Understanding Requirements Volatility in Software Projects – An Empirical Investigation of Volatility Awareness, Management Approaches and their Applicability

Rahul Thakurta
Indian Institute of Management Calcutta,
Joka, Diamond Harbour Road,
Kolkata 700104, India
fp092004@iimcal.ac.in

Frederik Ahlemann
Institute of Research on Information Systems,
European Business School,
D-65375 Oestrich-Winkel, Germany
frederik.ahlemann@ebs.edu

Abstract

Requirements volatility during software project development is known to be the most critical risk, and managing this is paramount to success in software project. The research described in this paper is based on a combination of interviews and a survey in two phases and aims to investigate the organizational practices in dealing with this risk, and how it is influenced by the adopted project execution strategy with regard to process model selection decisions.

The results indicate study participants' heightened perception of the risk of requirements volatility. Thirteen different approaches to managing projects under volatility could be identified, of which the practice of involving the business side was the most frequent. Differences could be observed in the usage of these approaches based on the chosen project execution strategy. The influence of business emerged as the highest determinant of the seven identified factors governing execution strategy selection. The variation in these factors under volatility has also been pointed out. The current practice regarding usage of these execution frameworks reveals a gradual shift towards customization. In this regard some incongruence between perception and practice under requirements volatility was also evident, and has been addressed in the paper. The study findings are expected to assist project managers in their choices related to project administration under requirements volatility.

1. Introduction

Despite advances in software engineering over the past 30 years, most software projects still experience numerous requirements changes during their life cycle, which is brought about by the dynamic nature of development activities [19]. Requirements volatility, which has been defined as the change in requirements (in terms of the number of additions, deletions, and modifications) during software project development [6], has been reported as one of the main factors causing a project to experience challenges [31].The literature has pointed out to the causes and effects of requirement volatility, and also outlined some of the approaches in order to deal with the problem.

The identified causes of requirement volatility include presence of inconsistencies or conflicts among requirements [19], evolving user/customer knowledge and priorities [13], activities carried out during the project like defect fixing, functionality correction [19], technical, schedule or cost related problems [13], change in work environment [13], and process model selection decisions [15]. Process models (also known as the systems development life cycle model) in this regard describe the various stages involved in an information system development project, and provide a mechanism to plan for and manage project execution [22].

The effects of requirement volatility have been stressed in relation to both the process characterizing the project development (for example decrease in productivity [35], reducing team moral and increasing attrition rate [6]), and the final outcome (i.e. schedule and effort overruns [36], increase in number of defects [9]). The final result is an increase in failure rates of projects as has been reported in a Standish Group [32] study.

Various suggestions have been put forward in order to enable effective planning, monitoring, and controlling of the changes as they happen over the software project lifecycle. The management strategies in this regard can be classified as adoption of specific frameworks (like formation of change control boards

IEEE computer society

978-0-7695-3869-3/10 $26.00 © 2010 IEEE

[10], and specifying the project execution strategy upfront like selecting the process model for the project), and adoption of specific techniques during project development (for example usage of joint application design (JAD), and configuration management [10], base lining requirements [33], and usage of contextual management techniques based on the project setting [5]).

It can be seen that the process models have been identified both a cause and also as an approach to managing requirement volatility in projects. It appears that the characteristic of these process models influences requirement volatility, and a careful choice could lead to some control over the problem. In this context, the advantages and disadvantages of some of the available process models (refer to Table 1) in handling volatile requirements have been summarized in Sudhakar [27]. Madachy [15] and Rajlich [21] both observe that the waterfall model is quite rigid under changing requirements.

projects under requirements volatility. Starting with an assessment of project managers' perception of the problem of requirements volatility affecting such decision choices, we examine the pattern of usage of process models as a project execution strategy, and aspects related to its suitability and applicability under requirements volatility. The influence of this upfront execution strategy on usage of subsequent management approaches has also been examined. We expect our study to contribute to the current knowledge base on managing requirements volatility by identification of some of the preferred approaches which software practitioners might find useful. The results are also expected to assist inexperienced project managers in their project management related decisions under requirements volatility. Even though the term "project execution strategy" is much broader in context, we have used it in our paper to consistently refer to process model related decisions made in the project.

