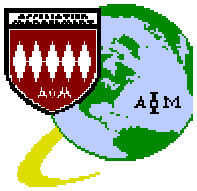

Journal of Information Technology Management

ISSN #1042-1319

A Publication of the Association of Management

MANAGEMENT OF REQUIREMENT VOLATILITY - A STUDY OF ORGANIZATIONAL COMPETENCY AND HOW IT IS INFLUENCED BY THE PROJECT ENVIRONMENT

RAHUL THAKURTA
XAVIER INSTITUTE OF MANAGEMENT BHUBANESWAR
rahul@ximb.ac.in

ABSTRACT

Requirement 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 requirement 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 depending upon the project and process characteristics. The current scenario regarding adoption of different frameworks and tools for managing changing requirements has also been pointed out. The study results are expected to assist project managers in their choices related to project administration under requirement volatility.

Keywords: requirement volatility, project risk, management approach, involving business side

INTRODUCTION

Despite advances in Software Engineering over the past 30 years, most of the software projects still experience numerous requirements changes during their life cycle, brought about by the dynamic nature of the development activities [15]. Requirement volatility, which refers to the change to requirements during the software development life cycle, has been reported as one of the main factors that cause a project to be challenged [26]. Approaches like use of risk driven process models [4], joint application design (JAD), prototyping and configuration management change control board [12], use of requirements baselining [28] have been suggested for managing projects under volatility. Despite advancement in project management tools and methodologies, a Standish Group study [27] found out high instance of project failure and requirement fluctuations, indicating some level of organizational inadequacy in properly managing change. The importance of contextual change management techniques in the present scenario of rapidly evolving business and technological paradigm was highlighted by Ebert and Man [7], but has not been investigated upon.

This study seeks to investigate the organizational perception regarding the problem of requirements changes. We want to find out the level of awareness concerning requirement volatility, the management approaches available to cope with the problem and their suitability under different project environments. Some of the available organizational frameworks to address requirement volatility have been reported. The effect of

requirement volatility on various project activities has also been looked into. We expect our study to contribute to the current knowledge base on managing requirement 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 requirement volatility.

This research project was carried out in two phases. An exploratory research design was employed in the 1st phase where requirement volatility and its association to project attributes and management techniques were examined in depth using interviews. Hypothesis identified at the end of phase-1 were tested using an online survey in the 2nd phase. Findings of the interviews have been reported simultaneously to strengthen our claim and present contrasting viewpoints and arguments.

This paper is organized as follows. The research methodology is discussed next where we elaborate on our research approach comprising of interviews in the 1st phase and survey in the 2nd, list the study hypothesis, provide descriptions of measurement constructs and study sample, and then discuss on validation of our research approach. The results are provided in the next section, and subsequently discussed where interview findings have been used to strengthen the claims made, and also to highlight contrasting viewpoints and arguments. The final section summarizes the key findings and provides direction of further research opportunities.

METHODOLOGY

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 requirement 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 Kaplan and Duchon [13]. 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.

