Causes of requirement change – A systematic literature review

Conference Paper · January 2012

DOI: 10.1049/ic.2012.0003

CITATIONS
13

READS
907

5 authors, including:

Muneera Bano
Swinburne University of Technology
39 PUBLICATIONS **296** CITATIONS

SEE PROFILE

Salma Imtiaz
International Islamic University, Islamabad
14 PUBLICATIONS **75** CITATIONS

SEE PROFILE

Naveed Ikram
Riphah International University
55 PUBLICATIONS **321** CITATIONS

SEE PROFILE

Mahmood Niazi
King Fahd University of Petroleum and Minerals
100 PUBLICATIONS **2,771** CITATIONS

SEE PROFILE

Causes of Requirement Change – A Systematic Literature Review

Muneera Bano [1], Salma Imtiaz [1], Naveed Ikram [2], Mahmood Niazi [2, 3], Muhammad Usman [1]

[1] Department of Computer Science and Software Engineering, International Islamic University Islamabad, Pakistan
muneera@iiu.edu.pk , salma.imtiaz@iiu.edu.pk, m.usman@iiu.edu.pk
[2] Faculty of Computing, Riphah International University Islamabad, Pakistan
naveed.ikram@riu.edu.pk
[3] School of Computing and Mathematics, Keele University, Keele UK
[3] Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, KSA
m.k.niazi@cs.keele.ac.uk, mkniazi@kfupm.edu.sa

Abstract* — *Context: Research shows that one of the main reasons of project failure is changing requirements. The success or failure of software projects largely depends upon how we respond to changing requirements. The knowledge about the causes of requirements change can improve our ability to make better decisions and manage changing requirements effectively.
Objective: In this paper we present findings from an empirical study that was aimed at identifying the causes of requirement change and the frequency of these causes in different software development phases.
Methods: We performed a systematic literature review and went through all the stages required by the process. Although our search strings yielded a large amount of papers but after careful filtration we were left with only five papers (six studies) which reported empirical knowledge about the causes of requirement change.
Results: We have identified different causes and their frequency in software development phases. We have classified the extracted causes of requirements change into two major types i.e., essential and accidental causes.
Conclusions: It is surprising to find little empirical evidence on the causes of requirements change as requirements change has been widely quoted as one of the major challenges faced by requirements engineers. With this small number of evidences, it is hard to generalize the research results. There is a need for further empirical research to identify and fully understand the causes of requirement change.

Keywords – requirements engineering; causes; requirements change; systematic literature review.

I. INTRODUCTION

Changing requirements have been considered as a challenging area of research by the software engineering community [1]. It has been observed that requirements change during different phases of software development life cycle (SDLC) [2] and this change plays a vital role in success or failure of any project [3]. The fact is that more than half of the system's requirements will change before the actual deployment of the system [4]. Most of the software failures are attributed to poor requirements engineering in which ambiguous and incomplete requirements lead to changes throughout the SDLC [5]. In recent years the trend has changed from blaming the problem towards identifying the cause of that problem [6]. For managing requirement changes we need to identify the root cause of those changes in order to monitor and manage them effectively and also to find better solutions. There are numerous causes of requirement change highlighted in the literature such as change in technology, user preferences, and ever changing environment [7] [8] [9]. These causes can occur during any phase of SDLC and in turn lead to changes in the system's overall requirements.

Literature highlights the fact that early change anticipation in design, having knowledge of where to look for it and which requirements change more often than the others, would help in cost saving, reducing overall development time and increasing the success rate of projects [10]. Failing to respond to the requirements change results in delays, configuration problems, defects, and overall customer dissatisfaction for not getting the quality product [11]. Agile processes accept the reality of change and manage requirements by "embracing" rather than "controlling" changes. This fact has now been accepted by software developers and they consider responding to requirement change as being more important than focusing on stable requirements for benefits in terms of cost and development time. According to the survey of Version One [12], 55% of respondents consider agile methods to be successful in 90-100% of cases. Freezing the requirements in the early phases of SDLC would not provide the actual benefits required by the stakeholders. Accepting the fact that change is inevitable can accommodate and manage a successful project.

Requirement changes also have different risks associated with them in terms of cost and time and the impact is increased with the propagation of those changes from one phase to the next. Therefore it is necessary to gather and classify the causes, and to map their impact on software development artefacts.

The objective of this study is to collect evidence of different causes of requirement change by conducting a Systematic Literature Review and to categorize requirement changes into different logical groups. With the help of this categorization we hope to be able to manage the causes of requirement change rather than requirement change itself. Once we have the data from empirical evidences, we can also measure how often a change category occurs during different SDLC phases.

