2009 Fourth International Conference on Software Engineering Advances

A Software Requirements Change Source Taxonomy

Sharon McGee[1] and Des Greer[2]
School of Electronics, Electrical Engineering and Computer Science
Queens University
Belfast, United Kingdom
{[1]smcgee08|[2]des.greer}@qub.ac.uk

***Abstract*— Requirements changes during software development pose a risk to cost, schedule and quality while at the same time providing an opportunity to add value. Provision of a generic change source taxonomy which makes the distinction between factors contributing to requirements *uncertainty* and events that *trigger* change will support requirements change risk visibility, and also facilitate richer recording of change data. In this paper we present a collaborative study to investigate and support the management of software requirements volatility within the development lifecycle. Previously published change 'causes' are elicited from the literature, consolidated using expert knowledge and classified using card sorting. The resulting change trigger taxonomy constructs were initially validated using a small set of requirements change data, and deemed sufficient and practical as a means to collect common requirements change source statistics across multiple projects.**

Keywords- Requirements change; requirements uncertainty; requirements management; project management; card sorting.

I. Introduction

While accepting that requirements changes are inevitable during software development, the increased cost of changes later in the development lifecycle [1][2], combined with the threat that volatility poses to project schedule, cost [3][4], and defect rates [5][4], means that requirements volatility constitutes one of the top ten risks to successful project development [6]. Pfleeger [7] recommends that "We must find a way to understand and anticipate some of the inevitable change we see during software development".

Working with an industrial partner, our shared objective is to design and conduct a series of studies that collectively address this challenge. Our aims are 1) To investigate the correlation between the source of change and requirement type and 2) To assess the impact of change source upon requirements volatility. For the purpose of change management, it is generally recommended that change requests are held in a database with attributes such as 'origin' and 'change type' [8]. An obvious starting point would therefore be to analyse existing change control databases. However, it has been observed that reasons for change are insufficiently recorded *for the purpose of analysis* [9]. While this statement cannot be said to apply generally, it has also been the experience of the authors. In this study we set out to build a taxonomy of requirements change based on the source of the change. This classification of change sources should be useful as a pick-list (along with other pre-defined attributes) in change diaries across multiple projects within one organization, for the purpose of future analysis.

Thus, the following questions are addressed:-

1. What are the sources of requirements change?
2. How can they be classified according to change source domain?

This paper is organised as follows. Section 2 describes previous studies related to the classification and causes of requirements change. Section 3 outlines the research approach and methods used in this study. In Section 4 we describe the research process, and illustrate the derived taxonomy. Section 5 discusses our findings with respect to previous work, and outlines possible application limitations. Finally we end our paper with conclusions and further work.

II. Related Work

Studies designed to classify requirements changes, beyond addition, deletion and modification fall into one of two camps. The first are those that advocate the need for a domain-specific taxonomy. Lam [10-12], who addresses the problem of managing volatility by process control, recommends that volatility classification should capture the domain-specific nature of change in order to facilitate change estimation and reuse. This is echoed by Stark [13] who analyses the impact of maintenance changes on release schedule.

By contrast, a second more generic approach is taken by Harker [14] who divides empirically gathered requirements changes into five categories depending upon the source of the change - fluctuations in the organization and product market environment, increased understanding of requirements, consequences of system-usage, and changes necessary due to customer migratory or adaptation issues. Based on Harker's study, an appraisal by Sommerville [15] includes compatibility requirements relating to business process change in place of migratory and adaptation issues. Working from data held in a change control database within an industrial setting, Nurmuliani [16] catalogues volatility by type (addition, subtraction, deletion), origin, and reason for change. Noting that most change requests used in the study had little information about the reason for change, a further study was undertaken using card sorting to classify the recorded changes [9]. This resulted in a list of 'super-ordinate constructs' classified by reason for change – product strategy, hardware/software environment changes, scope reduction, design improvement, missing requirements,

978-0-7695-3777-1/09 $26.00 © 2009 IEEE
DOI 10.1109/ICSEA.2009.17

clarification changes, testability and functionality enhancement.