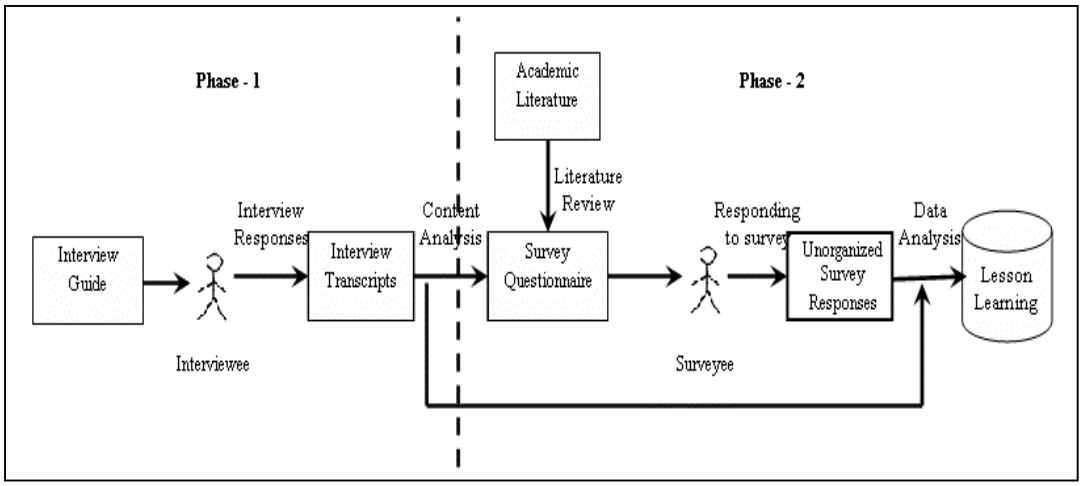

Figure 1: Research Approach

Phase – 1: interviews

In the first phase of the study – extending from November 2007 to March 2008 – in-depth interviews were carried out with eleven senior software project managers belonging to various organizations. An interview guide, pre-tested on Maykut and Morehouse [14] guidelines had been used for the basis of enquiry, and it contained questions on the interviewees' demographics, the organizational settings, project background information, awareness and criticality of requirement volatility, and the degree and effect of requirement volatility on the project. All the interviews were carried out by phone and they normally lasted between one and one-and-a-half hours. The interviews were carried out in a single round. The feedback and experience of an interview were used to subsequently refine the questionnaire and rectify errors in conducting the process. The process was continued as long as there was no further lesson learning from the interviews, and the content provided were repetitions of the earlier interviews. At this point the interviews were stopped and we proceeded into analysis.

Interview notes were analyzed by means of the constant comparative method [8]. The data analysis was inductive as we intended to capture the individual perceptions on the problem. 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. The codes that had common elements were subsequently merged to form categories [22]. The categories derived in this manner were then clustered for pattern identification.

Phase – 2: survey

A web-based survey instrument was utilized in phase-2 to test the phase-1 observations on a much wider sample. 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 requirement 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, specific requirement volatility management approaches used and whether these were organization specific, and the final project outcome (18)) that had experienced problems with requirement volatility This method would ensure that the survey results were reliable and not merely software practitioners' opinions and generalizations.

Statistical data analysis was carried out using the SPSS® 15.0 statistical package. The statistical assumptions required for the different tests were checked in advance. In situations of non-conformation to these assumptions, results provided for further analysis are based on frequency of occurrences matching the test criteria.

Research Hypotheses

Academic literature on software project risks has emphasized the growing risk posed by requirement volatility [26]. In their study on consolidating project risk factors, Schmidt et al. [19] highlight the time-variant evolutionary nature of software risks. With so many different approaches being referred to in the literature for managing requirement 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 requirement volatility as a significant risk affecting software projects*

The high extent of requirements fluctuations [27] could be due to lack of appreciation of threat posed by requirement volatility, or inherent slackness in engaging to requirements change management activities. Our next hypothesis presents the argument:

H2: *The majority (>50%) of project managers are not proactively engaged in requirements change management activities in their projects*

Ebert and Man [7] had emphasized the need of contextual change management techniques catering to the particular project development environment. With the availability of the different software process models [24], and considering every software project to be unique [10] having own set of requirements, project constraints, goals/objectives and stakeholders, we expect situation-specific approaches for dealing with requirement volatility. The following premise tries to confirm the belief:

H3: *Not all the approaches to managing requirement volatility available in practice are used equally*

Various software organizations employ maturity models to develop and refine their software development process. A model like the SW-CMM (Software Capability Maturity Model) [20] 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 [16]. Hence, we argued that:

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

Measurement of Constructs

The constructs of this research include project size, project contract, application categories, process maturity ratings, risk of requirement volatility, and impact on project management activities.

Projects were classified based on Yourdon [29] as small (less than 10 people, duration of 3-6 months), medium (team of 20-30 people, duration of 1-2 years), large (team of 100-300 people, duration of 3-5 years) and mind-boggling (team of 1000-2000 people, duration of 7-10 years). The different types of project contracts which we have used for our research has been elaborated in Table 1.

The different application categories we chose for our research included commercial, MIS, systems, military, contract or outsourced products, described by Jones [11] as follows (Table 2).

Process maturity ratings classify the process maturity of an organization. As defined in CMU [5], 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) as given below (Table 3).