Kitchenham adapted Systematic Literature Reviews (SLR) from Evidence Based Medicine (EBM) and introduced this new way of doing research in order to find an evidence

about a specific research question [13]. Evidence Based Software Engineering (EBSE) aims to reduce the gap between research and practice by promoting a rigorous process of finding evidences that are empirical and stand against quality criteria. SLR is an important part of EBSE. Kitchenham also provided guidelines for researchers to help them with this new methodology [14]. Systematic Review is now considered as an affective research methodology in software engineering by which new and interesting facts can be discovered about a research area based on existing evidences [15].

A Systematic Review has three main phases; Planning, Execution and Reporting. We have carried out all three phases and in this paper we report both the process and the results obtained from it.

The paper is organized as follows; Section II describes Background and Motivation. Section III gives description of research questions. Section IV describes the planning phase of systematic review methodology. Section V describes the execution of SLR. Section VI describes the results. Section VII is discussion and Section VIII is the conclusion. Section IX point out the limitations and section X suggests future work.

II. Background and Motivation

A software change request can have many reasons; e.g. the designers may forget to mention a requirement and it is a missing feature in a resultant system [8] [16] [7], or an error or bug has been identified, the rectification of which is a requirement [17], or misunderstanding of requirements due to the lack of active stakeholder participation [16] [18]. Other examples can be: changes in the priorities of organizations will often motivate changes to the requirements by stakeholders [8] [18] [19], the market trends and competition continuously introduce new requirements to be accommodated [7] [8] [9], new legislation or rules by the organization can force the requirements to change [8] [18] [6] [9], people jump to the solution before understanding what really the problem is [7] [20].

To establish and fully comprehend the implicit causes of requirements change it is imperative to have convincible response to requirements change [21]. The requirements problems can propagate to other phases of SDLC if these issues are not addressed in the early stages, i.e. requirement change can impact the design, and ultimately coding and resulting in a system that is not desirable due to these changes. Change is an issue which can have various effects on the software project, therefore, it is critical to understand the cause and effect relationship for a decision making process in requirement change management. The root cause analysis of requirement change can help to prevent the problems in the requirement phase. It has been observed that if the practitioners have the knowledge of why the requirement change request is presented and what was the underlying cause that generated this request, it would help them in the decision about how to address that change.

The aim of our SLR was to collect empirical evidences for the causes of requirement change. Once the empirical work is brought together it would enable us to see the pattern for causes of requirements change. The knowledge generated by this review will help the project managers and developers in the requirement change management process for analyzing the cause and effect relationship and to make the appropriate decision regarding any requirement change request. Classifying and grouping requirement change is one of the important factors that needs to be examined and analyzed in the requirement change management [18] [22], because it helps the practitioners in understanding the category of requirement change and to evaluate its impact on the software development process and product. One of the objectives of our study is to analyze the causes of requirements change and to categorize them into logical groups. These categories can then be assessed for their frequencies of occurrence during different software development phases. In spite of all the significance of requirement change claimed in reported literature, it has also been pointed out that there is a little empirical work done for research and exploration in this area [6]. By the time we were conducting this Systematic Literature Review, there were only few studies that were using Systematic Literature Reviews in Requirements Engineering. The focus of one was on Requirements Elicitation Techniques [23] [24] and the other was about Requirements Errors [25]. Taxonomy for sources of Requirements Change was proposed by McGee and Greer [26]. Hence by that time no such synthesized study existed to show state of the art in cases of requirement change.

There are different tools, techniques and methods being proposed by the research community to manage the changing requirements. However there is a need for understanding the underlying causes of requirements change in order to better manage its impacts on software development.

III. Research Questions

Our research questions are based on our motivation, i.e. the answer to these questions should provide us with the causes of requirement change which would lead us to perform a categorization of requirement changes. We also want to identify the rate of occurrence of each requirement change category in different SDLC phases. This would help us in identifying when a certain change frequently occurs. We have designed these research questions after an extensive literature review. In addition, the research questions are based on the guidelines given by Kitchenham [14] “Assessing the frequency or rate of project development factor such as the adoption of a technology of the frequency of project success or failure”. On the basis of our research focus, we formulated the following two research questions.

RQ1: What are the different causes of software requirement change?
The objective of this RQ is to identify the different causes of requirement change from the existing literature.

RQ2: What is the rate of occurrence of requirement change types during different SDLC phases?

The objective here is to identify how often a particular category of requirement change occurs in the SDLC phase. The structured form (PICO) of our research questions is as follows;

PICO	RQ1	RQ2
Population	software projects	software projects
Intervention	None	None
Comparison	None	None
Outcome	causes of requirement change	rate of occurrence of requirement change in SDLC

TABLE I. STRUCTURED FORM OF RESEARCH QUESTIONS

IV. SYSTEMATIC LITERATURE REVIEW PLANING

Our review method consists of Search, Planning and Execution by Study Selection, Quality Assessment, Data Extraction, Data Synthesis and generating results. The first deliverable of the process was a SLR protocol [33], which had the planning of the whole process. Based on available resources we selected Springer link, IEEE Explore, ACM Digital library, Cite Seer library, Science Direct and EI Compendex for searching primary studies. The search process was conducted according to the following steps [14]:

1. Deriving major search terms from RQs;
2. Identifying alternative spellings and synonyms for the major terms; also alternative terms used in literature were considered.
3. Using Boolean OR to incorporate alternative spellings and synonyms; and when Boolean AND to link the major terms.
4. Breaking down the strings so that they could be applied to different databases. Assigning Unique IDs to every sub search string. Customizing them for all selected resources to be applied for searches.
5. Using End Notes and Zotero, to manage the resultant references and citations from the search process where supported.