Process Model	Definition
Waterfall Model	It is a software development model (with strictly one Iteration/phase) in which development proceeds sequentially through the phases: requirements analysis, design, coding, testing (validation), integration, and maintenance
V-Shaped Model	This is an extension of the waterfall model but instead of moving down in a linear way, the process steps bent upwards after the coding phase in a typical V shape.
Prototyping Model	It is a software development process that begins with requirements collection, followed by prototyping and user evaluation
Incremental-Iterative Model	Here the software project is divided into mini-projects, each of which is an iteration that results in an increment. Each iteration represents a mini-waterfall model.
Spiral Model	This supposes incremental development, using the waterfall model for each step, with more emphasis on managing risk.
Rapid Application Development (RAD) and Time Box Model	It is a software development process that allows usable systems to be built in as little as 60-90 days, often with some compromises.
Agile Methodologies	Agile is an evolutionary approach to software development which is performed in a highly collaborative manner by self-organizing teams with the objective of producing high quality software in a cost effective and timely manner. Some of the different Agile Approaches are Extreme Programming (XP), Scrum, Pair Programming, etc.

Table 1: Classification of Process Models

The characteristics of extreme programming have also been investigated [1] for usage under volatility. The studies thus bring out the benefits and problems associated with using these process models, but they do not address the organizational practice related to their use, suitability, and applicability under requirements volatility. Given the importance ascribed to adoption of contextual change management approaches depending upon the project setting [5], the scenario assumes a greater significance, which our study explores.

In this study, we investigate the prevalent project execution and management approaches adopted in projects under requirements volatility.

The overall study is a part of a comprehensive research endeavor to explore the dynamics that underlie software requirements volatility. Parameters representing the cause and effects of some of the preferred management approaches identified from this study will be fed into software project dynamics model. The project dynamics model represents the project development steps and associated managerial decisions in a set of feedback loops in order to comprehend the process, and devise project management strategies. We intend to discover how some management decisions can influence volatility and subsequently impact project outcome.

This paper is organized as follows. In section 2 we first outline the methodology underlying the empirical investigation. Following this, we present the list of hypotheses identified at the end of the first exploratory phase of our research, and provide descriptions of the measurement constructs and the sample. Section 3 presents the results where we first summarize the findings of the first phase of the study, and then detail on the phase two findings. Observations based on the study results are then enumerated in section 4. Finally section 5 summarizes the key findings.

2. Methodology

2.1. Research Approach

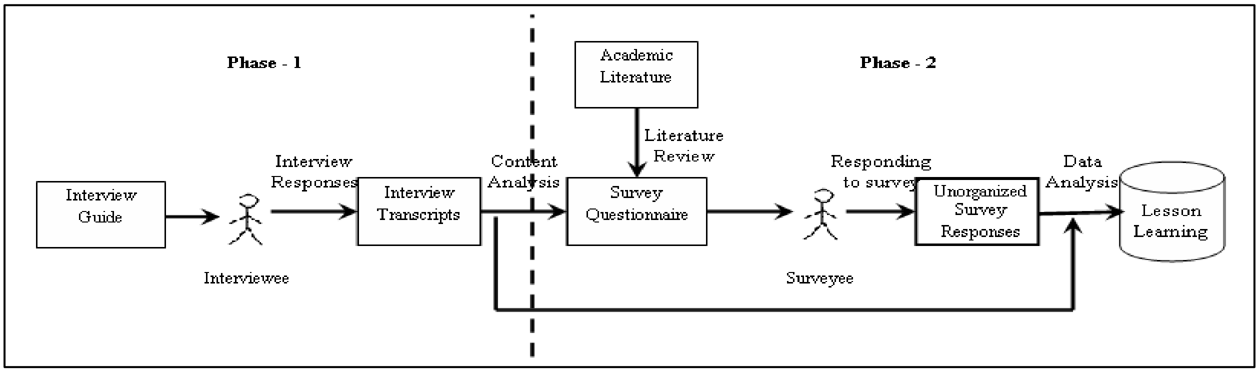

Figure 1: Research Approach

The research presented in this paper was carried out in two phases. An exploratory research design was employed in the first phase in which requirements volatility and its association with project attributes and management techniques were examined in depth. Senior project managers associated with software development were interviewed to gather insights into the problem. A content analysis of the interview data led to development of pertinent hypotheses for subsequent validation. In contrast, the second phase was confirmatory in its approach. It started with the list of hypotheses identified at the end of the first phase; subsequently, a web-based survey was developed to test the hypotheses in an attempt to generalize some of the findings.

This multi-methodology approach has several advantages as outlined in [11, 16]. Combining methods provides a richer contextual basis for interpreting and validating results, and also strengthens the robustness of the findings. The overall organization of the study is presented in figure 1. As depicted, interviewee responses and literature evidence were combined to develop the survey instrument for investigation. Subsequent analysis led to the revelation of emergent patterns.