As can be seen there is little agreement in the terminology used for classifying requirements change, and it would seem at first sight that studies to date have little commonality. This may be due to the different contextual basis of the studies, or perhaps that classification was established at different levels. It is possible, for example, that Nurmuliani's change reason of 'missing requirement' is included within Harker's change source of 'increased understanding'.

A genre of studies related to requirements engineering risk and uncertainty is also of relevance. Mathiassen [17] classifies requirements engineering risks by reliability, complexity and availability, and relates these to appropriate techniques.

Due to the divergence of change sources compiled in these studies, none of the classifications exclusively met the needs of the subsequent stage of this research. However, their findings, together with requirements change causes derived from other studies are used to provide a collection of change constructs upon which to base our classification effort. Appendix 1 contains a full list of change source constructs elicited from the literature.

III. RESEARCH APPROACH

This study is the first of a family of studies [18] employing a collaborative research approach, in that it seeks to contribute to the body of knowledge in this area, whilst answering to the need of our industrial partner to better understand, manage and measure requirements changes. Collaboration with industry is generally recommended to ensure relevance and better transfer of research results [19]. In this instance the industrial partner gave of their time to provide expert knowledge of project management.

A. Preliminary Studies

To explore the scope and complexity of the problem, and decide upon appropriate and effective research methods, a number of initial investigations were undertaken. Three unstructured interviews during which project managers in the main reflected upon their current project demonstrated the need for a focus for 'memory-jogging'. A subsequent self-administered questionnaire exposed difficulties with change construct interpretation and understanding, and a review of a change database revealed that not all changes were recorded, particularly those relating to the technical solution. Therefore methods were sought that would maximise the opportunity for consensus building, provide a visual basis for brainstorming, and maximise the potential for knowledge sharing and exchange.

The unit of analysis is our industrial partner organization. Participants were sampled from the company's Project Managers by convenience, within the stratum of those with at least 12 years experience in IT.

B. Organisational Context

Our industrial partner in this research employs 300 staff, 136 of whom are involved with software development. They have 6 offices around the UK and Ireland and deliver IT solutions to clients across both the public and private sectors. Most of their contracts involve a single customer and roughly 80% of these relate to governmental work. Nearly all project managers are Prince2 certified and work with a range of traditional and agile methodologies.

C. Workshops

In requirements engineering, group elicitation techniques such as workshops aim to foster stakeholder agreement and buy-in [20], and are a mechanism whereby individuals can make decisions through the consensus building leadership of a facilitator [21]. In view of this, they were used to familiarize all participants with the constructs, come to a common understanding of their meaning, and reach a consensus of opinion at the end of the study regarding the structure of the taxonomy to be used.

D. Card Sorting

Card sorting is a knowledge elicitation technique which involves categorizing a set of cards into distinct groups according to a single criterion. Maiden [21] suggests that card sorting is one of the most suitable techniques for acquiring knowledge of data (in contrast to knowledge of behavior or process). Salient amongst its advantages are its simplicity, focus on participants terminology, and ability to elicit semi-tacit knowledge [22]. A special edition of the journal 'Expert Systems' in 2005 [23] was dedicated to the subject and it has widespread use in psychology, knowledge engineering and requirements engineering. Accordingly, card classification was deemed an appropriate means of deriving a taxonomy of change sources.

IV. TAXONOMY DEVELOPMENT

This section describes the process of taxonomy development.

A. Construct Development

Electronic keyword searches were performed to assemble candidate academic papers, industrial articles, and text books. Citations referring explicitly to requirements change sources/causes/uncertainty/creep/risk were followed in a forward direction in search of the initial source. This resulted in a total of 73 papers and text books which were reduced to a final 14 sources by the criteria 'discovered empirically' or 'seminal work/text book'. As 'seminal work' was subjectively assessed, this cannot be considered a systematic review. However, without this criteria, papers such as 'Issues in requirements elicitation' [24] would not have been included. The authors felt that this would be an oversight.