We derived our major search terms from structured research questions, PICO (Table I).

RQ1: *(Software Projects, Causes, Requirement change)*
RQ2: *(Software Projects, Rate, Occurrence, Requirement Change, SDLC, Phases)*
We had 3 terms for RQ1 and 5 for RQ2. We did pilot testing using these terms in order to find all possible variations of the terms used in literature.. Due to the restrictions imposed by online databases we broke down our search strings into 36 different sub strings for RQ1 and 20 sub strings for RQ2. First two substrings for RQ1 and RQ2 are provided below as an example. The rest of the strings and their customized forms for all databases can be found in [33].

RQ1:
SRQ1.1 ((software OR "software project") AND reason AND requirement AND (chang OR volatil*))*
SRQ1.2 ((software OR "software project") AND rationale AND requirement AND (chang OR volatil*))*

RQ2:
SRQ2.1 ((software OR "software project") AND rate AND requirement AND chang)*
SRQ2.2 ((Software OR "software project") AND frequen AND requirement AND Chang*)*

Study selection process was performed by two researchers of the group in order to reduce the personal bias issues. We have used the following steps in the inclusion and exclusion criteria for RQ1 and RQ2;

	RQ1	RQ2	Decision against criteria
Step1 (Initial Selection)	Is this study related to requirements change in software projects and to see whether they actually provided causes of requirement change?	Is the study related to requirements change in software projects and to see whether they actually provided frequency of requirement change in SDLC?	If YES, go to Step2 If NO, Exclude
Step2 (Final Selection)	Are the causes derived from empirical evidence?	Are the frequencies derived from empirical evidence?	If YES, Include If NO, Exclude If Uncertain, forward for group discussion

TABLE II. INCLUSION/EXCLUSION CHECK ITEMS

The secondary search process was performed after the inclusion/ exclusion of primary searches. In the secondary search we scanned the references of included papers to find papers which were relevant to our research questions but were missed during the primary search. We mainly retrieved two types of studies, observational studies and case studies. For observational studies we used the quality checklist provided by Kitchenham [14] and for case studies we used the checklist developed by Höst and Runeson [27]. The weights for the answers to the observational study are, Yes=1, No=0, Partial=0.5, NR=not related. Quality

assessment was carried out separately by three different members of the team to remove any bias. The conflicts were resolved with discussion among all team members. We had designed two data extraction forms; one for general information extraction (e.g title ,author(s), year of publication, citation etc.) and one for extracting data related to our RQ [33]. Data was extracted and saved in the predesigned data extraction forms. It was done by two members separately.

V. SYSTEMATIC LITERATURE REVIEW EXECUTION

"No year" restriction was applied on the search. The execution of the search strategy was carried out during the period of December 2008 to March 2009. Therefore the retrieved results had papers published up till March 2009. The following table shows the overall summary of the search results.

	IEEE	ACM	Science Direct	Springer link	Citeseer	EI compendex	Total
RQ1	668	37	118	137	0	1898	2858
RQ2	184	0	15	35	0	159	393

TABLE III. SUMMARY OF RESULTS

The planned selection process had two parts: an initial selection from the search results of papers that could plausibly satisfy the selection criteria, based on a reading of the title and abstract of the papers; followed by a final selection from the initially selected list of papers that satisfy the selection criteria, based on a reading of the entire papers. The selection process was performed by two researchers and they have identified 3 papers as shown in Table IV.

Database	Total retrieved	Initial Selection	Final Selection
IEEE	668	35	2
ACM	37	3	1
Citeseerx	137	10	0
EI Compendex	1898	39	0
Science Direct	118	9	0
Springerlink	0	0	0
Total	**2898**	**96**	**3**

TABLE IV. SUMMARY OF INCLUSION/EXCLUSION FOR RQ1

Using SLR and secondary searches processes we have identified 5 papers in total as shown in Tables V and VII.