2.1.1. Phase – 1: Interviews

In the first phase of the study – extending from November 2007 to March 2008 – in-depth qualitative interviews were carried out with eleven senior software project managers of various organizations within Germany. An interview guide, pre-tested on Maykut's [17] guidelines, had been used for the basis of enquiry, and it contained questions on the interviewees' demographics, the organizational settings, project background information, systems development life cycle methodologies, software project success, the awareness and criticality of requirements volatility, requirement change pattern, and the degree and effect of requirements volatility on the project. Interviews were carried out by phone and they normally lasted between one and one-and-a-half hours.

Interview notes were analyzed by means of the constant comparative method [7]. The responses were codified, the codes being generated from the data, rather than predetermined. Each code representing a theme or idea with which each part of the data was associated. New themes were assigned to data that fell outside the possible alternatives. The codes that had common elements were subsequently merged to form categories. The categories derived in this manner were then clustered for pattern identification.

2.1.2. Phase – 2: Survey

A web-based survey instrument validated against Straub's [26] guidelines was utilized in phase two to test the phase one observations on a much wider sample. Pre-testing was utilized to improve the reliability of our questionnaire. The survey contained five sections (nature of question items, and number of questions per section are indicated within brackets) informing the respondents of the purpose of the study, requesting demographic information (like name, age, gender, nationality, size and business focus of respondents' organization (7)), questioning their association with software projects (like years of experience, predominant project execution strategies adopted over the last five years and their flexibility and suitability in incorporating changing requirements (14)) and their take on requirements volatility (like percentage of projects that were considered endangered because of requirement volatility, risk perception of requirement volatility on Likert Type scale, whether the respondents attempted to measure or manage volatility (5)). One section was devoted to details of a recently completed software project (like project size, duration, maturity level, type of application developed, respondents' assigned role, adopted upfront project execution strategy, specific requirement volatility management approaches used and whether these were organization specific, and the final project outcome (18)) that had experienced problems with requirements volatility This method would ensure that the survey results were reliable and not merely software practitioners' opinions and generalizations.

Data analysis was carried out using the SPSS 15.0 statistical package [14]. The basic assumptions required for the different tests were checked in advance. Where the sample size was inadequate to carry out statistical analysis, results provided for further analysis were based on observed frequencies.

2.2. Research Hypotheses

Academic literature on software project risks has emphasized the growing risk posed by requirements volatility [31]. In their study on consolidating project risk factors, Schmidt et al. [23] highlight the time-variant evolutionary nature of software risks. With so many different approaches being referred to in the literature for managing requirements volatility, we wished to ascertain its importance in the present settings. In the first phase, a few of the interviewees preferred to treat volatility more as a "*way of life*" rather than as a risk. However, in respect of the majority of interviewees, we proposed that:

H1: *Project managers' still perceive requirements volatility as a significant risk affecting software projects*

Flexibility to changes emerged as a desirable attribute of a project execution strategy that project managers interviewed during first phase preferred under a volatile environment. Individuals might also have different views of the execution strategy suitability to handle requirements volatility. Nevertheless, a pattern of agreement related to the perception of flexibility and suitability of the project execution strategy choices was evident during the interviews (choices referring to the different process models that are available). This led us to the following general hypothesis, which we expected to be true:

H2: *There is a positive correlation between project execution strategy choices that are considered flexible and choices that are regarded as suitable under requirements volatility*

The omission of execution strategy steps has been cited as one of the factors leading to project failure [25]. Some projects may not have an upfront strategy, or the execution strategy may be unsuitable considering the project characteristics. Our next hypothesis intended to test the significance of the above circumstances on projects found at risk due to requirements volatility. At a later stage, we intend to carry out sensitivity analysis with data from a few such projects, using inappropriate execution strategies, to derive performance improvement recommendations.

H3: *Over 50% of the projects that are at risk due to requirements volatility either lack an upfront execution strategy, or the execution strategy is inappropriate with regard to the project*

Various software organizations employ maturity models to develop and refine their software development process. A model like the SW-CMM (Software Capability Maturity Model) [24] defines five stages of maturity (i.e. initial, repeatable, defined, managed, and optimizing), with each stage consisting of a number of "Key Process Areas" (KPA's), which indicate the essential need that has to be satisfied as the maturity level increases. Level two (repeatable) is related to organizations which are capable of repeating successful projects of the same type depending on individual competencies. At this level, "requirements management" has been identified as a KPA that needs to be satisfied. Even though requirements management as an activity does not include all the processes required for managing volatility, it has been identified as closely associated with change management practices [20]. Hence, we argued that:

H4: *Organizations with a process maturity rating of two or higher have a plan for managing requirements volatility*

2.3. Measurement of Constructs

The constructs of this research include process maturity, project execution strategy, and requirements volatility.