During the collation of change source constructs it became apparent that reasons for change such as 'diverse user community' and 'New tools/technology' were often gathered together under the umbrella term 'cause' [14, 15, 25]. Clearly there is a distinction between *uncertainty* giving rise to change and events that *trigger* a change. Whilst an event can lead to a change without preceding uncertainty, uncertainty can not result in a change unless an event resolves or intervenes to mitigate the risk of uncertainty. It could be argued that change is 'caused' by a combination of uncertainty and trigger, although in reality causation cannot be proved to arise from one, other or both due to the presence of confounding environmental factors. Accordingly, uncertainties and triggers, collectively referred to as sources of change, were separated. This separation was not difficult since in most cases the semantics of the constructs related to an *event* (trigger) or a *situation* (uncertainty).

B. Initial Workshop

The first workshop taking 2 ½ hours introduced the constructs to 3 project managers and each construct was clarified for meaning. In so doing, constructs sharing a similar meaning were amalgamated, and those represented by other constructs at a finer level of granularity were removed. Additionally, constructs such as 'New Functional Feature', which would necessarily arise as the consequence of resolved uncertainty or opportunity were also removed. The most debated of the constructs was 'changes following prototyping'. Though quoted as a cause of change, it was the opinion of the participants that this change source should be thought of as a technique, having no more causal significance than other techniques such as 'requirements inspections'. Either all techniques should be included and constructs added accordingly, or constructs pertaining to increased understanding should represent the techniques. The final consensus favoured the latter argument, though the addition of technique constructs remains a question for further research. Four triggers were added and as a result of this process, the number of constructs was reduced from 73 to 46. Making the distinction between trigger and uncertainty was confirmed both to be viable and useful, since triggers could more easily be attributed to requirements changes. The constructs are listed in Appendix 1 under the headings 'Trigger Construct' and 'Uncertainty Construct'. What remained was to classify the triggers and assign uncertainty constructs accordingly, thus endorsing the classification and confirming that uncertainties had corresponding change events.

C. Participant Card Sorting

Individual card-sorting ensured that the opinions and contribution of each project manager were represented. The process was first validated by a pilot card sort with 1 project manager and 1 researcher. Each card sorting session was audio-recorded and reviewed, and photographs were taken of card classifications. This process took between 45 minutes and 1 ½ hours.

Each of the 23 trigger constructs (as they appear in Appendix 1) was written on a card and assigned a random number which could be seen clearly in the photographs. Six participants were asked to classify the triggers according to their source.

All participants classified the triggers into between 3 and 5 categories, and there was homogeneity between the classifications, although in all cases they were named differently. For example, one project manager referred to 'ownership' of the categories; another used process labels such as 'customer interface' and 'Requirements engineering'. Naming convention aside, 14 of the 23 constructs were placed in the same pattern by all participants, that is, co-resided in 3 groups. Notably, differences of card placement related to degree of granularity of classification. For example, 4 participants grouped 'increased customer understanding' and 'first engagement of customer' alongside constructs relating to understanding the technical solution. The classifications of the remaining 2 project managers conveyed the importance of distinguishing between changes that arose due to increased understanding of the problem, and those relating to the technical answer to that problem. Only 1 classification illustrated the distinction between market factors and those concerning the customer organisation, the remainder considering them similarly 'external' to the project.

D. Final Workshop

Four project managers attended a final workshop lasting 3 ½ hours. Stimulating and interesting discussion resulted in a unanimously agreed trigger taxonomy to which uncertainty constructs were attributed.

Beginning with 3 untitled groups containing a total of 14 trigger constructs, it remained to come to a consensus of opinion regarding the remaining 9. As observed by one of the participants, the granularity differences were a matter of perception. For example, as a project manager, constructs such as 'market stability' or 'customer organisation strategic change' were equally external to their control. However, from the perspective of a customer, this is perhaps not the case. Therefore, the final taxonomy was built according to the variance of classifications made during the card sorting procedure. Consequently, a taxonomy comprising 5 groups was derived and agreed.

These groups comprised the change domains illustrated in Table 1. Uncertainty constructs were attributed to their associated domain. At this stage several additional uncertainty constructs were added. Most notable amongst these were technical uncertainty, and technical complexity of solution. Though considered general project risks [26], they had not previously been recognised as a source of requirements change. This may be because they do not alter the vision of the problem, but rather the way in which the problem is addressed.