Table 1: Project Contract Types

Contract Type	Description
Fixed Price (or 'Firm Fixed Price')	The owner (business) specifies the work and the contractor (software development organization) gives a price. In this case the contractor assumes almost all of the risk and as a result reaps whatever profit there is.
Time and Materials (T&M)	Simple billing at pre-negotiated rates for labor and materials on a project. Some "Fixed Price" contracts specify this as a method for determining costs of change orders.
Cost Plus Fixed Fee (Also called 'Cost Plus')	This type of contract shifts most of the risk to the owner (business), but also allows the owner a high degree of flexibility. The contractor (software development organization) has profit at risk and will seek to minimize cost/duration to return a higher proportional profit margin.
Cost Plus Percentage of Costs (CPPC)	This is very similar to the 'Cost Plus Fixed Fee' contract except that the contractor (software development organization) bears even less risk. Their fee is calculated based on a percentage of actual costs.

Table 2: Application Categories

Application Category	Description
Commercial Software	This refers to software that has been designed and developed for sale to the general public.
MIS Software	This refers to the normal business software used by organizations for internal operations.
System Software	This refers to code that controls physical devices such as computers or telecommunication systems and also includes operating systems, databases and middlewares
Military Software	This constitutes all project which are constrained to follow various military standards
Contract or Outsourced	This refers to Integration products purchased from 3rd party

Table 3: Process Maturity Ratings

Maturity Level (L)	Description
L-1 (Initial)	The software process within the organization is ad-hoc and chaotic with ineffective management procedures and project plans
L-2 (Repeatable)	The organization can successfully repeat projects of the same type. However projects success depends more on individual managers and organization folklore acting as a process description.
L-3 (Defined)	The organization has a defined process with formal procedures to ensure the application to all software projects
L-4 (Managed)	In addition to the above activity, the organization has a formal program to collect quantitative process and product metrics and analyze and use these for process improvement activities
L-5 (Optimizing)	In addition to the above activity, the organization demonstrates its commitment to continuous process improvement.

Several studies on software project risk have stressed the importance and criticality of requirement volatility as a risk item (Schmidt et al. [19]; Tiwana and Keil [26]). These studies rate requirement volatility relative to other risk items on Likert-type scales [9]. Here, we adopt a similar approach, and using a 5-point scale (5: very high risk) we measure the degree of risk that the respondents attribute to the problem of requirement volatility affecting software projects.

Finally, the impact on project management activities were also measured using a 5-point scale, the different management activities corresponded to eight knowledge areas highlighted in PMBoK [17].

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 requirement volatility. 80% of these 1470 projects were found to use any of the available process models. Of these, use of waterfall model was reported to be highest (39.9%), followed by the iterative-incremental model (22.8%). Agile methodologies were used in 9.9% of the projects. Also a high percentage (68.4%) of the 1470 projects was reported to be successful. However the data did not indicate of the remaining projects that failed, how many failed primarily because of requirement volatility.

The respondents were also asked to report on a recently completed project that was considered at risk due to requirement 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-5 organizations were found to be the most represented (32.6%), followed by level-3 ones (11.6%). The respondents reported being mostly associated with MIS applications (41.9%). Commercial applications and systems applications were equally represented in the sample (23.3%). With regard to project contracts, 44% of the projects used the fixed price contract. Time and material contract was used in 40%, and Cost plus contract in 12% projects. Surprisingly use of cost plus percentage of costs contract was not reported in the sample. Finally, with regard to project outcome, 36.4% of these 42 projects were considered to be successful. 54.5% of projects were regarded as partial failures (defined by the interviewees as a situation where the vendor lost money), the rest (9.1%) failed completely.

Validation of Research Methods

The survey instrument was validated using Straub's [21] guidelines. Pre-testing was utilized to improve the reliability of our questionnaire. Three of our interviewees also completed the survey questionnaire. Comparison of their data enabled to evaluate the construct validity of the questionnaire. The interview results also helped us to form survey questions and interpret the answers of the respondents.

Our sample represented a broad range of IS project types across a variety of industries and spanning across small to large organizations. The extensive representation of projects and organizations should reduce concerns of bias in the sample. Non response bias which poses a serious threat to the validity of the results was tested by comparing early (those received on first invitation) and late (those received after the 2nd follow up) respondents [2]. Results revealed absence of any significant difference along key sample and project characteristics (α=0.05)

However, the results of the research should be interpreted with some caution. There are chances of observation and information bias during the interviews due to the involvement of a single observer. The results may be biased because of recollection error as some respondents reported association and details of projects executed long back. Project specific responses were also not validated against available data or views of co-project members.