The first construct classifies the process maturity of an organization. As defined in [4], we chose the CMMi-DEV (Capability Maturity Model Integrated for Development) process maturity framework with its five levels (i.e. initial, repeatable, defined, managed, and optimizing) for our study.

Project execution strategy has been used here in a narrow sense to refer to the process model choice decisions. Here we ascertain the usage frequencies of the various process models [29] and their ranking on flexibility and suitability dimensions. A classification of the process models is provided in table 1. The preferred five point Likert scale [8] was used to measure both the flexibility and suitability dimensions of these, with five indicating “very highly” flexible or suitable.

Several studies on software project risk have stressed the importance and criticality of requirements volatility as a risk item [23, 31]. These studies rate requirements volatility relative to other risk items on Likert-type scales. Here, we adopt a similar approach, and using a five point scale (five: very high risk) we measure the degree of risk that the respondents attribute to the problem of requirements volatility affecting software projects.

2.4. Sample Description

In order to minimize guessing responses, the survey targeted experienced software professionals with at least five years of expertise, or who had worked as project leaders, managers or in equivalent positions. A simple random sampling strategy was adopted and invitations to participate in the survey were mailed to members of online mailing lists such as PMNet, IEEE Computer Chapter, SEWorld, and members of the networking portals Xing, and LinkedIn. Follow-up invitations were emailed twice in intervals of two weeks. The survey was made available online for two months. An access counter indicated that a total of 176 respondents visited the survey page, out of which 112 (64%) individuals finally completed it. Some of the completed questionnaires were out-of-sample responses and problem responses (such as multiple responses) and therefore had to be discarded. The final sample size was 82 (47%). Factors such as the number of questions (44), depth of information sought, unfamiliarity of the area, or amount of time required, might have led to the survey’s low completion rate. Most of the respondents were male (73%). 51% of the respondents had over ten years of software project experience. 62% of the respondents had been involved in more than ten projects. Information Technology Services were the most represented (35%) industry type, followed by software (14%), consulting / engineering services (13%) and banking (12%). 62% of the organizations represented by the respondents were large ones with more than 1000 employees.

The survey respondents reported being involved in 1470 projects since 2003. More than half of these projects (54.9%) were considered at risk due to requirements volatility. 19.9% of these 1470 projects did not use any upfront execution strategy. In this respect detailed information on 653 projects were available out of which 39.9% used the waterfall life cycle model while 22.8% were based on iterative-incremental life cycle models. Agile methodologies were used in 9.9% of the projects.

The respondents were also asked to report on a recently completed project that was considered at risk due to requirements volatility. Data were obtained on 42 such projects. 32.6% of these projects represented organizations that did not use any of the maturity ratings. Of the others, level five organizations were found to be the most represented (32.6%), followed by level three ones (11.6%). The respondents reported being mostly associated with MIS applications (41.9%) consisting of software developed for internal usage. Commercial applications (those developed for sale to the general public) and systems applications (those which govern the functioning of physical devices) were equally represented in the sample (23.3%).

3. Results

3.1 Summary of Phase -1 Findings

Here we briefly touch upon the findings that emerged during phase one of our study consisting of interviews. The details of these results are given in [30]. Phase one intended to capture individual experiences, opinions, perceptions and knowledge regarding the problem of requirement volatility affecting software projects. The followings are the key findings that emerged out of the qualitative analysis.

1. The interviewees were appreciative of the problem of requirement volatility affecting software projects. This indicates that large reported instances of project failure under requirement volatility are not fallout of a lack of awareness, but could be because of management ineffectiveness in handling the scenario.

2. Some level of acceptability of the threats associated with requirement volatility was echoed. Instead of viewing it as a significant threat, suggestions were made to accept it "*as part of life*" and adapt.
3. Among the process executions strategies, waterfall model emerged to be the most used despite being acknowledged as "*the least suited*" under requirement volatility. In terms of the desired characteristics of a process execution strategy to be considered acceptable in a volatile environment, the interviewees stressed on the following:
   - Flexibility to accommodate mid-stream requirement changes
   - Should prescribe clear and crisp processes
   - Flexibility to maintain traceability across phases
4. Nine different approaches to managing requirement volatility were pointed out (elaborated later). However the deployment of these approaches under some situation was found to be rather ad-hoc, indicating deficiencies in this regard.

