

# Software Requirements Change Taxonomy: Evaluation by Case Study

Sharon McGee<sup>1</sup> and Des Greer<sup>2</sup>

School of Electronics, Electrical Engineering and Computer Science  
Queens University  
Belfast, United Kingdom  
{<sup>1</sup>smcgee08|<sup>2</sup>des.greer}@qub.ac.uk

**Abstract**— Although a number of requirements change classifications have been proposed in the literature, there is no empirical assessment of their practical value in terms of their capacity to inform change monitoring and management. This paper describes an investigation of the informative efficacy of a taxonomy of requirements change sources which distinguishes between changes arising from ‘market’, ‘organisation’, ‘project vision’, ‘specification’ and ‘solution’. This investigation was effected through a case study where change data was recorded over a 16 month period covering the development lifecycle of a government sector software application. While insufficiency of data precluded an investigation of changes arising due to the change source of ‘market’, for the remainder of the change sources, results indicate a significant difference in cost, value to the customer and management considerations. Findings show that higher cost and value changes arose more often from ‘organisation’ and ‘vision’ sources; these changes also generally involved the co-operation of more stakeholder groups and were considered to be less controllable than changes arising from the ‘specification’ or ‘solution’ sources. Overall, the results suggest that monitoring and measuring change using this classification is a practical means to support change management, understanding and risk visibility.

**Keywords**- *Requirements Change; Requirements Evolution; Collaborative Case study*

## I. INTRODUCTION

Software requirements continue to evolve during software development and maintenance, and the associated risk to cost, project schedule and quality appeals to the need for increased understanding of the phenomena. The case-study introduced here is the second in a family of collaborative empirical initiatives, each of which addresses objectives related to the ultimate goal of requirements change anticipation. Requirements changes can vary greatly in terms of their cost and value; the metric ‘requirements changes = 2’ which results from the addition of one change costing £100 to a second change at a cost of £1000 is not that informative. The first step, therefore, is to establish a means by which a change can be classified and measured. A system of classification intended for the purpose of change measurement and monitoring should be practical and easy to apply to changes, as well as reflective of cost and and/or value. A previous study [1] addressed the observation that existing classifications were incomplete, or difficult to use, and established standardized constructs to represent the

reason or cause of the requirements change. The study used the expert knowledge of experienced project managers to consolidate and classify 73 change source constructs elicited from the literature. Using individual card sorting and workshops, a classification of change sources was derived comprising the five change domains illustrated in table 1. In addition, an important distinction was made between constructs relating to a situation such as ‘insufficient stakeholder involvement’ and those relating to an event such as ‘business process change’. A full taxonomy relating the domains in table 1 to uncertainties (situational constructs) and triggers (event constructs) can be found in the appendix. With the initial focus on software development, the taxonomy was extended to include the maintenance phase of a project [2]. However, the informative or explanatory value of categorizing requirements change in this way, or any other, has not been determined.

TABLE 1 REQUIREMENTS CHANGE SOURCE DOMAINS

CHANGE DOMAIN	DESCRIPTION
Market	Differing needs of many customers, government regulations.
Customer Organisation	Changing strategic direction of a single customer, customer organisation considerations, political climate.
Project Vision	Change to the problem to be solved, product direction and priorities, stakeholder involvement, process change.
Requirements Specification	Change to the specification of the requirements of the established problem, resolution of ambiguity, inconsistency, increased understanding.
Solution	Change accommodating new technical requirements, design improvement, solution elegance.

The academic objective of the case study introduced here is to provide an empirically founded evaluation of the potential of the requirements change source taxonomy to provide a meaningful and practical means of change classification and measurement. At the same time there is an immediate business objective to improve visibility and understanding of requirements change. Effort was therefore required to clearly identify research questions and define

mutually expedient case study data. The following research questions are addressed:

Across change domains, is there a significant difference in:

- 1) Change cost;
- 2) Change value;
- 3) Proportion of opportunity vs. defect related change;
- 4) The activities during which changes are found;
- 5) The number of stakeholders involved; and
- 6) The level of project management control?

With our industrial partner, the Goal Question Metric (GQM) approach [3] was largely adhered to in order to firstly articulate these questions and secondly identify case study data. Past change data were used as the basis of discussion, and this was supported by UML modeling of project processes and work products which enabled the identification of the possible values of the variables under study. The project under investigation designed and delivered a solution within the government sector, lasting 16 months having a total cost of 4222 days effort. Overall 282 changes were recorded at a total cost of 2405.5 days effort.

This paper is organized as follows. Following a review of related research in section 2, the design of the case study, including variable selection and data collection protocol is presented in section 3. Section 4 introduces the results and these are discussed alongside the limitations of the study in section 5. Section 6 concludes and outlines the future direction of this work.

## II. RELATED WORK

As far as the authors are aware, there is no existing study that uses an empirical basis for the evaluation of requirements change classifications. (The reader is referred to [1] for details of the associated literature review). This is substantiated in a comprehensive literature review of change based studies undertaken by Banested [4]. In this review, three primary objectives for empirical studies of requirements change are identified, among them the characterization of evolution. A number of classifications have been proposed, focused upon software development, maintenance, or both, which often have the intention of meeting different objectives.