ID	Citation	Database
CRC-1	Ebert, C. and J. De Man (2005). Requirements uncertainty: influencing factors and concrete improvements. Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on.	ACM
CRC-2	Stark, G., A. Skillicorn, et al. (1998). A micro and macro based examination of the effects of requirements changes on aerospace software maintenance. Aerospace Conference, 1998. Proceedings., IEEE.	IEEE
CRC-3	N Nurmuliani, Didar Zowghi, Sue Fowell, "Analysis of Requirements Volatility during Software Development Life Cycle", Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04), 2004.	Secondary Search
CRC-4	Parastoo Mohagheghi, Reidar Conradi, "An Empirical Study of Software Change: Origin, Acceptance Rate, and Functionality vs. Quality Attributes", Proceedings of the 2004 International Symposium on Empirical Software Engineering (ISESE'04), 2004.	Secondary Search
CRC-5	Tamai, T. and A. Itou (1993). Requirements and design change in large-scale software development: analysis from the viewpoint of process backtracking. Software Engineering, 1993. Proceedings., 15th International Conference on.	IEEE

TABLE V. FINAL SELECTION OF STUDIES FOR RQ1

After analyzing the abstract of the two papers included in the secondary search, we found that they were not retrieved by our primary search process because they did not have one of our major search terms in their abstracts. The searching phase is concerned with retrieving all the studies that are relevant to our research question [28]. During searching for primary studies we have two main concerns; sensitivity and precision of search results [29]. By sensitivity we mean the ability of search strategy to identify all relevant papers. The precision of the search is the number of relevant papers retrieved among all the resulted papers. In our search we always have a trade-off between sensitivity and precision and by bringing a balance between these two we can have an optimal search [29]. We had *((software OR "software project")* as a compulsory term in all our string. The reason was that, during our pilot study our search strings, without this term, retrieved a large number of irrelevant studies from other domains of engineering.

Database	Total retrieved	Initial Selection	Final Selection
IEEE	184	27	2
ACM	0	0	0
Citeseerx	35	3	0
EI Compendex	159	4	0
Science Direct	15	0	0
Springerlink	0	0	0
Total	**393**	**34**	**2**

TABLE VI. SUMMARY OF INCLUSION/EXCLUSION FOR RQ2

For RQ2, two independent team members performed the inclusion/exclusion process and conflicts were resolved with discussion among all team members. Two papers were selected from IEEE and one paper was selected in the secondary searches as shown in Table VI. These papers were already included in the results of RQ1. Instead of giving new ID numbers we used the same numbers as they were given in the results for RQ1. The papers with ID CRC-3, CRC-4, and CRC-5 from Table V were selected for RQ2. The following table shows the summary of the whole process for both RQ1 and RQ2.

	Total	Initial Selection	Final Selection	Secondary Search	Final Inclusion
RQ1	2858	96	3	2	5
RQ2	393	34	2	1	3

TABLE VII. COMPLETE SUMMARY OF INCLUSION/EXCLUSION

The quality assessment of the selected studies was carried out by three team members independently. Among selected five papers, three were observational studies and two were case studies. The quality assessment checklist along with scores is given in Appendix A. To rank the studies together we evaluated their weights out of 10. We did not consider these weights to the extracted results from the papers as the number of finally selected papers is only five and we do not have any strong conclusion where the quality of the reporting paper can be questioned.

The next step was data extraction and was again performed by two team members separately.

VI. RESULTS

The following table shows the general information regarding the included studies;

ID	# of studies	Date of Review	Geographical Location	Year of publication
CRC1	One	6TH ~ 9TH JULY 09	France	2005
CRC2	One	6TH ~ 9TH JULY 09	USA	1998
CRC3	One	6TH ~ 9TH JULY 09	Australia	2004
CRC4	One	6TH ~ 9TH JULY 09	Norway	2004
CRC5	Two	6TH ~ 9TH JULY 09	Japan	1993

TABLE VIII. GENERAL INFORMATION ABOUT SELECTED PAPERS

The following table represents the extracted data to answer Research Question 1;

Study ID	Cause of Requirement Change
CRC-1	Requirement Uncertainty is the cause of requirement change Causes of Requirement uncertainty • Vague product vision and strategy • Key stakeholders are not involved • Unknown project dependencies • Business case not thoroughly evaluated • Requirements not sufficiently specified and analyzed.
CRC-2	System/System interface specification incorrect/inadequate). Functional specification (inaccurate/inadequate). User manual (training inadequate).
CRC-3	Customer needs / Market Demands Developer's increased understanding of the product Changes in Organization's Policy
CRC-4	Improving or Enhancing Product internally by Project Organization Improving or Enhancing Product externally by Customer Changing Environment
CRC-5	User understanding / Learning of system

TABLE IX. DATA EXTRACTION FOR RQ1

For Research Question 2, we extracted the rate of the occurrence of change with respect to the SDLC phase from selected studies.