3.2. Phase-2 Findings

Here we address the findings that emerged based on the analysis of the survey data

Usage of Project Execution Strategy under Requirements Volatility

#	Influencing Factors	Freq (%)
1	Influence of business / customers	26.2
2	Complexity of the project	21.4
3	Management preferences	19.0
4	Requirements set of the project	11.9
5	Maturity of the development process	11.9
6	Project initial estimates (size, effort)	4.8
7	Project ultimate objectives	4.8

Table 2: Factors Influencing Project Execution Strategy Choices

11.9% of the 42 projects endangered because of requirements volatility were found not to use any of the execution strategy choices. Of the rest, the iterative-incremental model was found to be the most used (33.3%). The waterfall model was utilized in 23.8% of the projects. Other process models, such as the v-shaped model, prototyping model, rapid application development, and agile methodologies, were found to be equally represented in the sample. The selection of these project execution strategy choices was found to be affected by the set of factors listed in table 2 (the last column indicates how many times each factor was cited as influencing execution strategy choice for endangered projects)

The top two factors in this case were found to be "influence of business/customers" and "project complexity". The relative importance of these factors however differed in a non-volatile environment (not shown). In the later case, except the top most factor (table 2), the next two determinants emerged to be initial project estimates and the process maturity.

Proceeding further, we then tried to assign to each of the execution strategy choices, the predominant influential factors from the list given in table 2. Because of sample size limitations, the analysis was carried out using the technique given in [2] which performs grouping depending upon the observed frequencies. Findings based on the 42 survey responses revealed waterfall model to be mostly driven by "management preferences." In addition, "influence of business" and "development process maturity" also influenced its selection. With the exception of the last two factors in table 2, all others were found to influence the selection of the iterative-incremental model, the top two being "complexity of the project" and "requirements set of the project." "Complexity of the project" was found to influence the non-adoption of agile methodologies.

Specific approaches adopted in projects under requirements volatility

#	List of Approaches	Freq (%)
1	Involving the business side in the project	11.3
2	Project scope negotiation	9.8
3	Rescheduling project deadlines	9.0
4	Engaging in requirements management activities	8.3
5	Documentation of processes, procedures, and activities	6.8
6	Adjusting project human resources	6.4
7	Using expert knowledge	5.6
8	Focusing on communications	5.3
9	Reducing project complexity	4.1
10	Readjusting project effort	8.6
11	Variable costing of additional requirements	4.1
12	Architecting product to withstand change	3.4
13	Training workforce	2.3

Table 3: Management approaches under requirements volatility

The first phase of our study brought out nine different approaches (1-9 in the table 3) in managing projects under requirements volatility. Based on available literature [10], four additional approaches were added (10-13). The list of approaches is given in table 3 (percentage values refer to the degree of usage of these approaches under requirement volatility).

Approaches 4, 5, 8, 12 and 13 (table 3) are more "proactive" by nature, which implies that the usage of these approaches is generally independent of the ongoing project status. Among the approaches listed in the table, the top two entries were found to be the most used, the first one, "involving the business side", being regarded highly important irrespective of the project characteristics.

We then attempted to classify the different approaches depending upon the execution strategy choice adopted in a project. The categorization scheme mentioned in [2] was also used here as the number of responses (42) was inadequate for statistical analysis. Extensive usage of "reducing project complexity" and "adjusting project human resource" approaches under volatility could be noted when using the waterfall model. The two recurrent management approaches to projects utilizing the RAD model were "using expert knowledge" and "variable costing of additional requirements." Iterative-incremental projects employed the "requirements management techniques" more often. As expected, agile projects tended to focus on "communications." However, "documentation of processes, procedures, and activities" were also reported. The usage of other approaches listed in table 3 was invariant across the choice categories, with approaches 1, 2, 3, and 10 (table 3) found to be used more frequently. A cross tabulation of the results with respect to the execution strategy and organization process maturity indicated increasing usage of "proactive" strategies at higher maturity levels (not shown).

Validation of Hypotheses

H1: *Project managers' still perceive requirements volatility as a significant risk affecting software projects*

The survey respondents were asked whether requirements volatility was still perceived as a significant threat to software projects despite all methodological advancements. We measured risk perception on the five-point scale with the descriptor "significant" to imply a rating of four or five. Out of 82 survey responses 72% classified the risk of requirements volatility to be in these two levels, the mean rating being 3.95. A one-sample t-test was carried out to test the difference between the mean rating obtained and the "significant" descriptor level, and the findings did not emerge to be statistically significant (null hypothesis: there is no significant difference between the mean rating, and the minimum significant level (four), p-value: 0.589, α: 0.05). Thus we failed to reject the null hypothesis and conclude that requirements volatility is still perceived as a significant software project risk.

