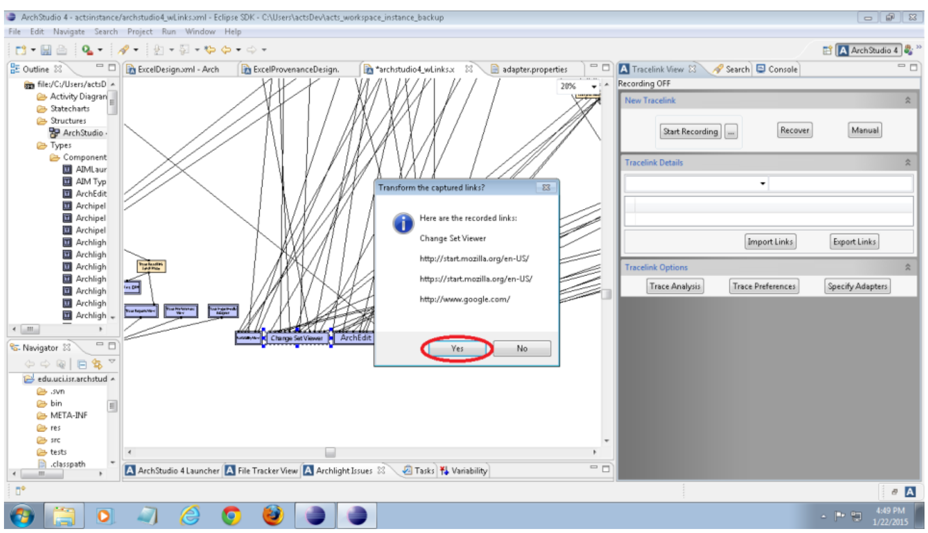


Figure C.14: Screenshot Step 9

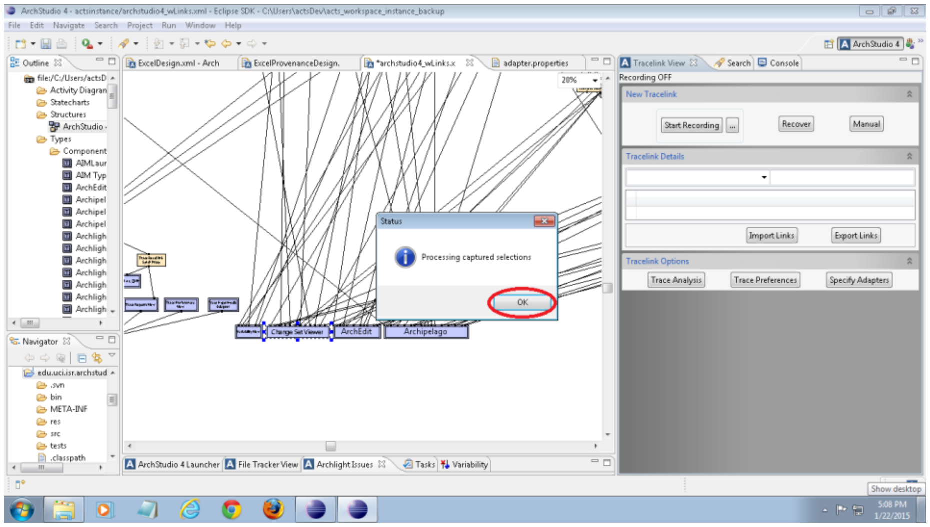


Figure C.15: Screenshot Step 9

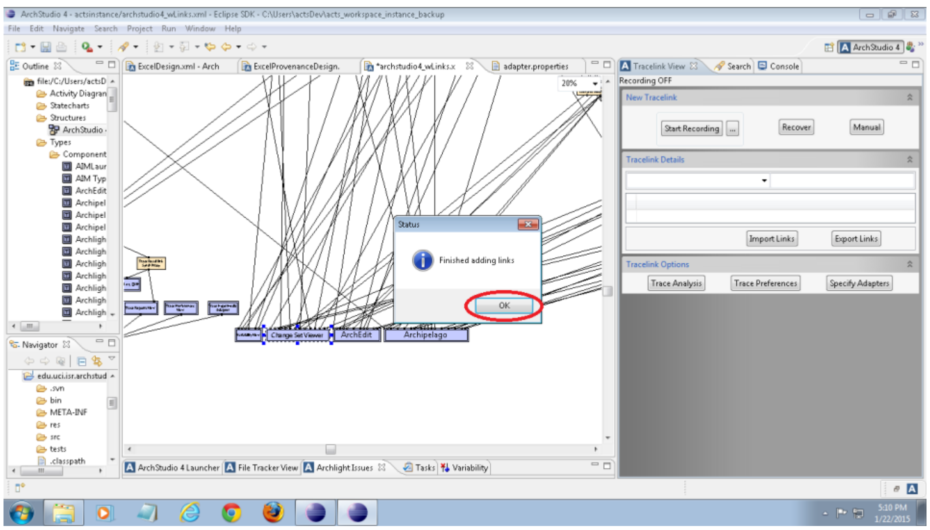


Figure C.16: Screenshot Step 9

10. A menu appeared showing the filters that the user could apply. Those filters removed the noise of the links. The user double clicked on “removeDuplicates_Use1b.xml” as shown in Figure C.17. For example, the user could apply only one filter at a time.

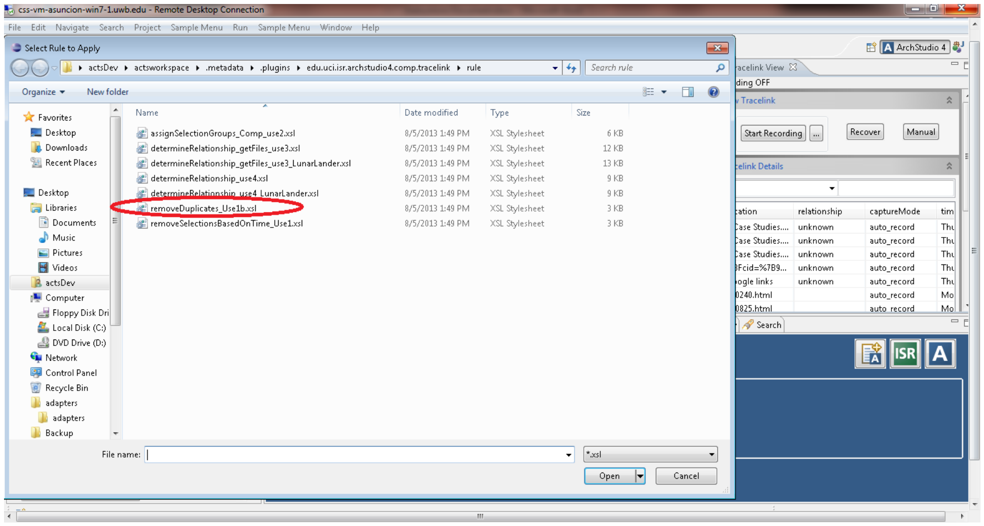


