Integrated Traceability Approach for an Effective Impact Analysis

Lubna Gul¹, Salma Imtiaz², Salman Munir³, Majid Munir⁴, Muhammad Ajmal Khan⁵

ABSTRACT

Change is inevitable, software undergoes continuous change during its life cycle. A small change can trigger high evolution because of the ripple effect identified during the activity of impact analysis. However, it depends on the traceability information, which is the connection between software development artifacts. The current traceability techniques lack the breadth and depth to carryout informative impact analysis. We have performed a detailed literature survey of traceability techniques from the year 2008-2018. These techniques are evaluated on the criteria for effective impact analysis present in the literature. The results highlight that no single technique fulfills the criteria for effective impact analysis alone, they can be combined together to achieve promising results. We have presented a hybrid approach that combines four traceability techniques to achieve the entire criteria for an effective impact analysis after careful evaluation. The techniques combined are: Information Retrieval, Pre-Requirement Specification Traceability, Value based Requirements Traceability Technique and Goal Centric Traceability Technique. Our proposed hybrid approach is empirically validated via a field experiment. Results are analyzed for time and effort utilized in maintaining and retrieving the traceability information. The results are promising as the hybrid approach achieves effective impact analysis within minimal time and effort. We plan to extend the validation to real world impact analysis situation via case study.

Keywords: Traceability Techniques, Hybrid Approach, Impact Analysis, Experiment. Evaluation

1. INTRODUCTION

The success of software depends on meeting the needs of the stakeholders. These needs manifest in form of software requirements. Requirement Engineering (RE) process identifies and manages all the stakeholders' requirements. A proper RE process helps in removing the risks of incomplete requirements, lack of stakeholder's involvement, requirement changes, resource wastage, wrong estimates *etc*. [1]. It is composed of five major activities i.e. elicitation, analysis, specification, validation and management of requirements [2]. The focus of this research is the Requirements Change Management (RCM) since it helps in evolution of requirements during software development process. Software requirements undergo continuous change during the course of software development and even

¹ Department of Computer Software Engineering, University of Engineering and Technology, Mardan, Pakistan.
 Email: lubna.gul@uetmardan.edu.pk (Corresponding Author).
² Department of Computer Science and Software Engineering, International Islamic University, Islamabad, Pakistan
 Email: salma.imtiaz@iiu.edu.pk
³ Faculty of Electrical Engineering and Information Technology, Technical University of Chemnitz, Germany.
 Email: Salman.munir@ibeo-as.com
⁴ International Master Program in Computer Engineering, Mid Sweden University Sundsvall, Sweden.
 Email: majidmuneer@hotmail.com
⁵ Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan.
 Email: Khan046@gmail.com

during maintenance phase. The software suffers from continuous change due to unclear user requirements, changing technology and laws *etc.*, but incorporating changes without observing its effects on other parts of system and artifacts can lead to poor estimates, rework, delay and even project failure [3, 4]. Change may seem to be small but its effect may impact critical areas of the system. Therefore, to perform better resource estimates and maintenance leading to quality product, a mechanism has been introduced to understand modifications and its influence called change impact analysis [5]. It is present in maintenance procedures of software development and is a core activity of requirements change management [6]. It is used to determine the ripple effect of continuously changing requirements during development and maintenance phase. It also determines and estimates the affected units in software after the proposed change is made. It also helps to identify the dependency among the critical functionality, initiate regression testing and estimate the cost of change.

An effective impact analysis requires careful evaluation of the impact of the proposed change to determine the complexities and consequences. This can only be achieved when traceability information is maintained during software development [6]. Traceability is defined as "the ability to describe and follow the life of a requirement in both forward and backward direction (*i.e.*, from its origins, through its development and specification to its, subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases)".

The necessary information of traceability information required for an effective impact analysis is discussed in detail.

Traceability information is categorized into pre and post traceability, vertical and horizontal traceability and traceability of functional and non-functional requirements [7, 8]. A traceability approach that provides all these kinds of traceability can be used for effective impact analysis. Some other important dimensions to traceability are syntactic or semantic traceability and ability to trace manually automatically or semi-automatically [9]. Semantic traceability is attained by giving meaningful labels to links while

Structural traceability is achieved by arranging links into a hierarchical structure [10, 11]. Manual, automatic or semi-automatic traceability represents different levels of tools support.

There are many requirement traceability techniques and tools present in literature [12-17] that support traceability. The techniques and tools identified from literature provide different type and level of traceability. The techniques are; Information Retrieval (IR) [18, 19], Keywords and Ontology [20], Event Based Traceability [21], Goal Centric Requirements Traceability [22], Scenario based Traceability [22, 23], Value Based Requirements Traceability [12], Feature Oriented Requirements Tracing [15], Rule Based Approach [17], Pre-Requirement Specification (Pre-RS) Tracing [16], Knowledge-based Techniques [24], Design Patterns [22,25], Hyper-text Based Approach [26], Aspect Weaving [22,27], Feature-Model Based Approach [28], QuaTrace [29], Process Centered Environments [22,30], Traceability Matrices [22] and TraceM [31].

The practical benefits that are gained from the individual use of these techniques are restricted. Industry performs very limited form of traceability which is usually supported by requirement management tools. An evaluation of these techniques is performed to determine their usefulness for effective impact analysis [6,8] till 2008. We have extended the evaluation to include literature from 2008-2018 and have proposed a hybrid approach to traceability based on the evaluation.