Study ID	SDLC Phase	Rate extracted from studies
CRC-3	Requirement analysis	6.90%
	End of requirement analysis	16.85%
	Design	4.44%
	End of Design phase	6.45%
	Code and Software Integration Testing	4.83%
	End of Integration Testing	1.02%
	Software Acceptance Testing and field test	0.68%
CRC-4	Before implementation	47.3%
CRC-5	*CASE A*	
	Preliminary design- requirement analysis	14%
	Detailed design – requirement analysis	19%
	Programming/testing-requirement analysis	3%
	Operation-requirement analysis	4%
	Detailed design- preliminary design	27%
	Program testing- preliminary design	11%
	Operation- predesign	5%
	Programming/testing-detailed design	15%
	Operation-detailed design	2%
	CASE B	
	Prototyping-requirement analysis	38%
	Preliminary design-requirement analysis	2%
	Detailed design – requirement analysis	1%
	Programming/testing – requirement analysis	1%
	Detailed design-pre-design	8%
	Programming/testing-pre design	20%
	Programming/testing- detailed design	30%

TABLE X. DATA EXTRACTION FOR RQ2

VII. DISCUSSION

We only found five papers (six studies) providing evidence for causes of requirements change. It is surprising to find little empirical work on causes of the requirements change as requirements change has been widely quoted as one of the major challenges faced by software engineers and project managers. In order to check the causes of requirements change mentioned in non empirical literature we extracted the causes of requirement change from those papers which were excluded in our second step of inclusion/exclusion criteria. We then compared them with the causes found in empirical evidences.

The comparison of the causes extracted from the empirical evidence found in literature with the causes of requirements

change reported in non-empirical literature is provided in table XI.

SR #	Causes from SLR		Causes from non empirical Literature
1	Requirement Uncertainty		
	1.1	*Vague Product Vision*	
	1.2	*Key Stakeholders not involved*	Stakeholder Engagement in Requirements Elicitation [16] [18]
	1.3	*Unknown Project Dependencies*	
	1.4	*Business case not thoroughly evaluated*	
	1.5	*Requirements not sufficiently specified and analyzed*	Missing requirements [8] [16] rewording requirements text, redundant requirements, resolving interdependencies [8] loosely defined initial requirements [16] [6]
2	System /System Interface Specification (Inaccurate/Inadequate)		
3	Functional Specification (Inaccurate/Inadequate)		
4	User Manual (training Inadequate)		
5	Customer Needs/ Market Demands		Software expansion, changing mission requirements or hardware platforms, External factors; changes in the tax law or changes derived from business process reengineering studies [7] aggressive market competition [7] [8] changes in user needs [8] Market needs [8] [9] system complexity, techniques [9]
6	Developers increased understanding of the product		Partial understanding, Ill defined requirements, Increasing robustness [7] Increase in developers' knowledge [8]
7	Organizational Consideration		change in Policies [8] organizational Factors [18] Fast changes of business requirements [19]
8	Improving or Enhancing Product internally by Project Organization		Error correction and feature changes [17] design improvement, Clarification changes , test scenario changes, Functionality enhancement / upgrade [8]
9	Improving or Enhancing Product externally by Customer		Error correction and feature changes [17] test scenario changes [8]
10	Changing Environment		rapidly changing technology, environmental factors, technology changes [8] environmental changes [8] [18] [6] government regulation[9] changing legislation during the project [16]
11	User Understanding/ Learning of System		System Use and User Development [18] Clarification changes [8] Partial understanding of requirement, ill-defined situations [7] increase the autonomous system's robustness [30] Discovery of requirements during design [20]

TABLE XI. COMPARISON OF EMPIRICAL AND NON EMPIRICAL CAUSES OF REQUIREMENTS CHANGE

The table XII represents the causes mentioned in non empirical literature, which could not be compared to the empirical results.

Ref#	Causes of Requirement Change [non empirical literature]
[4]	Product Strategy, Changes to media packaging/ licensing/ branding, Hardware/Software, Scope reduction, Lack of resources, Testability
[5]	Software development is a dynamic process. This often causes software requirements to change while development is still in progress. Most of the requirement changes were caused by the organization instead of the techniques, and the maturity of a company had a close relation with the pattern of requirements. Two key factors influencing the requirement change: One is the communication between software users and developers; the other is the method of requirements analysis and modelling.
[7]	Sometimes unnecessary requirements can be left out during the project. Due to shrinking budgets or running out of schedule it is also possible that certain parts of the requirements {missing requirements} are left out of the current project.
[8]	Situated Action and Task Variation Constraints of Planned Organizational Development

TABLE XII. CAUSES THAT CANNOT BE MAPPED TO EVIDENCES OF SLR BUT ARE MENTIONED IN LITERATURE

The idea was to compare them and see whether the non empirical literature is quoting the same reasons for requirements change or not. Most of the causes were supporting the empirical evidence but some of them did not have any empirical grounds. The comparison of the empirical and non empirical evidence highlights that some of the causes mentioned are semantically the same e.g. "requirement not sufficiently specified" is written as "missing requirements", "redundant requirements" and "loosely defined initial requirements". Similarly, for "customer needs/market demands" we have "changes in user needs" and "aggressive market competition". The causes of requirement change found in the empirical evidence which could not be compared to non-empirical evidence are "inaccurate or inadequate System/System Interface Specification", "inaccurate or inadequate Functional Specification", "vague product vision", "unknown project dependencies", "business case not thoroughly evaluated", and "user manual training