H2: *There is a positive correlation between project execution strategy choices that are considered flexible and choices that are regarded as suitable under requirements volatility*

We wanted to ascertain if the project execution strategy choices that the respondents regarded as flexible are also considered suitable under a volatile development environment. The survey respondents' mean flexibility and suitability ratings of each of the choices by means of individual 5-point scales (5: highest rating of either characteristic) are presented in table 4 (N: 58):

	Mean Rank (1-5 Scale)	
Execution Strategy Choice	**Flexibility**	**Suitability**
Waterfall Model	2.10	2.21
V-Shaped Model	2.64	2.64
Prototyping Model	3.46	3.58
Iterative-Incremental Model	3.51	3.02
Spiral Model	3.08	2.46
RAD and Time Box Model	3.27	3.00
Agile Methodologies	4.10	3.67

Table 4: Mean flexibility and suitability rating of each strategy choice

The results of paired t-tests to establish the difference between the mean ratings of each of the strategy choice, along with the correlation coefficients are provided in table 5 (null hypothesis: there is no significant difference between the ratings, alpha = .05).

With the exception of the iterative-incremental model, the mean flexibility and suitability ratings of the other models were found to be related, providing enough evidence to validate the hypothesis.

Table 4 also shows that the respondents found the agile methodologies to be most flexible in terms of accommodating mid-stream requirements changes. They were followed by the iterative-incremental and the prototyping models. The waterfall model was considered to be the least flexible of all.

H3: *Over 50% of the projects that are at risk due to requirements volatility either lack an upfront execution strategy, or the execution strategy is inappropriate with regard to the project*

We tested the above hypothesis by using the binomial test available in SPSS. 11.9% of the projects at risk (N: 42) did not use any execution strategy, and 40.4% of the remaining were unsure about the appropriateness of the strategy with regard to their project environment. Result of the 1-tailed test failed to prove statistical significance of the combined proportion at alpha = .05 (p value: 0.439).

Execution Strategy Choice	Null Hypothesis	P-value	Correlation Coefficient	Comment
Waterfall Model	True	0.410	0.622 (significant)	The ratings are related
V-Shaped Model	True	1.0	-	The ratings are related
Prototyping Model	True	0.377	0.687 (significant)	The ratings are related
Iterative-Incremental Model	False	0.015	0.165 (low, not significant)	The difference between the two ratings is significant
Spiral Model	False	0.014	0.771 (significant)	P-value of correlation coefficient = 0.002 (Significant at alpha: .05). Hence, the ratings are somewhat related
RAD and Time Box Model	True	0.137	0.536 (significant)	The ratings are related
Agile Methodologies	True	0.056	0.483 (significant)	The ratings are related

Table 5: Statistical test of difference between flexibility and suitability ratings

An investigation of the execution strategy selection factors in situations where the respondents were not convinced of the strategies appropriateness revealed that business's influence was the most dominant factor (29.4%), followed by management preference (23.5%), and development process maturity (17.6%).

H4: *Organizations with a process maturity rating of two or higher have a plan for managing requirements volatility*

The survey results disclosed that 65.9% of the respondent organizations (N: 44) have an organizational plan for handling changing requirements. Further sub-classification identified 25 responses from organizations with a maturity rating of two or higher. 84% of these were found to have frameworks for managing requirements volatility. This number was significant at alpha = .05 (p value: 0.003), with a corresponding effect size of 1.495 (Large) [3]. In the light of the above data, we rejected the null hypothesis and accepted H4.

4. Observations

We can make the following observations out of the study results

1. The findings on the perception of requirement volatility indicated that project managers are becoming more attuned to the fact that requirements are expected to change during the project. They are also very much appreciative of the risks associated with such changes. This is expected to lead to devising of strategies for managing requirement volatility in projects, where the project managers seek to manage the changes instead of trying to prevent them. All this is expected to increase the satisfaction of the project stakeholders and contribute towards more successful endeavours.
2. Decisions regarding selection of execution strategies in projects at risk were found to be largely influenced by business/customers. As the business side may not be aware of the problems posed by requirements volatility, this could be one reason why so many projects face problems or even fails under requirement volatility.
3. Among the execution strategy choices, agile methodologies were found to be the most suitable under requirement volatility. However the usage pattern revealed that the waterfall model was adopted very frequently (second in rank based on the survey findings) in projects endangered because of requirements volatility. Both the interview and the survey results revealed waterfall model to be unsuitable in similar conditions. The selection of the waterfall model was found to be driven mostly by management preferences and business influence. It appears that the available expertise in successfully using the waterfall model, or natural resistance to the adoption of new frameworks could be the reason for the continued use of the waterfall model, which future studies could investigate.
4. Among the specific approaches used for managing projects under requirement volatility, "involving business side" emerged as the most adopted approach. Usage of this approach was also frequent