2. RELATED WORK

A detailed literature survey of hybrid or integrated traceability techniques is performed, used for impact analysis. The review highlights that the selection of the traceability techniques is not based on any specific criteria. None of the hybrid approaches provide complete traceability and are not evaluated for the time and effort required in maintaining traceability. The focus of this research is on an integrated approach; therefore, we have presented techniques which are a combination of two or more traceability techniques given in Table 1.

Mehran University Research Journal of Engineering and Technology, Vol. 40, No. 2, Apil 2021 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

347

Table 1: Motivation for Hybrid Approach			
Hybrid Approaches	Positive Aspects	Limitations	References
PROMESIR	Provide support for functional requirements.	Techniques have been selected without evaluation. No support for Non Functional Requirement (NFR)	[32]
TraCS	Provides support for functional requirements and rationale to use heterogeneous traceability techniques.	It does not support Pre RS traceability. The techniques are selected without any evaluation and TraCS is not validated empirically.	[33]
Trustrace	Trace functional, non-functional requirements.	It does not perform pre requirements traceability. Traceability techniques have been chosen without evaluation.	[34]
Traceability Link Graph(TLG)	This approach was only applied to enhance post requirements traceability of functional requirements.	It can be applied to source code only not documentation. It does not support pre-requirements traceability and non-functional requirements.	[35]
VSM+RE+KP+CL	It removed some of the limitations of VSM rather than applying VSM alone. Provide support for functional requirements.	It does not provide any support for NFR and Pre requirements traceability.	[36]
Three Level Framework for Requirements Traceability (TLFRT)	Give encouragement for functional and non-functional requirements. Perform pre and post requirements traceability.	RTM is used for maintaining links which has scalability problem and evaluation of techniques are not performed properly for effective impact analysis.	[8]

PROMESIR, a hybrid approach is a combination of Latent Semantic Indexing (LSI) and Scenario-Based Probabilistic Ranking (SPR) to locate features in basic code via static and dynamic analyses respectively [32]. The results are quite acceptable and prove that PROMESIR performs better than these techniques implemented individually. However, the approach only supports functional requirements and can only be applied to source code. The rationale as to why latent semantic indexing and scenario based probabilistic ranking is used is not given. PROMESIR gives fast and accurate results with help of a computation intensive technology. The approach (PROMESIR) has no tool support.

TraCS [33] is another integrated approach where different traceability techniques are applied to improve Return on Investment (ROI) for saving and maintaining traceability within minimal cost. It offers a rationale to use heterogeneous traceability techniques; however, it does not support Pre-RS traceability. The techniques are selected without any evaluation and TraCS is not validated empirically.

Trustrace [34] is a trust developing process in which positive points of different traceability techniques have been integrated to develop experts trust over recovered links. IR-based [18, 19], goal-centric traceability [22] and rule-based traceability [17] techniques have been combined to introduce a hybrid technique that can trace functional, non-functional requirements and gives high precision and recall values. It permits an expert to declare conventional rules for recovering traceability links but it does not perform pre-RS traceability. The rationale of choosing the traceability techniques is also not provided. Traceability Link Graph (TLG) is textual analysis i.e. IR has been integrated with structural analysis i.e. JRipples to retrieve relevant links among requirement documents and improve their precision and recall values [35]. IR is a technique used to determine the similarities between texts while JRipples is a tool to determine incremental changes in source code only. This approach is only applied to enhance post requirements traceability of functional requirements. It does not support pre-RS traceability and non-functional requirements.

Three well known approaches namely Regular Expression (RE), Key Phrases (KP) and Clustering (CL) to Vector Space Model (VSM) and IR are integrated to achieve traceability [36]. These approaches when used in a combination have shown positive results and removed some of the limitations of VSM. It also does not provide any support for NFR and pre- RS traceability. A framework (TLFRT) [8] using Pre-RS, Value Based Requirement Traceability (VBRT) and Goal Centric Traceability (GCT) has been proposed for performing pre and post requirements traceability of functional and non-functional requirements. Requirement Traceability Matrix (RTM) is used for maintaining traceability links, though it suffers from scalability problem. Techniques were selected on the basis of empirical evidence found in literature. Table 1 presents the pros and cons of the

Mehran University Research Journal of Engineering and Technology, Vol. 40, No. 2, Apil 2021 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

348

hybrid approaches found in literature. Table 1 is motivation enough for a new hybrid approach.

3. EVALUATIONOF TRACEABILITY TECHNIQUES

The traceability techniques have been evaluated individually according to the criteria of impact analysis from multiple perspectives [6]. Each technique has its pros and cons presented in Table 2. The evaluation helped us in selecting techniques for making a hybrid approach. Some of the traceability techniques have already been evaluated [6] according to criteria for effective impact analysis. We have applied the same criteria to the set of traceability techniques surveyed from 2008-2018. The evaluation highlights the gaps, the attributes which are required for effective impact analysis but missing in the surveyed techniques. We have merged different aspects of these traceability techniques to come up with a hybrid solution that provides complete impact analysis. The approach is also empirically validated for time and effort required for saving traceability information. The proposed hybrid approach is found to take less time and effort than each individual technique. It is composed of Pre-RS, IR, VBRT and GCT. The hybrid approach is selected after proper evaluation of all the techniques available in literature for an effective impact analysis. Evaluation criteria has been selected carefully which full fills the criteria for multiple perspective impact analysis [6]. The criterion is chosen based on its significance and importance for impact analysis activity during software development. The importance of each parameter of evaluation criteria is explained in detail in the original study [6]. Detailed evaluation of all the traceability techniques present in literature from 1997-2018 is presented in Table 2. The symbols are to be interpreted as, X indicates no support, √ indicates full support and O indicates partial support whereas - means it is not specified in literature.