inadequate". The reason for that in our opinion is due to the difference in the level of abstraction. "Requirement uncertainty" is at a higher abstract level while other causes such as "vague product vision" etc are at low level of abstraction. The remaining causes of requirement change are at one abstract level and are thus also mentioned in the non empirical evidence. On the other hand there are many causes which are found in non empirical literature presented in table XII but they are not present in empirical evidence. This identifies the need to conduct more empirical studies to identify causes of requirement change. Strong mapping can be seen in the case of "requirement not sufficiently specified and analyzed"; "customer needs", "changing environment", and "users' understanding/learning of the system, organizational consideration and developers increased understanding of the product". The requirement change cause "improve or enhance product internally by the project organization" comes under organizational consideration. Similarly "Functional specification (incorrect/inadequate)" and "system interface specification (incorrect/inadequate)" are considered the same as "requirement not sufficiently specified and analyzed".

The extracted data for RQ2 provides the rate of requirement change over different phases of software development. The rate is mentioned in percentages. There are a total of 4 studies (Table X) which highlight the rate of requirement change during different phases of software development. Due to the variance in the extracted data we are unable to identify any pattern for requirement change. The only prominent pattern that could be seen in the extracted data was the highest rate of the change of requirements is during the requirement analysis phase. The evidence does not report the rate with the specific type of changes, thus we are also unable to associate the rate of requirement change with a specific change category, which was one of our initial objectives.

VIII. CONCLUSION

Brooks [31] classified software engineering difficulties as essential and accidental. Borrowing his terms, we have classified the extracted causes of requirements change as essential causes and accidental causes (Table XIII). We have observed in this study that essential causes are inherent in nature and we do not have control over them, for instance "changing market demand" is an essential cause as software development team or organization cannot control or avoid this. On the contrary, accidental causes can be controlled and avoided. "Business case not thoroughly evaluated" is an accidental cause for requirements change as techniques and mechanisms can be implemented to avoid this cause. Classifying causes for requirements change into essential and accidental causes is a novel way of looking at causes for requirements change and we believe it will help in an effective and efficient management of requirements change. We have observed in this study that in the essential causes focus should be on devising and employing techniques for efficiently dealing with their impact and hence trying to reduce the time and effort for requirements change management. We have also observed in this study that for accidental causes, focus should be on devising and employing techniques and quality procedures for avoiding their occurrence. The causes of requirements change can also be classified based on their origin. Causes can originate from within the project (e.g. "incomplete requirements specification") or causes can originate from outside the project as well. Outside the project, there are two possibilities; from the client organization (e.g. "change in organization policy") or from the business environment in which the client organization operates (e.g. Market demands). Different techniques are required to be devised and employed to deal with different type of causes and their impact.

	Business	Organizational	Project
Essential	Market Demands Changing Environment	Changing Organizational policy	Developers increased understanding of the product Users' understanding / learning of system
Accidental		Vague product vision and strategy Business case not thoroughly evaluated	Key stakeholders are not involved Unknown project dependencies Requirements not sufficiently specified and analyzed User manual/training inadequate

TABLE XIII. FINAL SUMMARY OF RESULTS

IX. LIMITATIONS OF THE RESULTS

The limitations in our results from the review process are as follows:

- Small number of evidences retrieved by the process has restricted us from formulating any solid conclusion.
- In Software Engineering no standard terms are being defined for researchers to use in their papers and the authors use inconsistent keyword selection [28] [32].
- Different key terms are used in different studies for mentioning same cause of requirement change. There were semantic problems while we had to understand what the authors were saying.
- The causes retrieved were not on the same level of abstraction, so the synthesis could not reveal proper results.

- Due to limited resources we are unable to claim that we have used all the available digital libraries. However, the digital libraries used are enough for the initial findings in our study.

X. Lessons Learned and Suggestions

Following are the lessons learned and a few suggestions related to the overall process of conducting the SLR:

- SLR is a good way of knowing the gaps in reported literature. A lot of publications talk about causes of requirements change, but only few investigated it empirically.
- The online data bases are not very helpful in carrying out the search process for Systematic Reviews. There is a need for a system, specifically designed for SLR and it should provide all the translation work with existing electronic databases.
- There should be some guidance on dealing with analysis of different types of studies. All the studies that we retrieved used different research design even when following same methodology. Due to this reason we were unable to assess them for quality as it could not be done with single quality instrument. Some means are required where different research designs can be evaluated and compared for quality.
- Automation of the process is needed to reduce the amount of labor work.