A traditional classification of change during software development includes the categories add, modify and delete. This has been used in the prediction of requirements change [5], as the measure of the health of a project [6] [7], and to support process technique selection [14]. Much empirical and theoretical work focused upon software maintenance re-uses or builds upon Swanson's classification [18] which includes corrective, adaptive and perfective changes. Chapin et al. [19] provide a thorough review of literature referring to maintenance change types, and propose a new classification which focuses upon the type of change being made. Both Kemerer & Slaughter [20] and Heales [21] take a different approach and classify changes according to what is being changed. Alternative empirically derived classifications include that proposed by Harker et al [16], and Nurmuliani [23]. While they share the objective of defining a generic

classification based upon the reason for the change, there is little commonality in either change construct or classification. Sommerville [17] largely adopts Harker's framework. From a different perspective, Nakatani et al [24] consider that different types of requirements mature at different times in the development process, and recommends the categorization of groups of requirements according to maturation type.

The classification under investigation in this study was derived initially from previous empirically founded change classifications that focused upon software development. In so doing, the resulting classification is more exhaustive in terms of change constructs, and can be regarded as a synergy of earlier work. Given its generic nature, it is readily applicable to software development projects and triggers can be used as a pick list when maintaining change data. Importantly, like that of Harker et al [16] and Sommerville [17], the ontological distinction is based upon the source of the change. This facilitates causal analysis which supports change review and management and also may contribute to change anticipation.

### A. Case Study Design

The study was designed in accordance with the case study guidelines outlined by Runeson and Host [11] and Wohlin et al [12] and is a single unit case study, in which the unit of analysis is the requirements change.

### B. Case Study Context

#### 1) Organisation

Our industrial partner in this research employs 300 staff, has offices in England and Ireland, and delivers 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. Of importance to collaborative research, their involvement is supported by both upper and middle management, and reflects their stated initiative to become a centre of project management excellence.

#### 2) Project

The project of interest in this study is in the government sector, has an estimated cost in excess of a million pounds, comprises on average 15 software developers and analysts, and follows a traditional waterfall lifecycle. Beginning in April 2009, the project was completed in August 2010 and data was collected during the entire development lifecycle. Since the software development work was the result of a successful tender, at the commencement of the project, the requirements made available to the software provider during that tendering process became the basis of the initial requirements specification effort. There were four main stakeholder groups involved, comprising the software provider and three departments on the customer side.

### C. Data Specification

As well as supporting the needs of the academic objective, the data to be collected will also replace the company's existing change control database and be used for project retrospective analysis. The Goal Question Metric

Approach defined by Basili [3], was operated initially in a focus group setting consisting of a researcher and 2 project managers. In addition to research questions regarding the cost and value of change (questions 1 and 2) the selection of research questions related to management issues (questions 3-6) reflected the needs of our industrial partner to understand and thus better manage their changing requirements. As well as discovering when change was happening, and whether it represented an opportunity to add functionality or attend to a defect, they wanted to determine if a greater number of involved stakeholders influences the number of changes seen. Also, an important issue was whether the change could have been avoided. Project management control was understood to mean ‘With hindsight could/should this change have been discovered earlier’, perhaps by the use of alternative techniques or additional resources.

The selection and practical implementation of metrics to answer the research questions was not straightforward. In the main a pragmatic approach was taken, which often required compromise between research and practice. While it was considered too labour intensive to include metrics for change KLOC, the addition of the data item ‘phase’ was therefore necessary for the analysis of cost comparison since average change costs may increase as the project progresses due to rework, rather than change size. Cost was measured in days and defined as the difference between the original estimate (if it existed) and the actual days effort required to implement the change. The research preference of expressing value in monetary terms was impossible in most situations, so a Likert scale, subjectively assessed, was employed instead. The additional data items ‘change trigger’ and ‘domain’ were added to relate changes to the change domains in table 1. No classification scheme had been previously used by the company, though ad-hoc reasons for change were included in descriptive text.

Given the need to define an agreed and standardized list of activities, UML modeling sessions led by the researcher and involving project managers as available, gave rise to the production of an activity diagram and a domain model. The data specified (excluding those relevant only to practice such as originator, dates etc.) is illustrated in Table 2.

It will be noted that many of the data items are subjective measures. Whilst appreciating the limitations imposed by non-objective measurement upon the analytical significance of results, the collection of subjective measures is becoming more widely accepted and advocated. [13] [14].

#### D. Data Collection Protocol

As changes were discovered, data was collected on a spreadsheet, by either the project manager or the senior analyst. Initially bi-monthly meetings took place to review the changes gathered though these became less frequent, due in part to the urgency of project delivery. The data was owned by the company until project sign-off, whereupon the company removed any company-confidential data before transference for research.

#### E. Data Validation

Effort was made to ensure that correct values have been entered against each change record. Observer triangulation was applied in the case of ‘cost’, ‘value’ and ‘opportunity?’ by the customer and project manager and remaining data items by project manager and senior analyst. Methodological triangulation between the qualitative ‘change description’ and the quantitative factor was achieved during the change review meetings with a researcher and project manager. A number of changes, randomly selected, were reviewed at these meetings. Roughly 60% changes were re-examined, though data quality was high and only a small percentage of changes were amended, usually due to completion of missing data items.

#### F. Data Review Process

During the data review meetings, in addition to data validation, the trigger placement within each change domain was reviewed and the taxonomy amended as required. For example, the change trigger ‘New Market Technology’ was added to the domain of *Market* to differentiate it from the trigger ‘New technology’ residing in the *Solution* domain. Also, some overlap between triggers in the *Requirements Specification* domain and those in the *Vision* domain were identified. For example, ‘Increased Customer Understanding’ could change both the *vision* and the *requirements specification*. A revised taxonomy can be found in the appendix. Perhaps not surprisingly, particularly towards the latter phases of the project the customer and software provider experienced increased difficulty in coming to agreement about whether the change represented an opportunity or a defect. This was evidenced in cases where the customer was expecting something implicit or assumed within the agreed documentation. Therefore an allowable value - ‘Undefined’ - was added to the ‘Opportunity?’ data item.