All of the techniques have some positive points as well as negative points. Most of the techniques like Rule Based (RB) Approach, Hyper-Text Based (HB) Approach, Feature-Model Based (FB) Approach, Process Centered Environments (PCE), Design Patterns (DP), Traceability Matrices (SM), Keywords and Ontology (KWO), Aspect Weaving (AW), Event Based Traceability (EBT), Scenario-Based (SB) Approach and Pre-RS are not validated empirically.

Technique→ ↓Criteria	VBRT [12]	FORT [15]	Pre-RS [16]	RB [17]	HB [26]	FB [28]	PCE [22,30]	DP [22,25]	Matrices (SL) [22]	KWO [20]	AW [22,27]	GC [22]	EBT [21]	IR [18-19]	SBT [22-23]	Qua Trace [29]	KBT [24]	Trace M [31]
Table 2: Evaluation of Requirements Traceability Techniques for Impact Analysis																		
Functional Requirement	√	√	√	√	√	v	√	X	√	X	√	X	√	√	√	√	√	√
NFR Support	X	X	√	√	√	√	√	√	O	√	O	√	√	O	X	X	-	-
Pre Traceability	X	X	√	X	X	X	X	X	X	X	X	X	X	X	X	X	X	O
Post Traceability	√	√	X	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Evolution Support	√	√	√	√	O	-	-	√	O	√	-	√	√	X	√	√	O	√
Visualization Support	√	√	√	√	X	√	-	√	X	√	-	√	√	√	√	√	O	√
Explicit Links	√	√	√	-	√	-	-	√	√	-	-	√	√	√	√	√	√	√
Implicit Links	√	√	√	-	√	-	-	√	X	-	-	√	√	√	√	√	√	√
Structural Traceability	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Semantic Traceability	√	√	√	√	√	√	√	√	X	√	√	√	√	√	√	√	X	√
Tool support	√	X	X	X	√	X	X	X	X	X	X	√	√	√	√	√	√	√
Validation	√	√	X	X	X	X	X	X	X	X	X	√	X	√	X	√	√	√

Pre-RS technique supports pre requirements traceability while all others techniques support post requirements traceability. GCT technique is the only empirically validated technique which covers non-functional requirements. Some techniques like Feature Oriented Requirements Tracing (FORT), RB, FB, PCE, DP, SL, KWO and AW are not supported by tools while some of them are not validated empirically and require implementation. IR technique is used to reduce the scalability issues while Value Based Requirements Traceability and Feature Oriented Requirements Tracing are used to prioritize links for reducing cost and time.

The hybrid approach proposed in this research combines Pre-RS for pre requirements traceability, IR and VBRT for valuable functional requirements and GCT for non-functional requirements. Therefore, full traceability is achieved by combining different techniques in a manner that only valuable traceability links are saved. Identifying the valuable links helps in reducing the time and cost required to save and maintain the traceability information.

4. HYBRID APPROACH

Requirements traceability is one of the most important features for attaining complete and effective impact analysis. Complete requirements traceability is one of the emerging trends of requirements engineering process. The software industry tries to avoid implementing traceability due to many factors e.g. limited time, high cost, extra effort required to save traceability information, lack of knowledge, lack of tool support, lack of appropriate training etc. [7,37-39]. Many traceability techniques have been introduced in literature but none of them provide complete requirement traceability for effective impact analysis. We aim to combine the already existing traceability techniques instead of making new one to achieve full traceability and effective impact analysis. We have integrated the techniques in such a manner to reduce the effort and time required to save and maintain the traceability information. Fig. 1 presents the detailed design of the sequence of the traceability techniques in hybrid approach for stepwise covering end to end traceability.
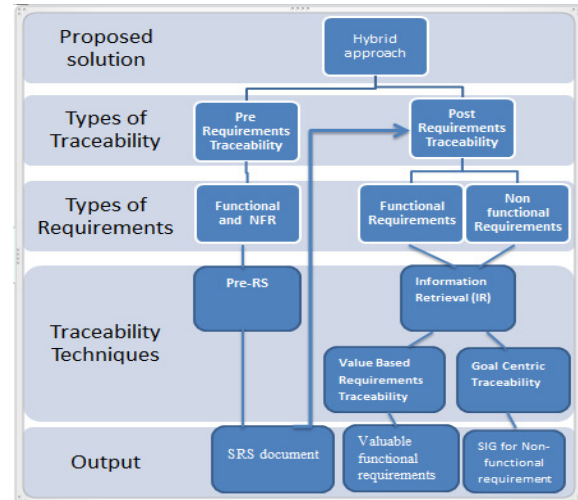

Fig. 1: Proposed Hybrid Approach for Effective Impact Analysis

Hybrid approach is a semi-automatic approach in which we integrate traceability techniques for fulfilling multiple perspective impact analysis criteria [6]. It is the combination of different traceability techniques to cover the impact of change on different requirements with in less time, cost and effort. Each technique has its pros and cons so we combine all the positive aspects of each technique to fulfill the whole criteria of impact analysis. The hybrid approach is divided into two segments. The functional and nonfunctional requirements between source and SRS document are traced by pre-RS tracing in the first segment whereas in second segment functional requirements are traced by using information retrieval technique from SRS. Traceability information of nonfunctional requirements is extracted with help of Goal Centric traceability. The Valuable post traceability links of functional requirements and non-functional requirements are extracted using VBRT and GCT respectively. Finally, we are able to get the complete set of traceability links of functional and nonfunctional requirements.