in projects using the waterfall framework. This suggests a gradual shift towards tailoring of project managing strategies depending upon the project context. Similar evidence of tailoring was the practice of documentation in projects using the agile methodology under requirements volatility. Some of the participant organizations were found to use customized agile frameworks to suit the needs of the involved parties.

5. Results also highlight that some of the approaches were used very frequently irrespective of the execution strategy choices. A reason behind such findings could be that some of the project managers used approaches on a trail-and-error basis. This was pointed out by the interviewees in phase one of the study. A proposed extension of the study could include the identification of the factors that influence the selection of approaches. Investigating this in association with project success or failure could lead to the development of a requirements volatility management maturity model by means of an approach highlighted by Sukhoo et al. [28].

5. Conclusion

Software projects continue to be troubled by requirements volatility. This paper has provided an update of the current scenario concerning the selection and usage of a project execution strategy under requirements volatility, as well as the frequently adopted management approaches for tackling the problem.

Findings indicate a shift in project manager's mindset towards requirement volatility with acceptation of the fact that requirement volatility is not an exception. The project execution strategies and approaches that are likely to be effective in a volatile environment also emerged out. Result advice for a need of caution in decisions regarding the selection of execution strategies in projects. The execution strategy choice was found to be influenced by the business in majority of the projects at risk under requirements volatility. In this regard, suggestions to train project stakeholders were put forward as they may not be aware of the potential hazards associated with volatility, and rather contribute to the cause. Depending on the execution choice decision made, the results are also expected to offer guidance in the selection of approaches, which in some cases was rather ad-hoc under volatility.

The nature of the study did, however, impose limitation on this study. Non response bias was tested by comparing early (those questionnaires received after the first invitation) and late (those questionnaires received after the follow-up invitation) respondents, but no significant difference could be observed. However, the possibility of observation and information bias during the interviews cannot be ruled out completely, due to the involvement of only a single observer. The results could, for example, be biased due to recollection errors. In addition, project-specific responses were not checked against available evidences. Our sample is relatively small and does not provide sufficient coverage of all situations, which limits generalization. However, the focus of our research is more on the comprehension of the scenario. Patterns uncovered in the research are early insights, and are expected to provide the basis for further work in this area.

The importance of our study was stressed by one interviewee as *"the area of study is quite interesting and has provided means of finalizing the points which might hamper a project in its execution and implementation."* Future research could statistically validate the different management approaches' usage patterns based on the process characteristics, and investigate the influence of cultural factors on the overall context. The trend towards adoption of more "proactive" management strategies at higher maturity levels could be investigated further possibility leading to the establishment of a requirements volatility management framework on the dimensions "nature of management approach" and "project characteristics." A case-study-based approach could be employed to provide an understanding of the project execution strategy selection mechanism. The behavioral and social factors that additionally influence the use of these strategy choices under unfavorable situations should also be worth exploring.

6. References

[1] Beck, K., Extreme programming Explained: Embrace Change, Addition-Wesley, MA, 1999.

[2] R. Berntsson-Svensson, and A. Aurum, "Successful Software Project and Products: An Empirical Investigation", ISESE'06, Rio de Janeiro, Brazil, 2006.

[3] Cohen, J., Statistical power analysis for the behavioral sciences (2nd ed.), Lawrence Earlbaum Associates, NJ, 1988.

[4] CMU/SEI-2006-TR-008, "CMMI for Development, Version 1.2 (doc), CMMI-DEV", Carnegie Mellon University Software Engineering Institute, 2006. http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html.

[5] C. Ebert, and J. Man, "Requirements uncertainty: influencing factors and concrete improvements", ICSE, 2005.

[6] Ferreira, S., Measuring the effects of requirements volatility on software development projects, Ph.D Thesis,

Department of Computer Science & Engineering, Arizona State University, Tempe, AZ., 2002.

[7] Glaser, B., and Strauss, A., The Discovery of Grounded Theory: Strategies for Qualitative Research, Aldine Publishing Co., Chicago, IL, 1967.