#### G. Analysis Procedures

Descriptive tables and graphs are complemented by statistical procedures to test hypotheses related to the research questions. These hypotheses are a rephrasing of each research question. For example, question 1, ‘Across change domains is there a significant difference in cost?’ becomes the hypothesis ‘There is a significant difference in cost between change domains’. Results are summarized in terms of these hypotheses. Procedures were selected on the basis of required underlying distribution and variable scale assumptions. Data pertaining to change cost did not follow a normal distribution (see results section D ‘The Cost of Change’) and many of the data items have a nominal (categorical) scale as indicated in table 2. What follows are short descriptions of the appropriate statistical test, and the research questions that they are employed to address.

### H. Hypothesis Testing

The Kruskai Wallis test allows comparison of groups of data scores (ordinal or scale type), and tests whether the scores could be thought to come from different groups, that is, that there is a significantly different central tendency for each group. Simply put, when data does not conform to a normal distribution, (as is the case with the costs recorded in this case study), using this test is one of the ways groups can be compared without reference to mean values. This test uses score rankings in place of actual scores to perform the statistical test. Post-hoc procedures include examination of

pairs of groups to determine where the main differences lie (Mann Whitney test). These tests will be used to examine the change costs observed for changes within each change domain (research question 1). The Chi-squared test looks for relationships between two categorical variables, by comparing the observed frequencies in certain categories with expected frequencies. This test is appropriate for examining the ordinal scale for value as well as the nominal variables selected to represent managerial considerations (research questions 2 to 6). The reader is referred to [25] for details of these tests.

TABLE 2 DATA SPECIFICATION FOR REQUIREMENTS CHANGES

Name/ Research Question	Description	Allowable Values
ID	Unique Identifier	
Trigger (all)	Change Source Trigger Eg. Change to business case, Increased customer understanding, New technology available.(Nominal Objective)	A complete list can be found in the appendix.
Domain (all)	Change Source Classification. This was derived where possible from the trigger using the taxonomy in Appendix 1 and reviewed. (Nominal, Objective).	Market, Organisation, Vision, Specification, Solution.
Phase (1)	Project phase when change identified (Nominal Objective)	Requirements (Req), Design and Code (D&C), System Test (SysTest), User Acceptance Test (UAT).
Discovery_ Activity (4)	Activity during which change was identified (Nominal, Objective)	Provide Business Case, Define Goals, Define Vision, Derive Initial Requirements, Define Functional Requirements, Define Technical Requirements, Define Quality Requirements, Balance Requirements, Approve Requirements, Define Manual Processes, Derive System Requirements, Specify Scenarios, Define Architecture, Build & Unit test, System Test, Specify UAT, Perform UAT, Implement Solution.
PM_control (6)	Project manager's control of change identification (Ordinal, Subjective)	Very low, low, med, high, very high.
Stakeholders (5)	Number of stakeholder roles involved agreeing the change (Ordinal Objective)	One, Two, > Two.
Cost (1)	Change cost expressed in days (Ratio, Subjective)	
Value (2)	Business value to the customer (Ordinal, Subjective)	Very low, low, med, high, very high.
Opportunity? (3)	Opportunity or defect (Nominal, Subjective)	Opportunity, Defect.
Description	Free text – qualitative	

### III. RESULTS

#### A. Overall Look at Changes during the Developmental Lifecycle

From project inception to delivery, over a duration of 16 months, a total of 282 requirements changes were recorded, at a cost of 2405.5 days effort which represents more than 50% of the final project cost of 4222 days. Table 3 illustrates the phase during which these changes were discovered, and the change source domain. Since the project followed a strict traditional waterfall process, the phases are temporally contiguous.

TABLE 3 NUMBER OF CHANGES PER PHASE PER DOMAIN

	Req	D&C	SysTest	UAT	Total
Market	0	1	0	0	1
Organisation	30	4	0	0	34
Vision	15	1	1	7	24
Specification	22	58	5	102	187
Solution	0	33	3	0	36
Total	67 (24%)	97 (34%)	9 (3%)	109 (39%)	282

As can be seen, a high proportion of these changes occurred during the User Acceptance Testing and Design and Code phases of the project. Since this was a project intended for a particular customer rather than a market-based initiative, it is not surprising that there is only one market change (which related to following market trends in COTS usage). This change (costing 30 days effort) was removed for all subsequent analysis, and means that this study is limited to the examination of the remaining four change domains. In addition changes involving only requirement deletions (12) at zero cost are excluded from future analysis, reducing the total number of changes considered from 282 to 269.

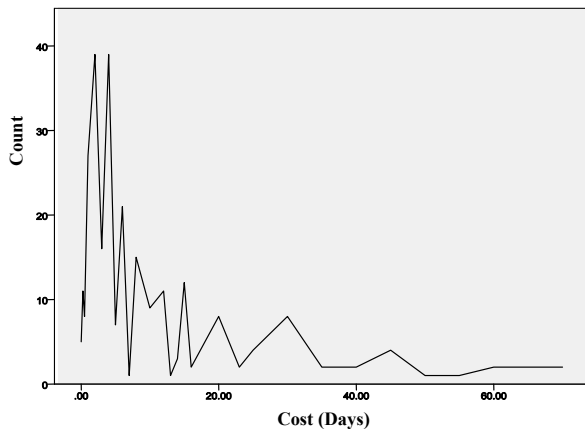


Figure 1. Frequencies of Change Costs Across All Change Domains.