5. TOOL SUPPORT

Pre-RS is the only technique which supports pre requirements tracing back to their source. The functional and non-functional requirements have been determined manually and stored in an XML file. Documentation is also saved in a simple Microsoft

Mehran University Research Journal of Engineering and Technology, Vol. 40, No. 2, Apil 2021 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

350

Word file as SRS (Software Requirement Specification) document. The Pre-RS is not supported by any tool.

The documentation of pre-requirements traceability is followed by IR and VBRT for performing post requirements traceability. IR automatically generates traceability links among different artifacts and reduces effort [8]. Automated generation of traceability links via tool support helps in reducing time as compared to manual tracing [40]. VBRT is used to prioritize traceability links to reduce cost, time and effort by tracing valuable links only [12]. For this purpose, a requirements management tool called ReqSimile is used which supports traceability. ReqSimile is a general purpose tool that supports impact analysis and change management during software evolution and also provides support for identification of traceability links. The stakeholders manually give priorities to different requirements and store valuable requirements for reducing time and effort [12]. It directly extracts requirements from SRS document in Microsoft Word format and stores links in database file using Microsoft Access.

Finally, GCT is used for modeling of non-functional requirements identified in SRS document. The experimental results show that GCT supports and manages the effects of change on NFR (Non-functional Requirements). GCT is supported by many software tools. StarUML is used for sketching different non-functional requirements. This tool is used after extracting nonfunctional requirements by using IR. Users had to convert NFRs in the form of goals and further divide it into sub goals by using StarUML.

6. EXPERIMENT

Science and engineering contains different tools, approaches and techniques for process validation. Survey, simulations, field experiments, controlled experiments and case studies are the most commonly used techniques for validating any process [41]. Field experiment and controlled experiment are commonly used to evaluate processes or techniques. We have chosen field experiment to validate this research by checking the claims of hybrid traceability approach. We are also interested to identify the impact of the

proposed hybrid approach in real life setting in terms of time and effort. The time and effort taken by the hybrid approach will be compared to the most common way of traceability practiced by the industry, which is requirement traceability matrices [8], therefore, we have compared the hybrid approach with a RTM approach to evaluate its effectiveness.

The experimental process performed to validate hybrid approach is described in Fig. 2. A brief summary is presented in form of experimental tasks which includes sequence of tasks in shown in Table 3.
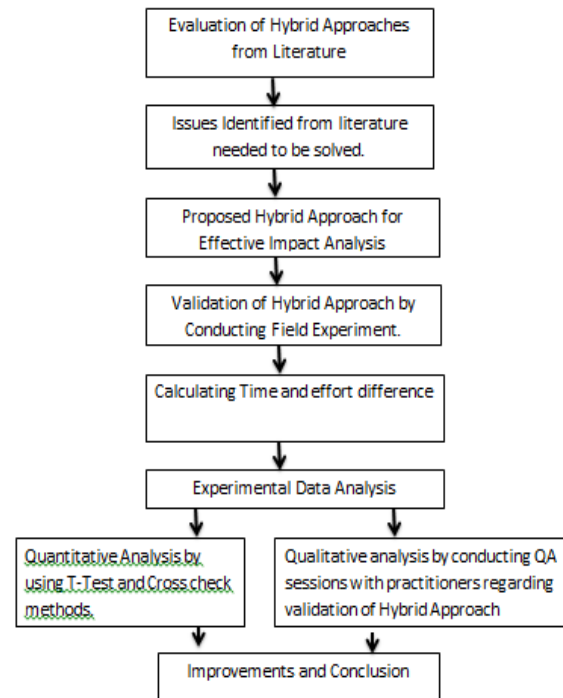

Fig. 2. Overview of Experimental Design

We chose RTM for comparison with our proposed approach because it is commonly used in industry. It is very cheap in terms of cost and does not require much training, can easily be implemented with help of Microsoft excel sheets. The alternate H (Hypothesis) as well as the NH (Null Hypothesis) is given below.

H1:   The proposed HA decreases the time used for saving and maintaining traceability than RTM.

NH1:   The proposed HA has no effect on the time used for saving and maintaining traceability

Mehran University Research Journal of Engineering and Technology, Vol. 40, No. 2, Apil 2021 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

351

than RTM while providing complete traceability.

H2: The proposed HA decreases the effort used for saving and maintaining traceability than RTM while providing complete traceability.

NH2: The proposed HA has no effect on the effort used for saving and maintaining traceability than RTM while providing complete traceability.