TABLE I. CHANGE DOMAINS AND DESCRIPTIONS

Change Domain	Description
Market	Differing needs of many customers, government regulations, external to project.
Customer Organisation	Strategic direction of a single customer, customer organisation considerations, external to project.
Project Vision	Problem to be solved, product direction and priorities.
Requirements Specification	Specifying the requirements of the established problem.
Solution	Technical answer to problem.

Nonetheless, as discovered by Curtis [25] 'creeping elegance' is a source of change and a risk to budget and schedule slippage. There was some debate about the positioning of 'project size'. Initially considered to be a risk to change in all domains, it was further reasoned that size has an effect due to the increased difficulty of conceptualizing the problem. Therefore 'size' was placed in the domain of project vision.

The derived taxonomy is shown in figure 1. The change domains relate to both triggers and uncertainties. There is a many to many relationship between the uncertainties and triggers within each change domain. Of interest was the observation by the project managers that the structure of the domain also reflects the amount of control they have of the uncertainties, with least control at the top - 'Market' and tighter control at the bottom – 'Solution'.

E. Validation of Change trigger Constructs

The capability of the change trigger constructs to describe the source of a change was initially validated by one of the participating project managers who used a small sample of changes (13) across two projects to ensure that each had a corresponding trigger which accurately reflected the source of change. No changes were made at this stage. This taxonomy will firstly be used within the context of development, assessed for informative capability and internal validity, before considering the broader scope of applicability. Further validation of these trigger constructs is the subject of an on-going study using a current project.

V. DISCUSSION

This section evaluates the taxonomy thus derived with respect to previously published change classifications and outlines some possible limitations of this work.

A. Comparison with Previous Classificatons

The classification proposed in this study bears little synergy with change reasons derived by Nurmuliani [16] as many of these reasons such as 'missing requirement' and 'new functional feature' were considered to be consequences of other events, rather than sources of change. By comparison, there is some resemblance to the classification of change sources defined by Harker [14]. In particular, a combination of *market* and *customer organisation* domain sources equate to Harker's 'mutable' class defined as "changes that arise in response to demands outside the system". By making the distinction between changes that occur in response to market demands, and those answering to