XI. Future Work

We found little empirical evidence for both RQ1 and RQ2. Some of the root causes of requirements change, mentioned in non empirical literature are not supported by empirical literature. There is a need to verify them by conducting more empirical studies to reach a useful conclusion. The causes of requirements change seem to have a complex cause and effect relationship. The experiences of the practitioners working in the industry would be highly valuable in forging an insight into the true nature of these causes. Therefore we strongly recommend further empirical work in the form of case studies and survey etc. on identifying and evaluating the causes of requirements change and their relationship with each other, as they are considered to have an impact on the success or failure of software projects.

References

[1] G. De Michelis et al., *Cooperative information systems: A manifesto*. Citeseer, 1996.

[2] B. J. Williams, J. Carver, and R. Vaughn, "Change Risk Assessment: Understanding Risks Involved in Changing Software Requirements," in *Proc. International Conference on Software Engineering Research and Practice, Las Vegas, Nevada*, 2006.

[3] B. Curtis, H. Krasner, and N. Iscoe, "A field study of the software design process for large systems," *Communications of the ACM*, vol. 31, no. 11, pp. 1268-1287, 1988.

[4] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques. 1998*. Chichester, UK: John Wiley and Sons.

[5] S. Ferreira, J. Collofello, D. Shunk, and G. Mackulak, "Understanding the effects of requirements volatility in software engineering by using analytical modeling and software process simulation," *Journal of Systems and Software*, vol. 82, no. 10, pp. 1568-1577, 2009.

[6] D. Zowghi and N. Nurmuliani, "A study of the impact of requirements volatility on software project performance," in *Software Engineering Conference, 2002. Ninth Asia-Pacific*, 2002, pp. 3-11.

[7] C. Jones, "Strategies for managing requirements creep," *Computer*, vol. 29, no. 6, pp. 92-94, 1996.

[8] N. Nurmuliani, D. Zowghi, and S. P. Williams, "Using card sorting technique to classify requirements change," 2004.

[9] C. Mao, Y. Lu, and X. Wang, "A study on the distribution and cost prediction of requirements changes in the software life-cycle," *Unifying the Software Process Spectrum*, pp. 136-150, 2006.

[10] J. F. Hoorn, "Software Requirements: Update, Upgrade, Redesign," 2006.

[11] C. Ebert and J. De Man, "Requirements uncertainty: influencing factors and concrete improvements," in *Proceedings of the 27th international conference on Software engineering*, 2005, pp. 553-560.

[12] VersionOne, "Survey: 'The State of Agile Development'," Survey.

[13] B. Kitchenham, D. Budgen, and O. P. Brereton, "Evidence-based Software Engineering and Systematic Literature Reviews," *Product-Focused Software Process Improvement*, pp. 3-3, 2006.

[14] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," *Engineering*, vol. 2, no. EBSE 2007-001, 2007.

[15] M. Staples and M. Niazi, "Experiences using systematic review guidelines," *Journal of Systems and Software*, vol. 80, no. 9, pp. 1425-1437, 2007.

[16] G. P. Kulk and C. Verhoef, "Quantifying requirements volatility effects," *Science of Computer Programming*, vol. 72, no. 3, pp. 136-175, 2008.

[17] L. Lin and J. H. Poore, "Pushing requirements changes through to changes in specifications," *Frontiers of Computer Science in China*, vol. 2, no. 4, pp. 331-343, 2008.

[18] S. D. P. Harker, K. D. Eason, and J. E. Dobson, "The change and evolution of requirements as a challenge to the practice of software engineering," in *Requirements Engineering, 1993., Proceedings of*

IEEE International Symposium on, 1993, pp. 266-272.

[19] L. Aversano, T. Bodhuin, and M. Tortorella, "Assessment and impact analysis for aligning business processes and software systems," in *Proceedings of the 2005 ACM symposium on Applied computing*, 2005, pp. 1338-1343.

[20] P. J. Byers and M. K. Wilkinson, "Towards an analysis and specification method for IDA systems," in *Information-Decision-Action Systems in Complex Organisations, 1992., International Conference on*, 1993, pp. 118-122.

[21] N. Nurmuliani, D. Zowghi, and S. Powell, "Analysis of requirements volatility during software development life cycle," in *Software Engineering Conference, 2004. Proceedings. 2004 Australian*, 2004, pp. 28-37.

[22] D. Rowe, J. Leaney, and D. Lowe, "Defining systems evolvability-a taxonomy of change," *Change*, vol. 94, pp. 541-545, 1994.

[23] A. Davis, O. Dieste, A. Hickey, N. Juristo, and A. M. Moreno, "Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review," 2006.

[24] O. Dieste, M. Lopez, and F. Ramos, "Updating a Systematic Review about Selection of Software Requirements Elicitation Techniques," in *Proceedings of the 11th Workshop on Requirements Engineering*, 2008.

[25] G. S. Walia and J. C. Carver, "A systematic literature review to identify and classify software requirement errors," *Information and Software Technology*, vol. 51, no. 7, pp. 1087-1109, 2009.