Table 3: Experimental Tasks		
Tasks	Activity Performed during Task	Description of Activities
1	Conducting Pre-requirements traceability from source.	Each group was collecting pre-requirements back to their source manually.
2	Saving SRS document.	Each group made SRS document by adding all user needs into proper arrangement by both groups.
3	Conducting Post-Requirements traceability of functional requirements.	Both groups extracted Functional requirements from SRS document and traceability links were obtained among the different functional requirements.
4	Prioritized functional requirements.	Each group prioritized traces by involving different stakeholders manually.
5	Conducting post requirements traceability of NFRs.	Each group traced Non-functional requirements from SRS document.
6	Save links of non-functional requirements.	Non-functional requirements were diagrammatically represented by both groups.
7	Save calculated time.	Total time for completing all tasks is measured.
8	Monitor and control change	If modifications in stakeholder prioritization happens then sub task "prioritize functional requirements" will be perform but if Any modifications happens in Pre-requirements then whole set of tasks will be perform again.

The first group consisted of four members who were selected for achieving experimental goals. The roles (requirement engineer, system developer, project manager and customer) were assigned to the four members. Initially the group was assigned the tasks of achieving impact analysis criteria using the proposed hybrid approach. The tasks were performed and experimental data collected. Afterwards same group performed the experimental tasks for achieving impact analysis criteria without the hybrid approach. RTM was used for saving traceability links of functional and non-functional requirements. Two projects of same nature and size were chosen and given to the group one by one. Both projects were medium in size i.e. M1 (medium sized project =>100<300 functional points size according to International Software Benchmarking Standards Group (ISBSG)) [42]. This was done to make sure that the results do not vary because of the size and nature of the project. Data collected from each group during experiment execution was checked qualitatively and quantitatively. To evaluate the impact of hybrid approach we performed an experiment which showed its effect in real life situations i.e. calculating time and effort difference between group with and without using the proposed hybrid approach. The experimental tasks performed are presented in sequence in Table 3.

6.1 Variables

6.1.1 Independent Variable: The hybrid approach is the independent variable.

6.1.2 Dependent Variables: The time and effort used by the hybrid approach to save and maintain traceability information is the dependent variable.

7. QUANTITATIVE AND QUALITATIVE ANALYSIS

The experimental results have been evaluated by a group using the hybrid approach and without using the hybrid approach. The qualitative analysis and data collection was performed by interviews and group discussions of every recorded task. The quantitative analysis of reported results was evaluated on the basis of difference by using T-Test and cross checking.

T-Test: The statistical test is used to analyze the results differences between real conditions and changing results due to testing time fluctuations [43]. Commonly two types of t-test are used dependent and independent mean t-test. We used dependent mean t-test (paired t-test) for our data analysis, as group is performing tasks using hybrid approach; same group

Mehran University Research Journal of Engineering and Technology, Vol. 40, No. 2, Apil 2021 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

352

is performing same tasks without using hybrid approach and they used RTM.

T-test was performed to equate time and effort needed for executing tasks for effective impact analysis. The experimental group performed the tasks assigned using the proposed hybrid approach and then the same group performed experimental tasks via RTM approach.

Means is the simply calculated by taking average of values while standard deviation is used to calculate the dispersal of set of values [44]. Table 4 the acceptance and rejection of hypothesis depends upon the value of p. P-value is the probability value of the result and assuming that null hypothesis is true. α (Alpha) represent the significant level used to calculate the probability of rejecting null hypothesis. If the value of p is greater than alpha, it shows that the null hypothesis is accepted, however if the value of p is less than alpha then the null hypothesis is rejected [45].

Group (with HA) = (M=43.125, SD=41.15045)
Group (without HA) = (M=67.5, SD=52.67827);
p = 0.015044 which is <0.05.

Similarly, effort can also be calculated as group using HA (M=172.5, SD=164.6018) and group without using HA.

Group (with HA) = (M=172.5, SD=164.6018)
Group (without HA) = (M=270, SD=210.7131)
Conditions; p = 0.015044", Here p > alpha i.e. 0.015044<0.05, calculated for effort.

For checking whether both the Null hypotheses are accepted or rejected, we examine the value of p. in both case time and effort the value of p is less than alpha so our alternate hypotheses (H) are accepted and null hypotheses (NH) are rejected.

H1: The proposed HA decreases the time used forsaving and maintaining traceability than RTM while providing complete traceability.
H2: The proposed HA decreases the effort used for saving and maintaining traceability than RTM while providing complete traceability.

Results in Figs. 3-4 show clear distinction between data recorded from group performing tasks before and after using HA. The results are defined as "There was a major difference in the time to complete tasks by group using hybrid approach as compared to group without using Hybrid approach.

Table 4: T-Test Results				
	Time (min)		Effort (person/min)	
	Group Using HA	Group without using HA	Group using HA	Group without using HA
Means	43.125	67.5	172.5	270
Standard deviation	41.15045	52.67827	164.6018	210.7131
P value	0.015044		0.015044	
Significant value α (Alpha)	95 % = 0.05		95 % = 0.05	
Results	Our H1 is acknowledged because there is difference between results and p< alpha.		Our H2 is acknowledged because there is difference between results and p < alpha.	

Cross Checking: Cross checking method was used to quantitatively analyze the retrieved data.