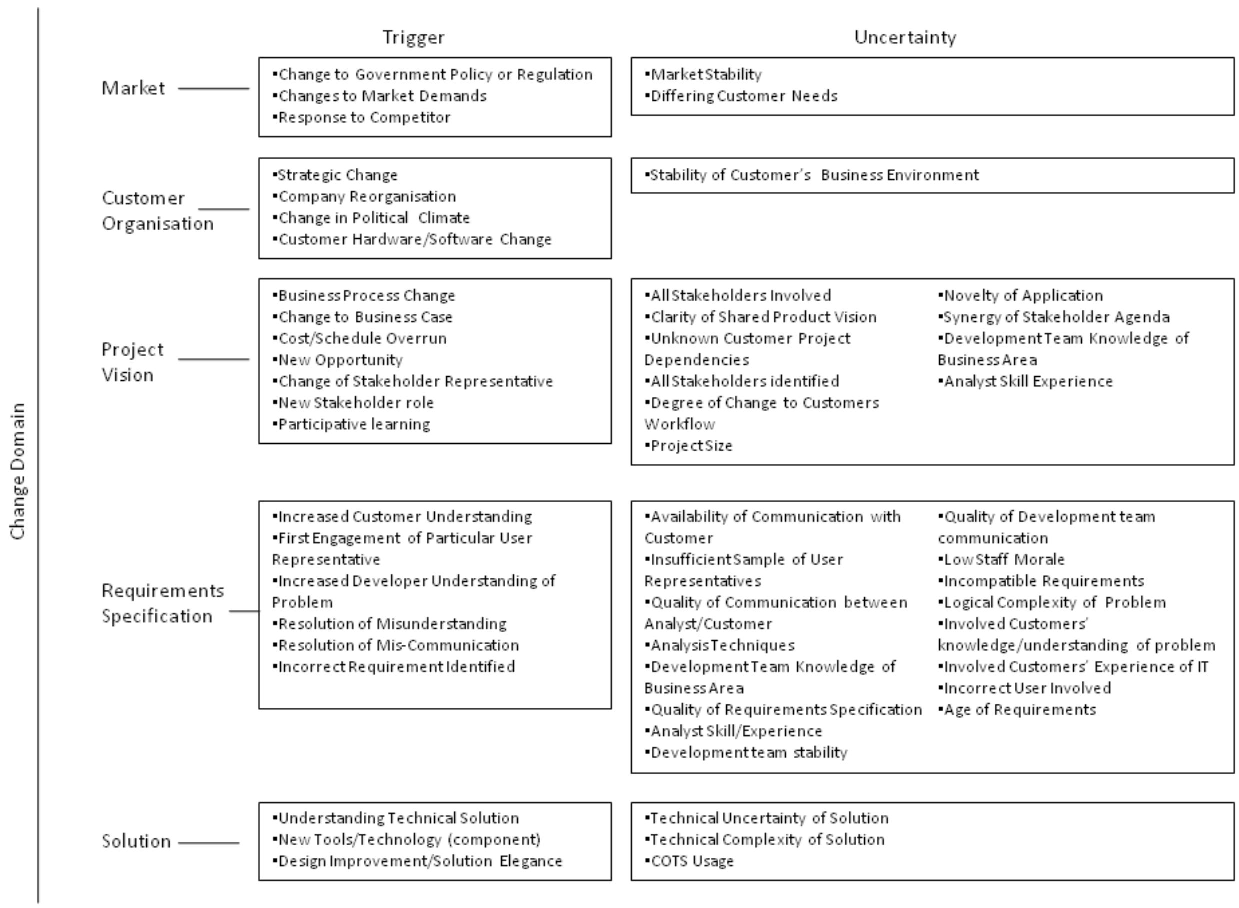

Figure 1. Software Requirements Change Source Taxonomy

customers' organisational considerations, the taxonomy developed here reflects the difference between customer-driven and market driven software development. Harker's 'emergent' requirements, "direct outcomes of the process of engagement in the development activities", correspond to constructs in both the *project vision* and *requirements specification* domain. In differentiating between *project vision* and *requirements specification* domains we are recognising the difference between variation in the product to be developed and change due to better understanding of the problem. This is an important distinction as it can support decisions regarding requirements elicitation techniques and rigour of documentation. There are no analogous domains within this taxonomy for the remainder of Harker's categories. These include prototyping or system usage, adaptive requirements and migration requirements, which were reasoned to be techniques, maintenance considerations, or new requirements. Sommerville's classification [15], while including Harker's 'mutable', emergent' and 'consequential' change (system usage) also removes adaptive and migration requirements. Instead 'change to business process' form a category which is included here in the *project vision* domain, since these types of changes result in a change of product direction. The *solution* domain has no direct parallel in any classification but reflects the reality that changes to the technical solution, though perhaps less visible, pose a risk to timely development.

While there are some differences in contained constructs, requirements availability as defined by Mathiassen [17] corresponds to *requirements specification* although constructs relating to requirements complexity and reliability are included in both *customer organisation* and *project vision.* That said, further categorising these domains according to reliability and complexity would allow the findings of both studies to be combined, thus relating technique to change source domain.

Of particular significance for our on-going research is that by comparison to the descriptive or uncertainty based nature of previous work, the clearly defined constructs within each change source domain allow comparative source data to be attributed to change databases. Therefore it would be possible to assess the impact on the project of a particular change source such as 'new stakeholder' or 'first engagement of customer representative', giving software providers some empirical data with which perhaps to leverage customer involvement. Further, it would be possible to assess the level of change in each change source domain. Should, for example, a high proportion of changes come from the domain of *project vision*, this would indicate the vulnerability of the 'problem' to change, thereby empirically illustrating the need for more 'agile' creative processes.

B. Limitations

Generality of results are often sacrificed for richness and complexity, reflecting an inherent conflict between internal and external validity [19]. Given the disparity between published change taxonomies and the debate among the participants of this study, it could be argued that change classification is by nature a subjective assessment. Motivated, however, by the potential for improvement to requirements change visibility and management, modelling change sources is a worthy initiative. The collaborative approach taken here has led to an internally usable model and reflects Sjoberg's recommendation [19] to "formulate scope relatively narrowly to begin with and then extend it gradually" Therefore, while no claims to external validity can be made, given that the constructs were drawn from a variety of empirically based studies, it is plausible that the results apply to contexts beyond the boundaries of this study. The initial constructs are provided here, along with methods description such that it should be possible to replicate this study. Given the collaborative nature of this research, and its immediate applicability, it has a high level of relevance.