Figure C.17: Screenshot Step 10.

11. As shown in figure C.18, if the user wanted to apply more filters, then they had to click on "Yes".

    On the other hand, if the user didn't want to apply more filters, then they had to click on "No".

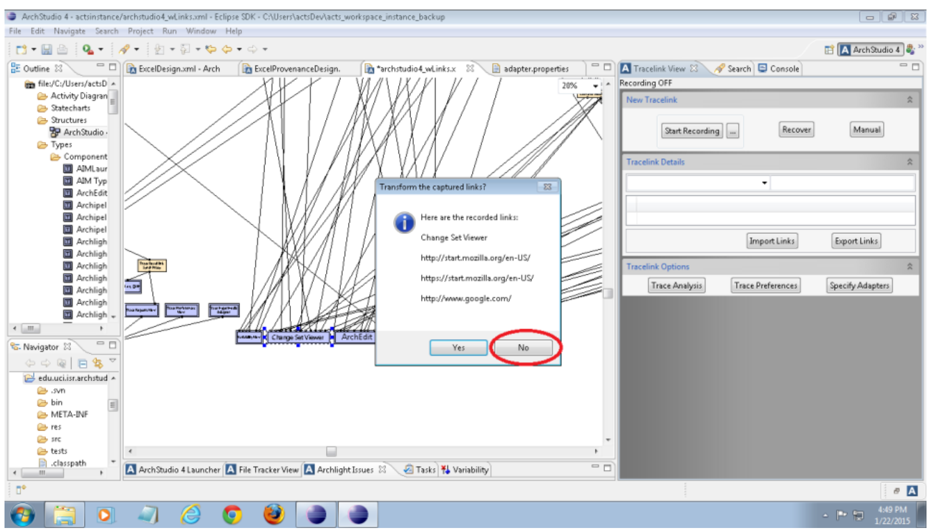


Figure C.18: Screenshot Step 11

12. Finally a message box will appear stating "Finished adding links"; the user would click on ok as shown in Figure C.19, then the user would review the traced links.