In Figs. 5-6, results were calculated by comparing the difference of time and effort for completing each experimental task using hybrid approach and without using hybrid approach.
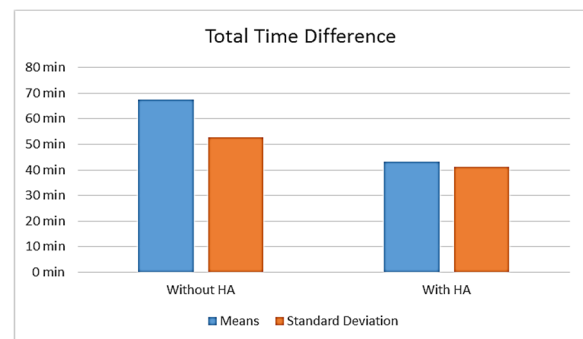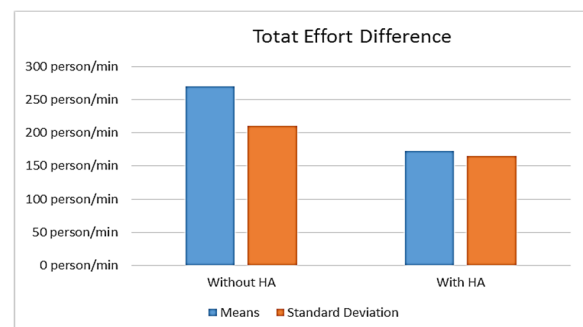

Fig. 3: Means and Standard Deviation for Time

Fig. 4.: Means and Standard Deviation for Effort

Mehran University Research Journal of Engineering and Technology, Vol. 40, No. 2, Apil 2021 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]
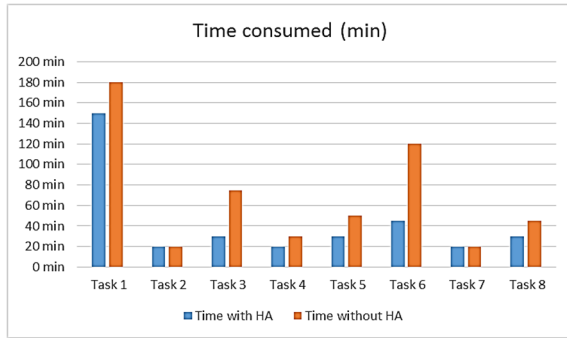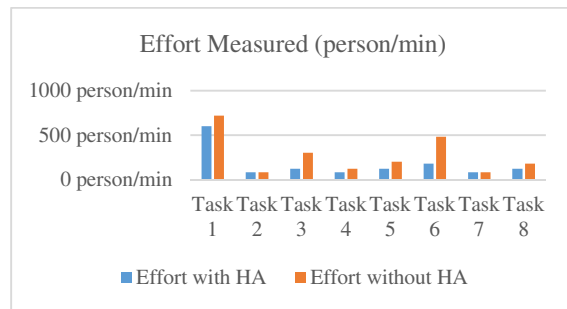
353

Fig. 5. Time Difference

Fig. 6: Effort Difference

The above results show that the Hybrid traceability approach is effective for covering the effective impact analysis criteria. Table 5 supports the hypothesis that Hybrid approach will save effort and time while achieving the whole criteria of impact analysis. Hybrid approach saves 3 hours and 15 minutes for covering the criteria of impact analysis and saves 13 man/hour efforts. It means that if we add four men to group without HA they will perform with same time and with same effort as done by group with HA.

Table 5: Cross Checking Results		
Tasks	Time Difference	Effort Difference
1	30 min	120 person/min
2	0 min	0 person/min
3	45 min	180 person/min
4	10 min	40 person/min
5	20 min	80 person/min
6	75 min	300 person/min
7	0 min	0 person/min
8	15 min	60 person/min
Result	195 min = 3hr 15mins	780 person/min = 13 man/hr.

The qualitative analysis was performed on the data obtained from interviewing the experiment's participants individually. They were interviewed to judge the pros and cons of integrated traceability approach. The interviews were conducted in two

sessions. First session was conducted before starting experiment in which we extracted the comments of stakeholders about traceability techniques and tried to avoid misinterpretation about stakeholder's opinions. Second session was conducted after completing experiment to examine the effectiveness of hybrid approach in actual life setting i.e. time and effort.

Selected group completed the experimental tasks, once with using hybrid approach and the other without using the hybrid approach. Group discussions were allowed among the members for generating reports after every task. Researchers critically reviewed those reports to compare the results both with and without using hybrid approach. The participants found the hybrid approach effective, however recommended usage of a single tool for performing impact analysis. They used different tools for completing all tasks during experiment execution. However, a lot of time is required to manually save and maintain traceability links with RTM; therefore, it is not a good option for impact analysis.

8. CONCLUSION

This approach facilitates the software developing organizations to manage end to end traceability in less time and effort. By combining the good points of Pre-RS, IR, VBRT and GCT techniques, we were able to get better results and effectively fulfill the impact analysis criteria within less time and effort. From literature study we also came to know that IR decrease the scalability problem of RTM, however RTM is inexpensive and feasible for those projects in which scalability is not a constraint. IR technique helped in reduction of time and effort than using other techniques. Time and effort were further optimized with help of VBRT by discarding the un-important traces and separating valuable traces. GCT covers non-functional requirements and represents them in form of soft goals and operationalization. The whole process used in the hybrid approach does not require any special expertise for managing traceability information. Different tools are used for evaluating hybrid approach in real life setting. Hybrid approach was analyzed by conducting a field experiment on a group of 4 individuals.