Our sample is relatively small and may not be representative of all development projects and organizations. Thus the survey and interview do not provide sufficient coverage of all situations. Some of the findings were also not statistically verified because of insufficient data points. However the focus of our work is more of understanding of the phenomena as experienced in organizations, and the resultant impact on project success. Patterns uncovered in the research are early insights, and is expected to provide basis for further work in this area.

RESULTS

Summary of Phase – 1 Findings

Here we briefly touch upon the findings that emerged during phase-1 of our study consisting of interviews. The details of these results are given in Thakurta and Ahlemann [25]. Phase-1 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 [27] 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. Six interviewees mentioned methods of measuring requirement volatility through the following three metrics:
    - Periodic count of requirement changes – number of additions, deletions, and modifications (of individual requirements or use cases) during a specified time interval
    - Extra effort expended - amount of effort the project had to spent on the modified requirements
    - Requirement change during the development lifecycle – cumulative change to the requirement finalized at the requirement specification stage
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.

Phase-2 Findings

Here we discuss validation of our research hypotheses based on findings that emerged out of analysis of the survey data.

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

The survey respondents were asked whether requirement volatility was still perceived as a significant threat to software projects despite all methodological advancements. We measured risk perception on the 5-point scale with the descriptor "significant" to imply a rating of four or five. Out of 82 survey responses 72% classified the risk of requirement 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 requirement volatility is still perceived as a significant software project risk.

We also enquired about the extent to which measurement of requirement volatility was practiced. Just over half of the respondents (55%) attempted to measure requirement volatility. Corresponding to the instances where requirement volatility was measured, the results also brought out the relative usage of the identified metrics (see above) as given below:

- Periodic count of requirement changes (38%)
- Extra effort expended (31%)
- Requirement change during the development lifecycle (19%)
- Others (12%)

We asked the survey respondents on their perception of the extent to which the different activity sets corresponding to each of the identified knowledge areas [17] gets affected by requirement volatility. The objective was to get sense of which of these activity sets had to be managed more in order to prevent projects failure under requirement volatility. Figure 2 shows the extent of effect on a 5-point scale where five indicate the highest level. The result was also found to be invariant across the industry types identified above.

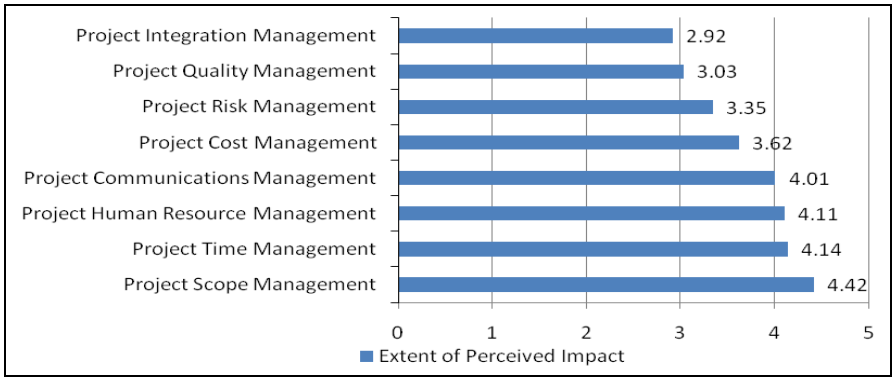

Figure 2: Extent of Impact on Knowledge Areas

As expected, the responding project managers saw scope management activities to be affected the most. Scope, time, and cost also form the "iron triangle", whose effective management influences the quality of the work [1]. The effect on time and cost management activities was also perceived to be on the higher side of the measurement scale (i.e. 4.14 and 3.62 respectively on the 5-point scale). The degree of perceived effect on project quality management activities was found to be comparatively low. In large scale projects the tasks associated with project quality management were somewhat abandoned in face of changing requirements. This thereby received considerably lower attention as compared to some of the other knowledge areas as shown in Figure 2. The project managers faced problems deciding on the project team size and also acquiring the desired capabilities as more and more changes were requested from the business side. Hence the perceived impact on the human resource management activities emerged as "high" (rating of 4.11 on the 5-point scale). Similar perception could also be noted for the project's communications management activities which were regarded as "*utmost important*" in the context of achieving the project's intended outcomes.