VI. CONCLUSION AND FURTHER WORK

This study used the expert knowledge of experienced project managers to consolidate and classify change source constructs elicited from the literature. The resulting taxonomy made the important distinction between uncertainty (situation) and trigger (event) giving rise to change, supporting requirements engineering risk visibility and facilitating the collection of clearly defined change source data. In differentiating between source domains pertaining to market, customer and project vision a software provider using this taxonomy can assess the level of changes that are arising due to a *change in the direction or vision* of the problem, by contrast to those pertaining to an *increased understanding* of the problem to be solved. In so doing, project managers can make use of internal empirical data to support process and technique selection, and risk management.

This study was founded upon previously published requirements change taxonomies, thus evaluating and building upon their efforts. Therefore it addresses the problems of divergent change source constructs, and reasons that some of the classifications previously described as 'causes' were either consequences of other changes, types of requirements, or maintenance considerations outside of the scope of the software development lifecycle.

Having answered the questions posed in this study, it is now possible to further our research and begin exploring what kinds of requirements are more susceptible to change arising within the change domains defined in this taxonomy. This is currently on-going with our industrial partner. In the meantime the derived taxonomy can assist practically in the identification and analysis of requirements volatility and has particular relevance to customer driven software development especially those working within the government sector.

ACKNOWLEDGMENT

We would like to thank the project managers whose valuable time was devoted to the organization of this taxonomy.