Mehran University Research Journal of Engineering and Technology, Vol. 40, No. 2, Apil 2021 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

354

The major contributions of the research are:

(i) Detailed literature survey to identify the traceability techniques, approaches and tools present in literature. The survey aimed to identify both individual as well as integrated techniques for traceability.

(ii) Evaluation of the traceability techniques based on the criteria for effective impact analysis [6] was performed.

(iii) Design and development of a hybrid approach which fulfills the criteria of effective impact analysis was performed.

(iv) Validation of the hybrid approach for effective impact analysis in terms of time and effort for saving traceability information with help of a field experiment.

9. FUTURE WORK

This hybrid approach has been analyzed and validated by an experiment. It can be statistically validated in more software companies for evaluating its usefulness in real scenarios. New researchers can implement this approach as a tool, and perform fully automatic traceability, since currently a semi-automatic tool support was used. The tools which were currently used were general purpose. A customized tool support will improve the approach. The approach can also be implemented as simulator, where actual embedded functions of this approach can be represented in the form of simulations by using MATLAB. This will help companies to understand and implement the approach in an effective manner.

ACKNOWLEDGEMENT

REFERENCES

1. Nuseibeh B., Easterbrook S., "Requirements engineering: a roadmap", *Proceedings of the Conference on the Future of Software Engineering*, pp. 35-46, ACM, 2000.

2. Sommerville I., Sawyer P., "*Requirements Engineering: A Good Practice Guide*", John Wiley and Sons, Inc.. 1997.

3. Arnold R.S., "*Software Change Impact Analysis*", IEEE Computer Society Press, 1996.

4. Ali S., Iqbal N., Hafeez Y., "Towards Requirement Change Management for Global Software Development using Case Base Reasoning", *Mehran University Research Journal of Engineering and Technology*, Vol. 37, No. 3, 639-652, July 2018.

5. Lee M.L., "*Change impact analysis of object-oriented software*", Ph.D. Dissertation, George Mason University, Fairfax, Virginia, USA, 1998.

6. Imtiaz S., Ikram N., Imtiaz S., "Impact analysis from multiple perspectives: Evaluation of traceability techniques", *Proceedings of the 3rd IEEE International Conference on Software Engineering Advances*, pp. 457-464, 2008.

7. Gotel O.C., Finkelstein, C.W., "An analysis of the requirements traceability problem", *Proceedings of the First International Conference on Requirements Engineering*, pp. 94-101, 1994.

8. Raja U., Kamran K. "*Framework for Requirements Traceability*", Blekinge Institute of Technology, 2008.

9. Regan G., McCaffery F., McDaid K., Flood D., "Traceability- Why do it?", *Proceedings of the International Conference on Software Process Improvement and Capability Determination*, pp. 161-172, Springer, Berlin, Heidelberg, 2012.

10. Bohner S.A., Gracanin D., "Software impact analysis in a virtual environment", *Proceedings of the 28th Annual NASA Goddard Software Engineering Workshop, Software Engineering Workshop*, pp. 143-151, Greenbelt USA, December 2-4, 2003.

11. Arkley P., Mason P., Riddle S., "Position paper: Enabling traceability", *Proceedings of the 1st International Workshop on Traceability in Emerging Forms of Software Engineering*, Edinburgh, Scotland, pp. 61-65, 2002.

12. Heindl M., Biffl S., "A case study on value-based requirements tracing", *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, 2005.

Mehran University Research Journal of Engineering and Technology, Vol. 40, No. 2, Apil 2021 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

355

13. Blaauboer F., Sikkel K., Aydin M.N., "Deciding to adopt requirements traceability in practice", *Proceedings of the International Conference on Advanced Information Systems Engineering*, pp. 294-308, Springer, Berlin, Heidelberg, 2007.

14. Cleland-Huang J., "Toward improved traceability of non-functional requirements", *Proceedings of the 3rd international Workshop on Traceability in Emerging Forms of Software Engineering*, pp. 14-19, ACM, 2005.

15. Ahn S., Chong K., "A feature-oriented requirements tracing method: A study of cost-benefit analysis", *Proceedings of the International Conference on Hybrid Information Technology*, Vol. 2, pp. 611-616, 2006.

16. Ravichandar R., Arthur J.D., Pérez-Quinones M., "Pre-requirement specification traceability: Bridging the complexity gap through capabilities", *arXiv preprint cs/0703012*, 2007.

17. Spanoudakis G., Zisman A., Pérez-Minana E., Krause P., "Rule-based generation of requirements traceability relations", *Journal of Systems and Software*, Vol. 72, No, 2, pp.105-127, 2004.

18. Hayes, J.H., Dekhtyar, A. and Osborne, J., "Improving requirements tracing via information retrieval", *Proceedings of the 11th IEEE International Requirements Engineering Conference*, pp. 138-147, 2003.

19. Cleland-Huang J., Chang C.K. and Wise, J.C., "Automating performance-related impact analysis through event based traceability", *Requirements Engineering*, Vol. 8, No. 3, pp.171-182, 2003.

20. Cysneiros L.M., do Prado Leite J.C.S., "Nonfunctional requirements: From elicitation to conceptual models", *IEEE Transactions on Software Engineering*, Vol. 30, No. 5, pp.328-350, 2004.

21. Cleland-Huang J., Chang C.K., Christensen M., "Event-based traceability for managing evolutionary change", *IEEE Transactions on Software Engineering*, Vol. 29, No. 9, pp.796-810, 2003.