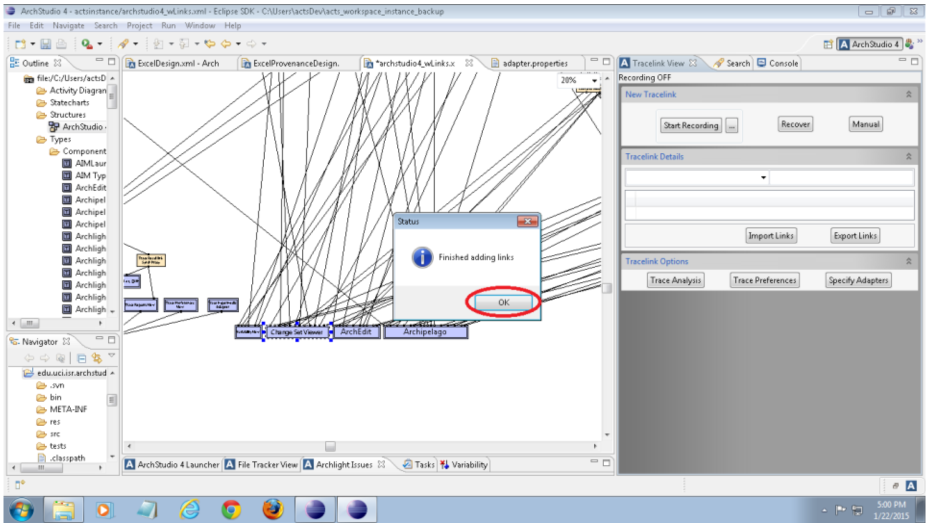


Figure C.19: Screenshot Step 11

D) Satterthwaite Approximation

- Would you recommend this tool to your peers?

Category	Non-traceability	Traceability
Strongly Recommend	4	1
Recommend	14	2
Indifferent	11	5
Do Not Recommend	1	0

Variable	Non-Traceability	Traceability
Stdev	0.749713	0.755929
N	30	8
average	2.7	2.5

Variable	Value
SE	0.300274
Tstat	0.666059
df	29
tcrit	1.699127

- Is ACTS more effective than your current technique of tracing?

Category	Non-traceability	Traceability
Yes	10	2
Maybe	15	5
No	5	1

Variable	Non-Traceability	Traceability
stdev	0.698932	0.64087
n	30	8
average	1.166667	1.125

Variable	Value
SE	0.300274
Tstat	0.666059
df	29
tcrit	1.699127

- Do you think the tool would facilitate software development by making relevant artifacts accessible to all members of the development team?

Category	**Non-traceability**	**Traceability**
Yes	22	7
Maybe	6	1
No	2	0

Variable	**Non-Traceability**	**Traceability**
stdev	0.606478	0.353553
n	30	8
average	1.166667	1.875

Variable	**Value**
SE	0.16699
Tstat	-1.24758
df	29
tcrit	1.699127

- On a scale 1 to 5, rate the usefulness of the links created with ACTS tool.

Category	**Non-traceability**	**Traceability**
Most Useful	7	1
Useful	9	4
Somewhat Useful	10	3
Not useful	4	0
Not useful at all	0	0

Variable	**Non-Traceability**	**Traceability**
stdev	0.999425	0.707107
n	30	8
average	2.366667	2.25

Variable	**Value**
SE	0.309508
Tstat	0.376943
df	29
tcrit	1.699127

- On a scale of 1 to 5, rate the acceptability of time spent to do the recording traceability links.

Category	Non-traceability	Traceability
Most Useful	11	2
Useful	8	4
Somewhat useful	6	2
Not useful	5	0
Not useful at all	11	0

Variable	Non-Traceability	Traceability
stdev	1.116748	0.755929
n	30	8
average	2.166667	2

Variable	Value
SE	0.336154
Tstat	0.495805
df	29
tcrit	1.699127

- On a scale of 1 to 5, rate the acceptability of time spent on finding relevant traceability links.