REFERENCES

[1] N. Nurmuliani, D. Zowghi and S. P. Williams, "Requirements Volatility and Its Impact on Change Effort: Evidence-based Research in Software Development Projects", Australian Workshop on Requirements Engineering, Adelaide, 2006.
[2] B. Williams, J. Carver and R. Vaughn, "Change Risk Assessment: Understanding Risks Involved in Changing Software Requirements", Proc. International Conference on Software Engineering Research and Practice, Las Vegas, Nevada, 2006.
[3] D. Zowghi and N. Nurmuliani, "A study of the impact of requirements volatility on software project performance", Proc. Ninth Asia-Pacific Software Engineering Conference, 2002.
[4] S. Ferreira, F. Collofello, D. Shunk, G. Mackulac and P. Wolfe, "Utilization of Process Modeling and Simulation in Understanding the Effects of Requirements Volatility in Software Development", International Workshop on Software Process Simulation and Modeling, Portland, Oregon, 2003.
[5] T. Javed, M. Maqsood and Q. Durrani, "A study to investigate the impact of requirements instability on software defects", ACM Software Engineering Notes, 29, 3, 2004.
[6] B. Boehm," Industrial Software Metrics Top 10 List", IEEE Software, 4(5), 1987.
[7] S. L. Pfleeger, "Software Metrics: Progress after 25 Years? ", IEEE Software, 25(6), 2008.
[8] K. Wiegers, Software Requirements, Microsoft Press, 2003.
[9] N. Nurmuliani, D. Zowghi and S. P. Williams, "Using card sorting technique to classify requirements change", Proc. 12th IEEE International Conference on Requirements Engineering, Kyoto, Japan, 2004.
[10] W. Lam, M. Loomes and V. Shankararaman , "Managing requirements change using metrics and action planning", Proc. 3rd European conference on Software Maintenance and Reengineering, Amsterdam, Netherlands, 1999.
[11] W. Lam and V. Shankararaman, "Requirements change: a dissection of management issues", Proc 25th EUROMICRO Conference, Milan, Italy, 1999.
[12] W. Lam and V. Shankararaman, "Managing change in software development using a process improvement approach", Proc. 24th Euromicro Conference, vol 2, Vasteras Sweden, 1998.
[13] G. Stark, A. Skillicorn and R. Ameele, "An Examination of the Effects of Requirements Changes on Software Releases", CROSSTALK, The Journal of Defense Software Engineering, 1998.
[14] 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", Proc. IEEE International Symposium on Requirements Engineering, San Diego, CA, USA, 1993.
[15] I. Sommerville, Software Engineering, Personal Education Ltd, 2007.
[16] N. Nurmuliani, D. Zowghi and S. Powell, "Analysis of requirements volatility during software development lifecycle", Proc. Australian Software Engineering Conference, Melbourne, 2004.
[17] L. Mathiassen, T. Saarinen, T. Tuunanen and M. Rossi , "Managing Requirements Engineering Risks: Analysis and Synthesis of the Literature", Helsinki School of Economics Working Papers W-379, 2004.
[18] D. Perry, A. Porter and L. Votta, "Empirical Studies of Software Engineering: A Roadmap", Proc. 22nd International Conference on Software Engineering, Limerick, Ireland, 2000.
[19] D. I. K. Sjoberg, T. Dyba and M. Jorgensen, "The future of Empirical Methods in Software Engineering Research", Proc. Future of Software Engineering, Minneapolis, MN, 2007.
[20] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap", Proc 22nd International Conference on Software Engineering, Limerick, Ireland, 2000.
[21]N. A. M. Maiden and G. Rugg. "ACRE: selecting methods for requirements acquisition", Software Engineering Journal, 11(3), 1996.
[22] S. Fincher and J. Tenenberg, "Making sense of card sorting data", Expert Systems, 22(3), 2005.
[23] Anonymous, Expert Systems Special Edition on Card Sorting, 22, 3, 2005.
[24] M. Christel and K. Kang, "Issues in Requirements Elicitation", Technical Report No. CMU/SEI-92-TR-012 Software Engineering Institute, 1992.
[25] B. Curtis, H. Krasner and N. Iscoe, "A field study of the software design process for large systems", Communications of the ACM, 31(11), 1988.
[26] H. Barki, S. Rivard and J. Talbot, "Toward an assessment of software development risk", Journal of Managment Information Systems, 10(2), 1993.
[27] B. Boehm, "Requirements that handle IKIWISI, COTS, and rapid change", Computer, 33(7), 2000.
[28] A. Lamsweerde, Requirements Engineering : From System goals to UML models to software specifications, John Wiley & Sons Ltd, 2009.
[29] R. Pressman, Software Engineering. A Practitioner's Approach, McGraw Hill, 2005.
[30] T. Moynihan, "'Requirements-uncertainty': should it be a latent, aggregate or profile construct?" Proc. Australian Software Engineering Conference, Canberra, ACT, Australia, 2000.
[31]T. Moynihan. "How experienced project managers assess risk", IEEE Software, 14(3), 1997.
[32] A. M. Davis and K. V. Nori, "Requirements, Plato's Cave, and Perceptions of Reality", Proc. International Conference on Computer Software and Applications, Beijing, China, 2007.
[33] C. Ebert and J. De Man, "Requirements uncertainty: influencing factors and concrete improvements", Proc. 27th International Conference on Software Engineering, ST Louis, Missouri, USA, 2005.