H2: *The majority (>50%) of project managers are not proactively engaged in requirements change management activities in their projects*

A high proportion (93.8%) of project managers were found to be proactively involved in requirement change management activities in their projects. In order to find out if this figure was "significant", we used the Binomial (1-tailed) test. Results based on 42 responses yielded the proportion to be statistically significant. Based on the findings we rejected the null hypothesis and

claimed that a majority of project managers indeed attempted to manage requirement volatility in their projects.

Some slackness in managing requirement volatility could additionally be noticed. Among the respondents who considered requirement volatility to be a significant threat (i.e. providing rating of four or five on the 5-point risk perception scale), only 87% were found to be engaged in managing requirement volatility.

H3: *Not all the approaches to managing requirement volatility available in practice are used equally*

The first phase of our study brought out nine different approaches (1-9 in the Table 4) in managing projects under requirement volatility. Based on available literature [12], four additional approaches were added (10-13). The list of approaches is given in Table 4 (percentage values refer to the degree of usage of these approaches under requirement volatility).

Approaches 4, 5, 8, 12 and 13 (Table 4) 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.

Table 4: Management Approaches under Requirement 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

We then attempted to classify the different approaches depending upon the project characteristics like project size, application category, project contract, etc. The categorization scheme given in Berntsson-Svensson and Aurum [3] was also used here as the number of responses (42) was inadequate for statistical analysis. Results indicated difference in usage of the approaches depending upon the size of the project. For both small and medium scale projects, the approach of "rescheduling project deadline" was highly employed. "Adjusting project human resources" was found to be mostly used in medium scale projects. Large scale projects resorted more to "project scope negotiation" and use of "expert knowledge". A similar categorization based on application category however failed to yield significant differences.

Selection of some approaches was governed by more than one criterion. Like use of "readjusting project effort" was prevalent among small and medium scale fixed price projects. Large scale projects using time-and-material contract mostly engaged in "requirements management activities".

No noticeable difference in pattern of usage of approaches could be observed among the different application categories. However "focusing on communications" was more for the commercial projects, as compared to the rest. A similar attempt to classify the approaches based on process maturity was also not fruitful.

The usage of other approaches listed in Table 4 was found to be invariant across the choice categories. Now in order to validate our stated hypothesis regarding usage of approaches, we used the confidence interval approach. The confidence interval (CI) of the difference between the proportion of the most used approach (i.e. involving the business side in the project) and the least used approach (i.e. training workforce) came out as -0.10 to 0.16. Since 0 is within the CI, hence the difference did not emerge to be statistically significant. However, the effect size of the difference came out as 0.383 (medium) [6]. Hence, based on the above evidences we could claim that not all the approaches to managing projects under requirement volatility are used equally in practice.

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

The survey results disclosed that 65.9% of the respondent organizations (N: 44) have an organizational plan for handling changing requirements. In this regard, prominent frameworks for managing requirement volatility included formation of "change control boards" and equivalent in finalizing project requirements, usage of "central requirement management (CRM)" system for

tracking requirements, and employing tools like "rational rose", "influx", "stats", "JIRA" and "Mercury" for change administration.

A further sub-classification (i.e. of the organizations having plans for handling changing requirements) identified 25 responses from organizations with a maturity rating of two or higher. 84% of these were found to have frameworks for managing requirement volatility. This number was significant at alpha = .05 (p value: 0.003), with a corresponding effect size of 1.495 (large) [6]. In the light of the above data, we rejected the null hypothesis and accepted H4.

OBSERVATIONS

Our study has captured the project manager's perception of requirement volatility and how they respond when projects become endangered by this. Although requirement volatility emerged as a significant threat affecting software projects, some preferred to take it as a "*way of life*" as one interviewee remarked: "*Isn't it time we accepted requirements change as a part of our daily life and adapt accordingly?*" This suggests that avoiding requirement volatility may not be completely possible in the current project development scenario. Rather one has to accept it and plan for the changes in order to ensure project success. However, despite general awareness there was some lack of understanding of requirement volatility. Requirement volatility was mostly viewed in terms of discrete changes occurring at different points of time during project development and not in terms of the pattern of behavior. This myopic view of a project is likely to lead to ad-hoc solutions inappropriate for the problem. This in turn could also increase chances of project failure.