Category	Non-traceability	Traceability
Most acceptable	9	2
Acceptable	5	4
Somewhat acceptable	11	2
Not acceptable	5	0
Not acceptable at all	0	0

Variable	Non-Traceability	Traceability
stdev	1.101723	0.755929
n	30	8
average	2.4	2

Variable	Value
SE	0.334497
Tstat	1.195825
df	29
tcrit	1.701131

- Rate the acceptability for referring time to back link retrieving and reviewing.

Category	Traceability	Non-traceability
Most acceptable	15	4
acceptable	6	2
Somewhat acceptable	8	1
Not acceptable	1	1
Not acceptable at all	0	0

Variable	Non-Traceability	Traceability
stdev	0.949894	1.125992
n	30	8
average	1.833333	1.875

Variable	Value
SE	0.434234
Tstat	-0.09595
df	29
tcrit	1.699127

- How accurate was the traceability links that were captured? Use your best estimate.

Category	Non-traceability	Traceability
91%-100% accurate	14	3
81-90% accurate	9	1
71%-80% accurate	2	2
60%-70% accurate	3	2
50- 60% accurate	0	0
Less than 50%	2	0

Variable	Non-Traceability	Traceability
stdev	1.436791	1.30247
n	30	8
average	2.066667	2.375

Variable	Value
SE	0.529968
Tstat	-0.5818
df	29

Variable	Value
SE	0.529968
Tstat	-0.5818
tcrit	1.701131

- If you would like to prioritize the friendliness, timeliness and helpfulness of this tool, which of the flowing sequence best describes your experience (1 was most remarkable, 3 was least remarkable).

Category	Non-traceability	traceability
1)Tool is user-friendly. 2)Tool minimizes the time spent to do traceability tasks. 3)Tool helps with developing the tasks.	4	1
1)Tool is user-friendly. 2)Tool helps with developing the tasks. 3)Tool minimizes the time spent to do traceability tasks.	1	0
1) Tool minimizes the time spent to do traceability tasks. 2)Tool is user-friendly. 3)Tool helps with developing the tasks	8	1
1)Tool minimizes the time spent to do traceability tasks. 2)Tool helps with developing the tasks. 3) Tool is user-friendly	9	2
1)Tool helps with developing the tasks. 2)Tool is user-friendly. 3)Tool minimizes the time spent to do traceability tasks.	1	1
1)Tool helps with developing the tasks. 2)Tool minimizes the time spent to do traceability tasks. 3)Tool is user-friendly.	7	3

Variable	Non-Traceability	Traceability
stdev	1.612095	1.767767
n	30	8
average	3.766667	4.375

Variable	Value
SE	0.690835
Tstat	-0.88058
df	29
tcrit	1.699127

BIBLIOGRAPHY

1. Guide for Software Engineering Majors (World Wide Learn) http://www.worldwidelearn.com/online-education-guide/engineering/software-engineering-major.htm

2. Asuncion, H. U. (2009). *Architecture-centric traceability for stakeholders (ACTS)*. University of California, Irvine.

3. Jennifer, K. (2010). Characteristics of an effective management information system [HUA1]. Chron. http://smallbusiness.chron.com/characteristics-good-management-information-system-59060.html

4. Vitez, O. (2015). The definition of Small Business Management. Chron., http://smallbusiness.chron.com/definition-small-business-management-3994.html

5. De Lucia, A., Marcus, A., Oliveto, R., & Poshyvanyk, D. (2012). Information retrieval methods for automated traceability recovery. In *Software and systems traceability* (pp. 71-98). Springer London.

6. Zhou, X., Huo, Z., Huang, Y., & Xu, J. (2008, July). Facilitating software traceability understanding with ENVISION. In *Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International* (pp. 295-302). IEEE.

7. Asuncion, H. U., & Taylor, R. N. (2012). A Holistic Approach to Software Traceability. In *SEKE* (pp. 412-417).

8. Asuncion, H. U., François, F., & Taylor, R. N. (2007, September). An end-to-end industrial software traceability tool. In *Proceedings of the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering* (pp. 115-124). ACM.

9. Cleland-Huang, J., Gotel, O. C., Huffman Hayes, J., Mäder, P., & Zisman, A. (2014, May). Software traceability: trends and future directions. In *Proceedings of the on Future of Software Engineering* (pp. 55-69). ACM.

10. Hayes, J. H., & Dekhtyar, A. (2005, November). Humans in the traceability loop: can't live with'em, can't live without'em. In *Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering* (pp. 20-23). ACM.