#### B. The Cost of Change

The analysis of change cost discounts the 12 deleted requirements. Fig 1 illustrates the frequencies of change costs for the entire project across all change domains.

Change cost is not normally distributed (Shapiro-Wilk  $W = 0.669$ ,  $p < 0.001$ ), and is highly positively skewed due to the lower limit of zero cost being fixed. Since examination of mean values for cost is therefore less meaningful, future analysis uses the median values as representation of central tendency.

Table 4 shows a breakdown of these costs as they pertain to project phase and change domain. Although the most significant cost was experienced during the initial phase of requirements confirmation, a high percentage of change cost occurred during User Acceptance Testing (38%). By far, the largest percentage of cost came from the *specification* domain (46%).

Figure 2 illustrates a comparison between median and total cost for each change domain. While total costs are significantly higher in the specification domain, the medians of these costs illustrate that on average changes due to organisation changes are the most expensive, followed by changes to the vision, specification, with the lowest average cost in the solution domain. The Kruskal Wallis test indicated that the differences are significant ( $H(3) = 75.038$ ,  $p < .001$ ) to the extent that these changes could be thought of as coming from different groups. While this indicates that there is a difference overall, it does not inform us of where the major differences lie. Performing selected Mann Whitney tests to test for differences between adjacent domains reveals that the median of the domain of organisation does not vary significantly to that of vision ( $U = 229.5$ ,  $z = -1.787$ ,  $p > 0.05$ ), but that vision differs from specification ( $U = 851.500$ ,  $z = -4.879$ ,  $p < 0.001$ ) and similarly specification differs from solution ( $U = 1901$ ,  $z = -4.006$ ,  $p < 0.001$ ). Since costs change over time, it is useful to explore the differences in domain costs for each phase of the project.

TABLE 4 CHANGE COST PER PHASE PER DOMAIN

	Req	D&C	SysTest	UAT	Total
Organisation	638.0	64.0	0.0	0.0	702.0
Vision	266.0	5.0	2.0	163.0	436.0
Specification	193.9	222.0	4.5	737.0	1156.5
Solution	0.0	78.0	2.5	0.0	81.0
Total	1097.0 (46%)	369.5 (16%)	9.0 (0.4%)	900.0 (38%)	2375.5*

\* Excludes Market change at a cost of 30 days

Figure 3 shows median change costs for each domain in each phase. It can be seen that the trend in all domains is generally reflective of the results we saw when all phases were included, with the most expensive changes occurring in the *organisation* domain and the least expensive in the *solution*. Costs tend to fall in the second and third phases, and in the case of the *vision* and *requirements specification* domain rise sharply during User Acceptance testing. From quantitative data alone, it is impossible to assess whether this rise in cost is due to increased change size (more function points per change) or rework of existing code and architecture. Performing the Kruskal Wallis test on phase one data alone indicates that there is significant difference in the costs in the three domains of organisation, vision and specification ( $H(2) = 15.239$ ,  $p < 0.001$ ).

Similarly, overall cost medians are significantly different in phase two ( $H(3) = 10.692$ ,  $p < 0.05$ ). As can be seen though, there is no difference in this phase between costs in the domains of requirements specification and solution. It is not possible to do median comparisons for phases three and four due to insufficient data.

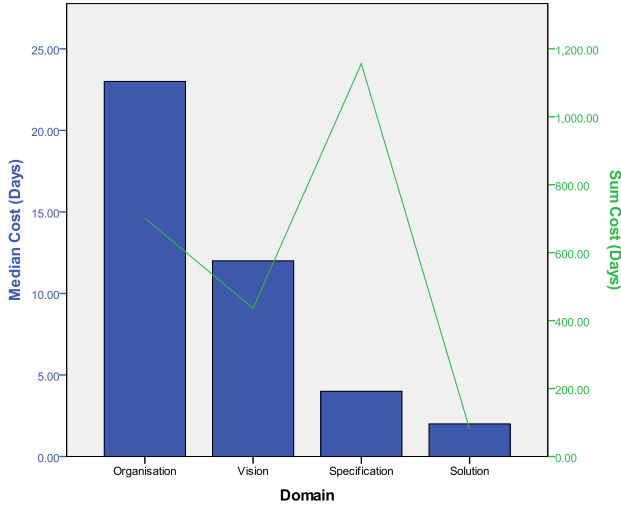


Figure 2. Median & Total Cost For Each Change Domain.

In answer to research question 1, the results support the hypothesis that change costs are not consistent across change domains, the most expensive changes coming from the domain of *organisation* and falling through the domains of *vision*, *specification* and *solution*.

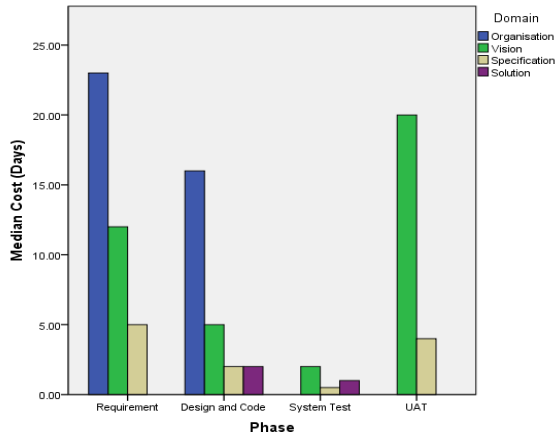


Figure 3. Median change costs per domain per phase.