Placed high on the process maturity ladder also did not mean a respite from the problems arising because of requirement volatility. Projects belonging to level-5 organizations' were found to be most represented among the endangered sample.

We investigated the impact of requirement volatility on the different knowledge areas identified in PMBoK. Among them, the impact of scope management, time management, and human resource management was found to be more. Other than these identified knowledge areas, the impact of requirement volatility was observed on sales and marketing activities, re-engineering activities and knowledge management activities. On one occasion, the "s*ystem architecture was totally negated by major requirements change*". All these impacted the "*corporate culture*" in some occasions. The observations highlight the adverse possibilities that requirement volatility can bring about in a project. Some slackness in managing projects under requirement volatility could be noted based on the survey results. The fact was strengthened by interview evidences as one of the interviewees remarked *"I know about volatility but I have never dealt with it in our projects"*. Based on the results it is therefore suggested that the project managers should plan their strategies upfront so that when the changes happen, they are in a better position to handle them and in the way ensure smooth administration of the project.

Different approaches for managing projects under requirement volatility could be identified. Some variation of usage of the approaches could be noted based on project characteristics. However irrespective of such project or process characteristics, the practice of "involving the business side" emerged as the most frequently used approach for managing projects under requirement volatility. Interestingly the sample of projects revealed waterfall process model to be the most used. Factors outside perceived project management control could be instrumental behind such selection as pointed out by one: "*It is not us that decide the life cycle model for our project. Our client has a specific process model and we have to follow the model even if it's not appropriate for the project*". Now as the business side may not be aware of the problems posed by requirement volatility, this could be one reason why so many projects face problems or even fails under requirement volatility.

Results also highlight that some of the approaches were used very frequently irrespective of the project characteristics. 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-1 of the study. A proposed extension of the study could include the identification of the factors that influence the selection of these approaches. Investigating this in association with project success or failure could lead to the development of a requirement volatility management maturity model by means of an approach highlighted in Sukhoo et al. [23].

CONCLUSION

Software projects continue to be troubled because of requirement volatility. This paper presented the perception of software project managers regarding the problem of requirement volatility, different approaches used for managing projects under requirement volatility, and how they are affected by the process and process characteristics. As evident in the literature, the appreciation of the threat posed by requirement volatility emerged as significant among the study respondents. Schedule, effort and development activities were found to be most

affected because of volatility. Different approaches to managing projects under requirement volatility could be identified of which the practice of involving the business side was found to be the most used. Frameworks like change control board was found to be used frequently in finalizing the requirements. Adoption of 3rd party tools was also reported. The results also illustrate some of the inadequacies of project management practices under volatility. A careful consideration of the results is expected to assist relatively inexperienced project managers to survive in similar situations and in the process contribute to more successful endeavours.

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.*" Other than addressing the study limitations, follow up work could look to statistically validate the different management approaches' usage patterns based on the process characteristics, and investigate the influence of cultural factors on the overall context. Additional research can also explore the reason behind large usage of waterfall models despite obvious limitations under requirement volatility [18]. Investigations into requirement volatility by taking a well rounded perspective and considering both the software project organization and the business users is also expected to leverage our overall understanding of the problem.

REFERENCES

[1] Ambler, S.W. "The 'Broken Iron Triangle' Software Development Anti-pattern", **http://www.ambysoft.com/essays/brokenTriangle.html** , October 2008.

[2] Armstrong, J. S. and Overton, T. S. "Estimating Non response Bias in Mail Surveys," *Journal of Marketing Research*, Volume 14, Number 3, 1977, pp. 397-402.

[3] Berntsson-Svensson, R. and Aurum, A. "Successful Software Project and Products: An Empirical Investigation," *Proceedings of the ISESE'06*, Rio de Janeiro, Brazil, September 21–22, 2006, pp. 144-153.

[4] Boehm, B.W. "Software Risk Management: Principles and Practices," *IEEE Software*, Volume 8, Number 1, 1991, pp. 32–41.

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

[6] Cohen, J., *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.), Lawrence Earlbaum Associates, Hillsdale, NJ, 1988.

[7] Ebert, C. and Man, J. "Requirements uncertainty: influencing factors and concrete improvements," *Proceedings of the International Conference on Software Engineering*, 2005, pp. 553-560.