[8] C. Gwinner, "5-point vs. 6-point Likert Scales", Infosurv White Paper, 2006.

[9] T. Javed, M. Maqsood, and Q. Durrani, "A study to investigate the impact of requirements instability on software defects", ACM Software Engineering Notes, May, 2004.

[10] Jones, C., Estimating Software Costs. McGraw Hill, New York, 1998.

[11] B. Kaplan, and D. Duchon, "Combining Qualitative and Quantitative Approaches in Information Systems Research: A Case Study", MIS Quarterly, 1988.

[12] A. Khan, and S. Balbo, "Agile versus Heavyweight Web Development: an Australian Survey", Department of Information Systems, The University of Melbourne, 2005.

[13] Kotonya, G., and Sommerville, I., Requirements Engineering: Processes and Techniques, John Wiley and Sons Ltd, Chichester, 1998.

[14] Landau, S., and Everitt, B. S., A Handbook of Statistical Analyses Using SPSS, Chapman & Hall/CRC; 1 edition, 2003.

[15] Madachy, R. J., A Software Project Dynamics Model for Process Cost, Schedule, and Risk Assessment, Ph.D. Dissertation, Department of Industrial and Systems Engineering, USC, 1994.

[16] M.L. Markus, "Finding a Happy Medium: Explaining the Negative Effects of Electronic Communication on Social Life at Work", ACM Transactions on Information Systems, 1994.

[17] Maykut, P., Beginning Qualitative Research: A Philosophical and Practical Guide (The Falmer Press Teachers' Library, 6). Routledge, 1 edition, 1994.

[18] C. Neill, and P. Laplante, "Requirements Engineering: The State of the Practice", IEEE, Software, 2003.

[19] N. Nurmuliani, D. Zowghi, and S. Fowell, "Analysis of Requirements Volatility during Software Development Life Cycle", ASWEC'04, Australia, 2004.

[20] D. Pfahl, and K. Lebsanft, "Using Simulation to Analyze the Impact of Software Requirements Volatility on Project Performance", Information and Software Technology, 2000.

[21] V. Rajlich, "Changing the paradigm of software engineering", Communications of the ACM, 2006.

[22] Scacchi, W., Process Models in Software Engineering, Final Version to appear in, J.J. Marciniak (ed.), Encyclopedia of Software Engineering, 2nd Edition, John Wiley and Sons, Inc, New York, 2001.

[23] R. Schmidt, K. Lyytinen, M. Keil, and P. Culle, "Identifying Software Project Risks: An International Delphi Study", Journal of Management Information Systems, 2001.

[24] SEI / Software Engineering Institute, "Capability Maturity Model for Software (CMM)", Carnegie Mellon, 2008. http://www.sei.cmu.edu/cmm/

[25] G. Stepanek, "Software Project Secrets: Why Software Projects Fail (Expert's Voice)", Apress, 2005.

[26] D.W. Straub, "Validating Instruments in MIS Research", MIS Quarterly, 1989.

[27] Sudhakar, M., Managing the Impact of Requirements Volatility, Master Thesis, Department of Computing Science, Umeå University, Umeå, Sweden, 2005.

[28] A. Sukhoo, A. Barnard, M.M. Eloff, and J. A. V. Poll, "An Evolutionary Software Project Management Maturity Model for Mauritius", Interdisciplinary Journal of Information, Knowledge, and Management, 2007.

[29] SUNY, "A Survey of System Development Process Models", Center for Technology in Government, University at Albany, 1998.

[30] R. Thakurta, and F. Ahlemann, "Requirements Volatility Awareness and Management – A Qualitative Study in German Context", CISTM, Delhi, India, 2008.

[31] A. Tiwana, and M. Keil, "The One Minute Risk Assessment Tool", Communications of the ACM, 2004.

[32] TSGI, "CHAOS Chronicles v3.0", The Standish Group International Inc., West Yarmouth, USA, 2003. http://www.standishgroup.com/chaos/toc.php.

[33] Wiegers, K., Software Requirements, Microsoft Press, Redmond, Washington, 1999.

[34] Young, R. R., Effective Requirements Practices, Addison Wesley, 2001.

[35] D. Zowghi, R. Offen, and N. Nurmuliani, "The Impact of Requirements Volatility on the Software Development Lifecycle", IFIP World Computer Congress, China, 2000.

[36] D. Zowghi, and N. Nurmuliani, "A Study on the Impact of Requirements Volatility on Software Project Performance", Proceedings of Ninth Asia Pacific Software Engineering Conference, Queensland, Australia, 2002.