### C. The Value of Change

Table 5 illustrates the proportion of subjectively assessed value accruing from these changes across the change domains. Just over half of all changes are of a very low value (51%), and of those the greatest proportion (74%) was in the *requirements specification* domain. The highest value changes are only in the domains of *organisation* and *vision*.

By contrast most of the changes in the *solution* domain are of very low value (91%). A chi-squared test (performed with value 'High' and 'Very High' changes added together due to low frequencies in these groups) reveals that there is an uneven distribution of values across the four domains ( $X^2(9) = 144.354$ ,  $p < 0.001$ ).

Answering research question 2, these results support the hypothesis that from the perspective of value, these requirements changes could be thought of as coming from different groups according to the change domains specified. The highest value changes come from the domain of *organisation* and the lowest from *solution*.

TABLE 5 CHANGE VALUE PER DOMAIN

	Value					Total
	Very Low	Low	Med.	High	Very High	
Organisation	0	7	9	5	6	27
Vision	2	9	11	0	2	24
Specification	102	64	13	3	0	182
Solution	33	2	1	0	0	36
Total	137 (51%)	82 (30%)	34 (13%)	8 (3%)	8 (3%)	*269

\* market change and changes representing requirements deletions removed.

### D. Opportunity/Defect

TABLE 6 NUMBERS OF CHANGES BY DOMAIN CATEGORISED AS OPPORTUNITY, DEFECT OR UNDEFINE

	Opportunity	Defect	Un-defined	Total
Organisation	24	2	1	27
Vision	18	5	1	24
Specification	104	62	16	182
Solution	13	20	3	36
Total	159 (59%)	89 (33%)	21 (8%)	269

Changes can represent an opportunity to enhance system functionality as well as the correction of a previous error. Table 6 illustrates how these are spread across the change domains. Changes representing an opportunity comprise the majority of changes in the domains of *organisation* (89%) *vision* (75%) and *specification* (57%), and represent a total cost of 1677 days effort. Defects, costing a total of 559 days effort are more often the cause of change in the *solution* domain (56%). When the customer and software provider have not been able to arrive at an agreement about whether the change represents an opportunity or a defect is has been referred to as 'Undefined'. In this case, most of these changes related to assumptions regarding functionality implementation methods. They represent a small proportion of all changes (< 10%), have a cost of 139.5 days effort, and

are mostly in the *specification* domain. The chi squared test is significant ( $X^2(6) = 21.662$ ,  $p = 0.001$ ) confirming that there is an uneven distribution of opportunity change across these change domains.

**In answer to research question 3**, these results support the hypothesis that the proportion of changes representing an opportunity as opposed to a defect are not evenly spread across domains. Opportunity change is more often seen in the domains of *organisation* and *vision*, while defects predominate the domains of *requirements specification* and *solution*.

#### E. Number of Stakeholders

As the software provider was considered a stakeholder, changes involving only one stakeholder were either those that required decisions to be made without customer involvement, or those where a single customer stakeholder group was able to make changes that required only agreement rather than negotiation with the software provider. A stakeholder number of '3' means three or more stakeholders groups involved in agreeing the change. Table 7 illustrates stakeholder role involvement in each change domain. In all domains there is greater proportion of changes requiring more than one stakeholder role. (89% of *organisation* changes, 96% of *vision* changes, 81% of *specification* changes and 56% of *solution* changes).

TABLE 7 CHANGES CATEGORISED AS NUMBERS OF STAKEHOLDER GROUPS INVOLVED IN AGREEING CHANGE PER DOMAIN

	Stakeholder Groups			
	1	2	3	Total
Organisation	3	18	6	27
Vision	1	15	8	24
Specification	19	149	14	182
Solution	16	20	0	36
Total	39 (14%)	202 (75%)	28 (10%)	269

However, in the domains of *organisation* and *vision*, there are proportionally more changes requiring the involvement of three or more stakeholders (22% and 33% respectively) compared with the *specification* domain (8%) and *solution* (0%). In the *solution* domain we see a greater proportion of single stakeholder changes (44%) than in any other domain. A chi squared test supports the hypothesis that there is dissimilarity in these domains when considering the numbers of stakeholder groups usually involved in the change ( $X^2(6) = 50.795$ ,  $p < 0.001$ ). Interestingly median costs also rise as the number of involved stakeholders increases. The median cost when one stakeholder is involved is 2 days effort, compared with 4 days for 2 stakeholders and rising sharply to 10 days for 3 or more stakeholders.

**In answer to research question 4**, these results support the hypothesis that the number of stakeholders involved in a change is not consistent across change domains. More often

a higher number of stakeholders are involved with *organisation* and *vision* change.

#### F. Discovery Activity

A high proportion of requirement specification changes were discovered during UAT (40%), though many of the organisation changes (62%) and vision changes (45%) were discovered earlier in the developmental lifecycle during the define functional requirements activity. Solution changes in the main were discovered during build and test (63%).