[8] Glaser, B. and Strauss, A., *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Aldine Publishing Company, New York, 1967.

[9] Gwinner, C. "5-point vs. 6-point Likert Scales," Infosurv White Paper, **http://www.inforsurv.com**, June 2006.

[10] Hausen, H. L. "Rule-Based Handling of Software Quality and Productivity Models," In: B. Littlewood and N. Fenton (ed.) *Software Reliability and Metrics*. Elsevier Science Publishers Ltd, London, April 1991.

[11] Jones, C. "Software Sizing," *IEEE Review*, Volume 45, Number 4, 1999, pp. 165-167.

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

[13] Kaplan, B. and Duchon, D. "Combining Qualitative and Quantitative Approaches in Information Systems Research: A Case Study," *MIS Quarterly*, Volume 12, Number 4, 1988, pp. 571-586.

[14] Maykut, P. and Morehouse, R., *Beginning Qualitative Research: A Philosophical and Practical Guide*, Routledger Farmer, London, 1994.

[15] Nurmuliani, N., Zowghi, D. and Fowell, S. "Analysis of Requirements Volatility during Software Development Life Cycle," *Proceedings of the 2004 Australian Software Engineering Conference*, Melbourne, Australia, April 13-16, 2004.

[16] Pfahl, D. and Lebsanft, K. "Using Simulation to Analyze the Impact of Software Requirements Volatility on Project Performance," *Information and Software Technology*, Volume 42, Number 14, 2000, pp. 1001–1008.

[17] PMBoK., *A Guide to the Project Management Body of Knowledge*, Newtown Square, Project Management Institute, Inc., PA, 2004.

[18] Rajlich, V. "Changing the paradigm of software engineering," *Communications of the ACM*, Volume 49, Number 8, 2006, pp. 67-70.

[19] Schmidt, R., Lyytinen, K., Keil, M. and Culle, P. "Identifying Software Project Risks: An International Delphi Study," *Journal of Management Information Systems*, Volume 17, Number 4, 2001, pp. 5-36.

[20] SEI / Software Engineering Institute. "Capability Maturity Model for Software (CMM)," Carnegie Mellon, http://www.sei.cmu.edu/cmm/, May 2008.
[21] Straub, D. W. "Validating Instruments in MIS Research," *MIS Quarterly*, Volume 13, Number 2, 1989, pp. 147-169.
[22] Strauss, A. and Corbin, J., *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*, Sage Publications, London, 1990.
[23] Sukhoo, A., Barnard, A., Eloff, M.M. and Poll, J. A. V. "An Evolutionary Software Project Management Maturity Model for Mauritius," *Interdisciplinary Journal of Information, Knowledge, and Management*, Volume 2, 2006, pp. 671-690.
[24] Green, D. and DiCaterino, A. "A Survey of System Development Process Models," Center for Technology in Government, University at Albany, http://www.ctg.albany.edu/publications/reports/survey_of_sysdev, February 1998.
[25] Thakurta, R. and Ahlemann, F. "Requirements Volatility Awareness and Management – A Qualitative Study in German Context," *Proceedings of the 6th Annual Conference on Information Science Technology and Management*, New Delhi, India, 2008.
[26] Tiwana, A. and Keil, M. "The One Minute Risk Assessment Tool," *Communications of the ACM*, Volume 47, Number 11, 2004, pp. 73-77.
[27] "CHAOS Chronicles v3.0.," The Standish Group International Inc., West Yarmouth, USA, http://www.standishgroup.com/chaos/toc.php, 2003.
[28] Wiegers, K., *Software Requirements*, Microsoft Press, Redmond, Washington, 1999.
[29] Yourdon, E., *Death March* (2nd ed.), Prentice Hall PTR, Upper Saddle River, NJ, 2003.

ACKNOWLEDGEMENTS

The author would like to express his sincere gratitude to Dr. Frederik Ahlemann of European Business School for his attention to the work and his continuous comments during the entire duration of the project.

AUTHOR BIOGRAPHY

Rahul Thakurta is an Assistant Professor of Information Systems at Xavier Institute of Management Bhubaneswar, India. His primary research interests are software process and project management, and technology adoption and diffusion. He is also the Managing Editor of *Research World*, and holder of the DAAD Research Fellowship.