[26] S. McGee and D. Greer, "A Software Requirements Change Source Taxonomy," in *Software Engineering Advances, 2009. ICSEA'09. Fourth International Conference on*, 2009, pp. 51-58.

[27] M. Host and P. Runeson, "Checklists for software engineering case study research," in *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*, 2007, pp. 479-481.

[28] T. Dyba, T. Dingsoyr, G. K. Hanssen, and T. SINTEF, "Applying systematic reviews to diverse study types: An experience report," in *First International Symposium on Empirical Software Engineering and Measurement, 2007. ESEM 2007*, 2007, pp. 225-234.

[29] O. Dieste and O. A. G. Padua, "Developing search strategies for detecting relevant experiments for systematic reviews," in *First International Symposium on Empirical Software Engineering and Measurement, 2007. ESEM 2007*, 2007, pp. 215-224.

[30] R. Lutz, S. Nelson, A. Patterson-Hine, C. R. Frost, and D. Tal, "Identifying contingency requirements using obstacle analysis," in *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*, 2005, pp. 263-272.

[31] F. P. Brooks, "No silver bullet: Essence and accidents of software engineering," *IEEE computer*, vol. 20, no. 4, pp. 10-19, 1987.

[32] J. Bailey, C. Zhang, D. Budgen, S. Charters, and M. Turner, "Search Engine Overlaps: Do they agree or disagree?," in *Second International Workshop on Realising Evidence-Based Software Engineering, 2007. REBSE'07*, 2007, pp. 2-2.

[33] Muneera Bano, Salma Imtiaz, Naveed Ikram, Muhammad Usman, and Mahmood Niazi, *"Systematic Literature Review for Causes of Requirements Change"*, Technical Report: TR/2010-01, ISSN: 1353:7776, School of Computing and Mathematics, Keele University, Keele, UK, 2010

APPENDIX A: QUALITY ASSESSMENT SCORES

PHASE	CHECKLIST for observational studies	CRC1	CRC2	CRC4
DESIGN	Are the aims clearly stated?	Y	Y	Y
	Are the variables used in the study adequately measured (i.e. are the variables likely to be valid and reliable)?	P	P	Y
	Are the measures used in the study fully defined?	Y	N	Y
CONDUCT	Did untoward events occur during the study?	N	N	N
	Are the data collection methods adequately described?	Y	N	Y
ANALYSIS	Do the researchers explain the data types?	P	N	Y
	Are the study participants or observational units adequately described?	Y	P	Y
	Were the basic data adequately described?	N	N	Y
	Are the statistical methods described?	Y	Y	Y
	Is the statistical program used to analyze data referenced?	N	N	Y
	Are the statistical methods justified?	Y	N	Y
	Is the purpose of analysis clear?	Y	Y	Y
	Are potential confounders adequately controlled for in the analysis?	N	N	N
	Do the numbers add up across different tables and subgroups?	Y	Y	Y
	Was statistical significance assessed?	Y	Y	Y
	If statistical tests are used to determine differences, is the actual p value given?	N	N	Y
	Is there evidence of multiple statistical testing or large numbers of post hoc analysis?	N	N	Y
CONCLUSION	Are all study questions answered?	Y	Y	Y
	Are negative findings presented?	N	N	Y
	If statistical tests are used to determine differences, is practical significance discussed?	NR	NR	NR
	How are null findings interpreted? (I.e. has the possibility that the sample size is too small been considered?)	NR	NR	NR
	Are important effects overlooked?	Y	Y	N
	How do results compare with previous reports?	N	N	N
	Do the researchers explain the consequences of any problems with the validity/reliability of their measures?	N	N	Y
TOTAL	**OUT OF 21**	**11**	**7**	**19**

TABLE XIV. QUALITY ASSESSMENT OF OBSERVATIONAL STUDIES

Sr#	CHECKLIST for Case Studies	CRC3	CRC5
1	Are the research questions, objects of study and case study context well defined?	1	0.5
2	Is it motivated that the case is suitable to address the research questions?	1	1
3	Are the hypotheses and propositions clear and relevant?	0.5	0.5
4	Are the data collection procedures sufficient for the purpose (data sources, collection, storage, validation)?	1	1
5	Are sufficient raw data presented to provide understanding of the case?	0	0
6	Are the analysis procedures sufficient for the purpose (repeatable, transparent)?	1	0
7	Is the case study based on theory and linked to existing literature?	0.5	0.5
8	Is a clear chain of evidence established from observations to conclusions?	1	1
9	Are threats to validity analyses addressed in a systematic way?	0 .5	0
10	Are different views taken on the case (multiple collection and analysis methods, multiple authors)?	1	1
11	Are ethical issues addressed properly (personal intentions, integrity issues, consent, review board approval)?	0	0
12	Are conclusions, implications for practice and future research, reported suitably for its audience?	1	0.5
TOTAL OUT OF 12		**8.5**	**6**

TABLE XV. QUALITY ASSESSMENT OF CASE STUDIES