TABLE 8 CHANGE DISCOVERY ACTIVITY PER CHANGE DOMAIN

	Org.	Vis.	Spec.	Sol.	Total
Define Vision	1	1	0	0	2
Define Functional Reqs	17	11	14	1	43
Define Technical Reqs	1	1	3	0	5
Balance Reqs	0	0	1	0	1
Approve Bus Reqs	1	0	0	0	1
Derive System Reqs	4	2	5	2	13
Specify Scenarios	0	0	2	0	2
Define Architecture	1	0	0	2	3
Build and Unit Test	0	1	25	23	49
System Test	0	1	5	8	14
Specify UATs	0	0	26	0	26
Perform UAT	0	7	101	0	108
No Activity	2	0	0	0	2
Total	27 (10%)	24 (9%)	182 (68%)	36 (13%)	269

A visual analysis of these changes, presented in table 8, would suggest therefore that changes in different domains are discovered during different activities in the developmental lifecycle. However, there is insufficient data to perform a chi squared test for inequality of change discovery activity spread amongst domains.

**In answer to research question 5**, there was insufficient data to perform statistical hypothesis testing.

#### G. Project Management Control

As stated, the process followed in this project adhered to a waterfall approach wherein attempts are made to define all requirements at the beginning of the project. 'Project Management control' captures a subjective assessment by the project manager regarding the ease by which these changes may have been discovered earlier. It was felt that some changes would have been impossible to find (pm control = 'Very low') even with improved techniques since they come from external sources of which the project team has no knowledge. An example of a change such as this is

changing the list of internet browsers that the system was intended to be compatible with, following an organisational study of browser usage. By contrast those that the project manager believed may have been uncovered with more time, or different techniques (pm control = 'Very high') would include changes such as screen layout modification.

These results, illustrated in table 9, indicate that all of the changes over which the project manager has the most control lie within the domains of specification and solution. There is a proportionally greater volume of 'Very low' control change in the domain of organisation (26%) than in vision (4%), specification (2%) and solution (3%). As it stands the data is insufficient to perform a chi squared test. However, when pm control = 'Very low' & 'Low' and pm control = 'High' & 'Very high' are compressed into single categories, the data meets the criteria necessary for this test and is significant ( $X^2(6) = 85.113$ ,  $p < 0.001$ ) indicating that in general the level of project management control differs according to the domain from which the change arises.

**In answer to research question 6**, these results support the hypothesis that the level of project management control is not consistent across change domains. It was felt that a higher proportion of *solution* and *specification* changes could have been discovered earlier by the use of alternative approaches or techniques, while much *organisation* and *vision* change would have occurred regardless of analysis effort.

TABLE 9 EXTENT OF MANAGEMENT CONTROL OVER CHANGES PER DOMAIN

	Project Management Control					Total
	Very Low	Low	Med.	High	Very High	
Organisation	7	4	16	0	0	27
Vision	2	3	18	1	0	24
Specification	3	13	126	31	9	182
Solution	1	1	5	18	11	36
Total	13 (5%)	21 (8%)	165 (61%)	50 (19%)	20 (7%)	269

#### IV. DISCUSSION

##### A. Results of Analysis

The analysis of this case study data has allowed us to assess whether there is any correlation between the change taxonomy groups and change attributes reflecting change size, value, stakeholder involvement, and project management control. Results indicate that there is distinction between changes falling into the classifications in this taxonomy. Not only do changes arising due to customer *organisation* changes cost more on average and accrue more value but it was also felt that they are more difficult to uncover, and generally involve the agreement of a higher number of stakeholder roles. This is in stark contrast to

*solution* changes which are in the main controllable and less costly than changes from other sources. This implies that the management approach and assessment of risk to project schedule, cost or quality should be reflective of different type of changes, and that change measurement and monitoring would be more informative if classified in this way. For example to reduce the uncertainty associated with higher risk of customer organisational change, it would be necessary for project analysts to broaden the scope of application analysis to wider organisational concerns. As well as differences in cost and value, there are also differences of management considerations between changes due to *vision* changes and those coming from *specification*. While it was possible to uncover changes from *specification* issues during Build and Test, any vision changes not already discovered were not found at this stage and remained until User Acceptance Testing.

Maintaining change data in this way across multiple projects would allow software providers to assess the efficacy of analysis techniques and guide future process selection decisions. For example, the high number of *vision* changes discovered during hands-on system usage during User Acceptance Testing may provide empirical support for the use of more agile techniques such as early prototyping or iterative delivery. Indeed, while it may be the case that agile techniques assuage late vision change, the observation that many *specification* changes were discovered during build and test may imply that the onus is upon analysis techniques as well as process procedures to reduce the types of changes that arise from *specification* issues.

Since a higher proportion of *organisation* and *vision* changes represent an opportunity to enhance previously agreed functionality as opposed to the correction of defects, the taxonomy also captures the notion that some change should be encouraged, and some types of change avoided. Despite the concerning fact that this project increased in size by over 50% due to requirement changes, over 70% of these changes represented a opportunity to enhance previously agreed functionality rather than correct errors.

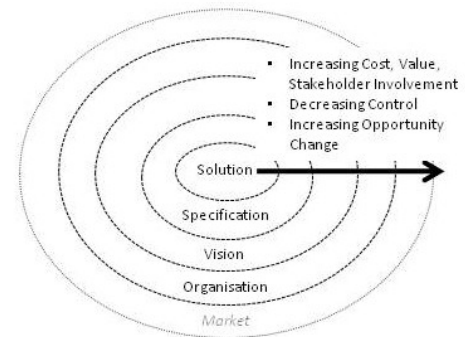


Figure 4. Requirements Change Ontology.