22. Cleland-Huang J., "Toward improved traceability of non-functional requirements", *Proceedings of the 3rd International Workshop on Traceability in Emerging Forms of Software Engineering*, pp. 14-19, ACM 2005.

23. Naslavsky L., Alspaugh T.A., Richardson D.J., Ziv H., "Using scenarios to support traceability", *Proceedings of the 3rd International Workshop on Traceability in Emerging Forms of Software Engineering*, pp. 25-30, ACM, 2005.

24. Lock S., Kotonya G., "Tool support for requirement level change management and impact analysis", Doctoral Symposium Proceedings, 1998.

25. Cleland-Huang J., Schmelzer D., "Dynamically tracing non-functional requirements through design pattern invariants", *Workshop on Traceability in Emerging Forms of Software Engineering, in conjunction with IEEE International Conference on Automated Software Engineering*, Vol. 10, p. 1, 2003.

26. Maletic J.I., Munson E.V., Marcus A., Nguyen T.N., "Using a hypertext model for traceability link conformance analysis", *Proceedings of the International Workshop on Traceability in Emerging Forms of Software Engineering*, pp. 47-54, 2003.

27. Chitchyan R., Rashid A., Sawyer P., Garcia A., Alarcon M.P., Bakker J., Tekinerdogan B., Clarke S., Jackson A., "Survey of aspect-oriented analysis and design approaches", *Report of the EU Network of Ecxcellance on AOSD*, pp. 1-82, Venice, Italy, May 2005.

28. Pashov I., Riebisch M., "Using feature modeling for program comprehension and software architecture recovery", *Proceedongs of the 11th IEEE International Conference and Workshop on the Computer-Based Systems*, pp. 406-417, 2004.

29. Von Knethen A., Grund M., "QuaTrace: a tool environment for (semi-) automatic impact analysis based on traces", *Proceedings of International Conference on Software Maintenance*, pp. 246-255, 2003.

30. Pohl K., Dömges R., Jarke M., "Towards method-driven trace capture", *Proceedings of the International Conference on Advanced Information Systems Engineering*, Springer, Berlin, Heidelberg, pp. 103-116, 1997.

31. Sherba S.A., Anderson K.M., Faisal M., "A framework for mapping traceability relationships", *Proceedings of the 2nd*

Mehran University Research Journal of Engineering and Technology, Vol. 40, No. 2, Apil 2021 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

356

International Workshop on Traceability in Emerging Forms of Software Engineering, pp. 32-39, 2003.

32. Poshyvanyk D., Gueheneuc Y.G., Marcus A., Antoniol G., Rajlich V., "Feature location using probabilistic ranking of methods based on execution scenarios and information retrieval", *IEEE Transactions on Software Engineering*, Vol. 33, Vol. 6, 2007.

33. Cleland-Huang J., Zemont G., Lukasik W., "A heterogeneous solution for improving the return on investment of requirements traceability", *Proceedings of the 12th IEEE International Requirements Engineering Conference*, pp. 230-239, 2004.

34. Ali N., "Trustrace: Improving automated trace retrieval through resource trust analysis", Proceedings of the 19th IEEE International Conference on Program Comprehension (ICPC), pp. 230-233, 2011.

35. McMillan C., Poshyvanyk D., Revelle M., "Combining textual and structural analysis of software artifacts for traceability link recovery", *Proceedings of Workshop on Traceability in Emerging Forms of Software Engineering*, IEEE Computer Society, pp. 41-48, 2009.

36. Chen X., Grundy J., "Improving automated documentation to code traceability by combining retrieval techniques", *Proceedings of the 26th IEEE/ACM International Conference on Automated Software Engineering*, IEEE Computer Society, pp. 223-232, 2011.

37. Cleland-Huang J., "Just enough requirements traceability", *Proceedings of the 30th Annual International Computer Software and Applications Conference*, Vol. 1, pp. 41-42, 2006.

38. Ramesh B., Jarke M., "Toward reference models for requirements traceability", *IEEE Transactions on Software Engineering*, Vol. 27, No. 1, pp.58-93, 2001.

39. Cleland-Huang J., "Requirements Traceability-When and How does it Deliver more than it Costs?", Requirements Engineering, *Proceedings of the 14th IEEE International Conference*, pp. 330-330, 2006.

40. De Lucia A., Oliveto R., Tortora G., "Assessing IR-based traceability recovery tools through controlled experiments", Empirical Software Engineering, Vol. 14, No. 1, pp.57-92, 2009.

41. Arnold R.S., Bohner S.A., "Impact analysis-towards a framework for comparison", *Proceedings of the Conference on Software Maintenance*, pp. 292-301, 1993.

42. Software Measurement Services Ltd. "Small project','medium-size project'and 'large project':what do these terms mean? [Online]", www.measuresw.com [Accessed: 21 August 2015].

43. Hole G., "Research Skills", February 2009.

44. Harris W. F., "Algebra of sphero-cylinders and refractive errors, and their means, variance, and standard deviation", American Journal of Optometry and Physiological Optics, Vol. 65, No. 10, pp. 794-802, 1988.

45. Hsu H., Lachenbruch P.A., "*Paired t test*", Wiley Encyclopedia of Clinical Trials, 2008.

Mehran University Research Journal of Engineering and Technology, Vol. 40, No. 2, Apil 2021 [p-ISSN: 0254-7821, e-ISSN: 2413-7219]

357