Appendix 1

ID	Trigger Construct	Source	Removed
45	Use of Prototype	[27] , [14]	Technique, covered by 21, 44
68	New Stakeholder (role)	[27]	
96	Customer Company Reorganization	[28], [16], [14], [27], [29]	
51	New solution Tools/technology	[27], [28], [25], [14], [8]	
54	Change to government policy or regulation	[28], [14],[8]	
20	Participatory Learning	[14],	
28	Local Customization	{14]	New Stakeholder
50	Customer migration to new solution	[14]	Type of requirement
39	Customer need change	[24], [25], [16], [29], [29]	Too woolly, convered by 47, 96, 54, 82, 21, 45, 42, 90
44	Developers Increased Understanding of problem	[25], [16],[17]	
56	Scope Reduction	[9][16]	By-product of 88, 66
34	Changes to packaging/licensing/branding	[9]	Removed, Covered by 61
65	Solution Elegance (Design Improvement)	[25], [9]	
67	Resolution of mis-communication	[9]	
49	Testability	[9]	
82	Business Process change (continuous improvement)	[15], [8]	
42	Response to competitor	[25], [8]	
16	Functionality Enhancement	[16]	Covered by 78, 65
11	Defect Fixing	[16]	Doesn't result in requirement change
69	Redundant Functionality	[16]	Covered by 66, 44, 20, 90, 67, 23, 51, 65, 21
8	Missing Requirement Identified	[16]	Not a reason/cause/source
86	Clarification of Requirement	[16]	Covered by 67, 23
21	Increased customer understanding	[28], [15], [8], [17]	
72	New Class of User	[28]	Result of other changes, covered by 82, 68
74	New Usage Condition	[28]	Covered by 78, 85
15	New way of doing things	[28]	Covered by 82, 96
77	Correction to Requirements specification	[28]	Covered by 23, 67
78	New Opportunity	[28]	
1	Change in the use of the information	[17]	Covered by 82
88	Cost or schedule overrun	[29], [28]	
49	Testability	[16]	Type of requirement
85	Change to Customer's hardware/software	[9]	
58	System Usage (after installation, not prototype)	[27], [15], [14]	Out of scope of project development
90	Changes to Market Demands	[8], [14], [16], [29]	
62	Resolution of Conflicting Requirement	[16]	Covered by 83
55	New Functional Feature	[28]	New Requirement
3	Improved Quality Feature	[28]	Change to requirement for another reason
14	Result of Change in political climate (needs of particular group emphasized)	[24], [8], [25]	
93	Change to customer's environment	[15]	Covered by 96, 68, 85, 47, 66
18	Changes in Underlying technologies	[25]	Covered by 85, 51
83	Incorrect Requirement Identified	[16]	
23	Resolution of Misunderstanding	[25]	
92	First or re-engagement of user representative	Added	
66	Change to business Case (Return on Investment, Total cost of Ownership	Added	
61	Customer Organization Strategic Change(New Marketing/Sales direction, change to organization goals)	Added	
12	Change of Stakeholder Representative	Added	

	Uncertainty Construct	Source(s)	Removed
31	Analyst skill or experience	[4], [24], [30], [8]	
95	Development team knowledge of business area	[24], [31]	
79	Analysis techniques employed (workshops, interviews, modeling etc)	[4], [8], [15], [29], [28]	
59	Project Size	[24], [17], [15]	
30	Novelty of product (business novelty)	[31], [8]	
41	Logical complexity of problem	[17], [31], [24]	
19	Availability of communication with customer	[17], [8]	
22	Involved customer's knowledge/understanding/clarity of requirements	[24], [32], [31], [8]	
64	Quality of Communication between analyst and customer	[24], [17], [32], [8]	
33	Involved customers experience with working alongside IT to produce solutions	[31]	
9	Diverse User Community	[31], [15], [8]	Summary of 29, 27, 40, 41, 2, 60
32	Incompatibility between requirements	[28]	
24	Lack of well understood model of utilizing system	[17]	unclear
55	Lack of well-understood model of the utilizing system	[17]	unclear
6	Lack of structure for activity or decision being supported	[17]	Covered by 22
52	Stability of Customers Business Environment	[24]	
76	COTS usage	[28], [27]	
2	All stakeholders identified	[27], [33]	
29	All Stakeholders involved	[24], [33], [27], [8]	
40	Clarity/unity of shared product vision	[30], [14], [33], [8]	
27	Synergy of stakeholder agenda	[28], [31], [14]	
43	Unknown Customer Project Dependencies	[33]	
46	Market Stability	[25], [32], [14]	
13	Differing Customer Needs	[25]	
38	Type of user doing specification (incorrect user involved)	[8][30] [17]	
89	Change in the utilizing system	[17]	unclear
80	Low Staff morale	[4]	
10	Large number of users	[17]	Not a risk if correct user involved - 38
87	Level of participation of users in specification	[17]	Covered by 19, 64
81	Lack of user experience of utilizing system	[17]	Covered by 22
63	Degree of Change to customers workflow	[31], [8]	
48	Quality of Requirements specification	Added	
71	Technical Uncertainty of Solution	Added	
84	Technical Complexity of Solution	Added	
53	Quality of Development team Communication	Added	
73	Age of Requirements(elapsed time since completion of requirements documentation)	Added	
60	Insufficient Sample of User Representatives	Added	
35	Development team (PM and analyst) stability	Added	