The results are summarized in figure 4. The arrow indicates the tendency for increasing cost, value and opportunity change from the *solution* domain through the *specification*, *vision* to the *organisation* domain. At the same time the level of project management control is decreasing. While



this study did not investigate changes arising from the domain of market, it has been included here for completeness in lighter shading. There is no direct mapping between a requirement and an element in this taxonomy. A single requirement can be thought to comprise a slice consisting of elements of all 5 domains in differing proportions depending upon the developmental phase and position within the requirements hierarchy. While the means of taxonomy derivation in this study has been empirical rather than the theoretical approach taken by Perry [22], it is possible to draw sensible comparison between Perry's 'real world' and our 'market' and 'organisation', his 'model of the real world' with our 'vision', his 'system requirements' with our 'requirements specification' and his 'technical theory' with our 'solution'.

### B. Limitations of Study Validity

Since the data was specified during focus group sessions, there is a shared understanding of the meaning of the data items and therefore little threat to construct validity. No claims to external validity can be made, and ideally this study should be replicated firstly within a similar context to attend to the possibility that the particular environmental or architectural characteristics in the project under study were responsible for the results. Subsequently widening software development context reflects Sjöberg's recommendation [15] to "formulate scope relatively narrowly to begin with and then extend it gradually".

### V. CONCLUSIONS AND FURTHER WORKS

While a number of requirements change classifications have been presented, there has been no attempt to evaluate their practical informative value. Informally, the question asked here is "How does this classification help me understand the consequences of change and why and when it is happening, so that I may be able to monitor and manage better?" The classification considered here is the software requirements change source taxonomy[1] comprising the change domains of *market*, *organisation*, *vision*, *requirements specification* and *solution*. Researchers worked closely with an industrial partner to identify, collect and validate suitable data to facilitate this investigation. While no results are available for the domain of market, findings indicate the following:

- There are significant differences in cost, value, control and stakeholder involvement between changes arising from each of the non-market sources.
- Generally changes from the *organisation* domain are more costly, have a higher value, more often represent an opportunity rather than a defect, but also have increased stakeholder role involvement and are considered less easy to control.
- Through the domains of *vision*, *specification* and *solution*, costs are falling, stakeholder role involvement is decreasing, and there is an increasing level of control.

- There is also some evidence that different activities are more likely to uncover change from particular domains.

The implication is that the assessment of risk and management of changes should be tailored according to the characteristics of these change domains. As a means of monitoring and measurement, use of this taxonomy is feasibly practical and will aid understanding of software evolution during development as well as providing opportunities for retrospective project analysis to aid future process and technique tailoring. In line with our ultimate goal of requirements change anticipation, planned empirical studies include the investigation of the attributes of requirements that may render them more susceptible to changes in certain change domains, and an exploration of software change causality. However, this work also opens other possibilities in terms of alternative lifecycle models, and software maintenance research.

### VI. ACKNOWLEDGEMENTS

We would like to thank the project manager and analysts who gave their valuable time to data specification and the collection for the purposes of this investigation.

### REFERENCES

- [1] S. McGee, D. Greer, "A Software Requirements Change Source Taxonomy", proc. 4<sup>th</sup> Intl. Conf. Software Engineering Advances, 2009, pp. 51-58.
- [2] S. McGee, D. Greer, "Sources of Software Requirements Change from the Perspectives of Development and Maintenance", International Journal on Advances in Software, 2010, pp. 118-200.
- [3] V. R. Basili. "Quantitative evaluation of software methodology", University of Maryland, TR-1519, July 1985.
- [4] H. Benestad, B. Anda, and E. Arisholm, "Understanding software maintenance and evolution by analyzing individual changes: a literature review", Journal of Software Maintenance and Evolution: Research and Practice, vol. 21(6), 2009, pp. 349-378.
- [5] A. Loconsole and J. Borstler, "An industrial case study on requirements volatility measures," 12th Asia-Pacific Software Engineering Conference (APSEC'05), 2005, pp. 249-256.
- [6] C. Jones, "Strategies for managing requirements creep," Computer, vol. 29(6), 1996, pp. 92-94.
- [7] R. Costello and D. Liu, "Metrics for requirements engineering," Journal of Systems and Software vol. 29(1), 1995, pp. 39-63.
- [8] L. Mathiassen, "Collaborative Practice Research", Information Technology and People vol 15(4), 2002, pp.321-345.
- [9] L. C. Briand, M. Differding and H. Rombach, "Practical Guidelines for Measurement-Based Process Improvement," Software Process Improvement and Practice, vol. 2(4), 1996, pp.253-280.
- [10] V. R. Basili, G. Caldiera and H. Rombach, "Experience Factory," Encyclopaedia of Software Engineering, 2002, DOI 10.1002/0471028959.sof110
- [11] P. Runeson and M. Host, "Guidelines for Conducting and Reporting Case Study Research in Software Engineering," Empirical Software Engineering, vol. 14(2), 2009, pp. 131-164.
- [12] C. Wohlin, M. Host and K. Henningsson, "Empirical Research Methods in Software Engineering," Lecture Notes in Computer Science, vol. 2765, 2003, pp.145-165.
- [13] N. Fenton and M. Neil, "Software Metrics: Roadmap," proc. Conf. on Future of Software Engineering, 2000, pp. 359-370.