11. Asuncion, H., & Taylor, R. N. (2007). Establishing the Connection Between Software Traceability and Data Provenance. ISR Technical Report# UCI-ISR-07-09, S. 5.

12. Moul, D., Brown, M., & Watson, R. (2015, January). Which IBM Rational requirements management tool is best for your teams? IBM Corporation Software Group, Web publication https://jazz.net/library/content/articles/doors/4.0.1/which-rm-tool/which-rm-tool.pdf

13. Antoniol, G., Canfora, G., Casazza, G., De Lucia, A., & Merlo, E. (2002). Recovering traceability links between code and documentation. Software Engineering, IEEE Transactions on, 28(10), 970-983.

14. Spanoudakis, G., & Zisman, A. (2005). Software traceability: a roadmap.*Handbook of Software Engineering and Knowledge Engineering*, *3*, 395-428.

15. Gotel, O., Cleland-Huang, J., Hayes, J. H., Zisman, A., Egyed, A., Grünbacher, P.,& Mäder, P. (2012). Traceability fundamentals. In *Software and Systems Traceability* (pp. 3-22). Springer London.

16. Quantitative and Qualitative Research. Explorable.com. Web publication. (2014, 23 Nov.). https://explorable.com/quantitative-and-qualitative-research

17. Jet Brains, .NET Profiling Experience Like No other. (2015). https://www.jetbrains.com/profiler/

18. Visual Studio, Team Foundation Server. (2013). https://msdn.microsoft.com/en-us/vstudio/ff637362.aspx

19. Archstudio 4. Software and Systems Architecture Development Environment. (2006, 26 November). http://isr.uci.edu/projects/archstudio-4/www/archstudio/

20. Heindl, M., & Biffl, S. (2005, September). A case study on value-based requirements tracing. In *Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering* (pp. 60-69). ACM.

21. Bianchi, A., Fasolino, A. R., & Visaggio, G. (2000). An exploratory case study of the maintenance effectiveness of traceability models. In *Program Comprehension, 2000. Proceedings. IWPC 2000. 8th International Workshop on* (pp. 149-158). IEEE.

22. Neumuller, C., & Grunbacher, P. (2006, September). Automating software traceability in very small companies: A case study and lessons learned. In *Automated Software Engineering, 2006. ASE'06. 21st IEEE/ACM International Conference on* (pp. 145-156). IEEE.

23. Antoniol, G., Canfora, G., Casazza, G., & De Lucia, A. (2000, February). Identifying the starting impact set of a maintenance request: A case study. In Software Maintenance and Reengineering, 2000. Proceedings of the Fourth European (pp. 227-230). IEEE.

24. Ramesh, B., Powers, T., Stubbs, C., & Edwards, M. (1995, March). Implementing requirements traceability: a case study. In Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on (pp. 89-95). IEEE.

25. Gotel, O., & Finkelstein, A. (1997, January). Extended requirements traceability: results of an industrial case study. In *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on* (pp. 169-178). IEEE.

26. Social Research Methods. William Trochim. Web Center for Social Research Methods. Web publication. (2006, 20 Oct.). http://www.socialresearchmethods.net/

27. Asuncion, H. U., & Taylor, R. N. (2012). Automated techniques for capturing custom traceability links across heterogeneous artifacts. In *Software and Systems Traceability* (pp. 129-146). Springer London.

28. Asuncion, H. U., & Taylor, R. N. (2009, May). Capturing custom link semantics among heterogeneous artifacts and tools. In *Proceedings of the 2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering* (pp. 1-5). IEEE Computer Society.

29. Asuncion, H. U. (2008). Towards practical software traceability. *ICSE Companion*, *8*, 1023-1026.

30. Egyed, A., & Grunbacher, P. (2002). Automating requirements traceability: Beyond the record & replay paradigm. In Automated Software Engineering, 2002. Proceedings. ASE 2002. 17th IEEE International Conference on (pp. 163-171). IEEE.

31. Link work items to support traceability. Microsoft Developer Link. Microsoft, Inc. Web publication. (2014, 23 Nov.). http://msdn.microsoft.com/en-us/library/

32. Yin, R. K. (2013). *Case study research: Design and methods*. Sage publications.

33. Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. *Biometrics bulletin*, 110-114.

34. Welch, B. L. (1947). The generalization ofstudent's' problem when several different population variances are involved. *Biometrika*, 28-35.