- [14] S. L. Pfleeger, "Software Metrics: Progress after 25 Years? ", IEEE Software, 25(6), 2008, pp.32-34.
- [15] D. I. K. Sjoberg, T. Dyba and M. Jorgensen, "The future of Empirical Methods in Software Engineering Research", proc. Future of Software Engineering, 2007, pp.358-378.
- [16] 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, 1993, pp.266-272.
- [17] I. Sommerville, Software Engineering , Ed. 9 Addison Wesley, 2010.
- [18] E. Swanson, "The dimensions of Maintenance", proc. 2nd Intl Conf Software Engineering, 1976, pp.492-497.
- [19] N. Chapin, J. Hale, K. Khan, J. Ramil, W. Tan, "Types of Software Evolution and Software Maintenance" Journal of Software Maintenance and Evolution: Research and Practice, 13(1), 2001, pp. 3-30
- [20] C. Kemerer, S. Slaughter, "An empirical approach to studying software evolution" IEEE Transaction on Software Engineering, 25(4), 1999, pp.493-509.
- [21] J. Heales, "Factors Affecting Information System Volatility", Proc. 21st Intl. Conf. Information Systems, Brisbane, Australia, 2000, pp.70-83
- [22] D. Perry, "Dimensions of Software Evolution", proc. Intl. Conf. Software Maintenance, 1994, pp. 296-303.
- [23] N. Nurmiliani, D. Zowghi and S. P. Williams, "Using card sorting technique to classify requirements change", proc. 12<sup>th</sup> IEEE International Conference on Requirements Engineering, 2004, pp. 240-248
- [24] T. Nakatani, S. Hori, M. Tsuda, M. Inoki, K. Katamine, M. Hashimoto, "Towards a Strategic Requirements Elicitation – A proposal for the Prince model", Intl. Conf. Software and Data Technologies, 2009, pp. 145-150.
- [25] A. Field, "Discovering Statistics using SSPS", Sage Publications Ltd, Third Edition, 2009.

#### APPENDIX SOFTWARE REQUIREMENTS CHANGE SOURCE TAXONOMY

	Trigger	Uncertainty
Market	<ul style="list-style-type: none"> <li>•Change to Government Policy or Regulation</li> <li>•Changes to Market Demands</li> <li>•Response to Competitor</li> <li>•Response to Gap in Market</li> <li>•New Market Technology</li> </ul>	<ul style="list-style-type: none"> <li>•Market Stability</li> <li>•Presence Of Competitor</li> <li>•Differing Customer Needs</li> </ul>
Customer Organisation	<ul style="list-style-type: none"> <li>•Strategic Change</li> <li>•Company Reorganisation</li> <li>•Change in Political Climate</li> <li>•Customer Hardware/Software Change</li> </ul>	<ul style="list-style-type: none"> <li>•Stability of Customer's Business Environment</li> </ul>
Project Vision	<ul style="list-style-type: none"> <li>•Business Process Change</li> <li>•Change of Stakeholder Representative</li> <li>•New Stakeholder role</li> <li>•Participative learning</li> <li>•Change to Business Case</li> <li>•New Opportunity</li> <li>•Increased Customer Understanding</li> <li>•First Engagement of User Rep</li> <li>•Cost/Schedule Overrun</li> </ul>	<ul style="list-style-type: none"> <li>•Novelty of Application</li> <li>•Synergy of Stakeholder Agenda</li> <li>•Semantic Relativism</li> <li>•All Stakeholders Involved</li> <li>•Clarity of Shared Product Vision</li> <li>•Unknown Customer Project Dependencies</li> <li>•All Stakeholders identified</li> <li>•Degree of Change to Customers Work-practice</li> <li>•Logical Complexity of problem</li> <li>•Involved Customer understanding problem</li> <li>•Customer experience of IT</li> <li>•Incorrect Stakeholder involved</li> <li>•COTS /framework Usage</li> <li>•Project Size</li> <li>•Development Team Knowledge of Business Area</li> <li>•Analyst Skill Experience</li> </ul>
Requirements Specification	<ul style="list-style-type: none"> <li>•Increased Customer Understanding</li> <li>•First Engagement of Particular User Representative</li> <li>•Incorrect Requirement Identified</li> <li>•Change of Stakeholder Rep</li> <li>•Participative Learning</li> <li>•Increased Developer Understanding of Problem</li> <li>•Resolution of Misunderstanding</li> <li>•Resolution of Mis-Communication</li> <li>•Resolution of Ambiguity</li> </ul>	<ul style="list-style-type: none"> <li>•Availability of Communication with Customer/Stakeholder</li> <li>•Insufficient Sample of User Representatives</li> <li>•Quality of Communication between Analyst/Customer/Stakeholder</li> <li>•Quality of team communication</li> <li>•Incompatible Requirements</li> <li>•Degree of Change to Customers Work-practice</li> <li>•Project Size</li> <li>•Logical Complexity of Problem</li> <li>•Analysis Techniques</li> <li>•Development Team Knowledge of Business Area</li> <li>•Quality of Requirements Specification</li> <li>•Analyst Skill/Experience</li> <li>•Development Team stability</li> <li>•Low Staff Morale</li> <li>•Customers' understanding of problem</li> <li>•Involved Customers' Experience of IT</li> <li>•Incorrect User Involved</li> <li>•Age of Requirements</li> </ul>
Solution	<ul style="list-style-type: none"> <li>•New Tools/Technology (component)</li> <li>•Design Improvement/Solution Elegance</li> <li>•Understanding Technical Solution</li> </ul>	<ul style="list-style-type: none"> <li>•COTS/framework Usage</li> <li>•Technical Uncertainty of Solution</li> <li>•Technical Complexity of Solution</li> <li>•Team experience framework</li> </ul>