Journal of Information Technology Management

ISSN #1042-1319

A Publication of the Association of Management

MANAGING SOFTWARE PROJECTS UNDER FORESEEN UNCERTAINTY

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

ABSTRACT

Uncertainty appears as a significant barrier to projects attaining their intended performance goals; thereby contributing to project failure. Literature on project management under uncertainty has recommended a contingency style of action however based on the premise that the level of uncertainty is static over project duration. We relaxed the assumption by considering variation in the level of uncertainty with project progress following three different patterns viz. increasing, decreasing and uniform. Implementation of the same was carried out by considering analogous patterns of addition of requirements resembling the scenario of foreseen uncertainty during software project development. System dynamics approach was further used in order to analyze the impact of some of the chosen project management policies on the project performance with temporal variations in uncertainty as described above. The results of this study necessitates adoption of different project management policies as the temporal variations in the level of foreseen uncertainty emerged as a contingency factor contributing to variations in project performance. The implications of the findings on research and practice have also been outlined.

Keywords: Foreseen Project Uncertainty, Scope Creep, Project Quality, Resource Management, System Dynamics

INTRODUCTION

Despite four decades of practice in formal software project management, the chance that a project fails to achieve its system and process estimates is still as high as 44% [35]. The reasons for the same have been attributed to the complexities associated with developing the software and the uncertainties characterizing the project development environment [42]. These two phenomena have been reported widely in the project management literature [28, 34]. However the effect is more pronounced in the case of software projects driven by the fact that the product/service to be delivered at the end remains obscure for a significant duration during project development.

Uncertainty has been defined as a risk condition or aspects pertaining to the project environment that contribute to project risks [29]. Uncertainty can arise out of multiple scenarios like unavailability of information, diminishing quality of information, etc. and creates difficulties in accurate estimation of the project outcome [25]. This adds to the risk that the project will fail to attain its performance goals, thereby increasing chances of project failure. Managing uncertainty thus poses a significant challenge to the practitioners and more so in the present business scenario characterized by rapid changes, and the needs to remain abridged with the competition [11, 12].

In this study, we investigate the nature of project management approach in terms of suitability of chosen policies under temporal variation of uncertainty in the context of software projects. We restrict our analysis to foreseen uncertainty which refers to uncertainty that have been identified in advance. To carry out the investigation, we represent foreseen uncertainty in terms of scope creep which refers to additional functionality added during the course of a project. This addition of requirements happen because of change orders or requests generated by the business users or customers during project development or because of functionalities identified by the project team itself, and has been acknowledged as one of the main factors responsible behind failure of software projects [17, 23].

For experimentation purpose, here we consider three fundamental patterns of requirement generation arising out of scope creep viz. increasing (linear rise), decreasing (linear decay), and uniform; and these are used to depict identical variations in the level of foreseen uncertainty with project progress. The experimentation based on simulation is then carried out on an established system dynamics model of software project management proposed by Abdel-Hamid [1]. By considering addition (generation) of requirements following the three different patterns indicated above, the results permit us to make comparison on the effect of some of the chosen project management policies on the project performance.

This paper is organized as follows. The next section discusses on the literature relevant for the work. Subsequently, we elaborate on the methodology adopted to carry out the research. The study results are presented next and then discussed. Finally, in conclusion, we summarize the findings of this investigation and delineate the future research opportunities.

RELEVANT LITERATURE

The importance of uncertainty in the context of managing projects cannot be undermined given its impact on the project outcome. Uncertainty has been acknowledged in project management from very early days. Network theories like the PERT (Program Evaluation and Review Technique), incorporated variation in task durations in the estimation of project completion time [11, 12]. Further developments included use of probabilistic branching techniques and other qualitative approaches [11, 12] in order to plan for uncertainty at the inception of the project.

Studies on project uncertainty have focused on characterizing uncertainty, understanding the types of uncertainty, and offering suggestion on how to manage projects under uncertainty. Of the characterizations offered by several authors, the characterization by Schrader et al. [31] has a more general connotation as observed by De Meyer et al. [11, 12]. Schrader et al. [31] carefully distinguished between uncertainty and ambiguity where uncertainty refers to a scenario where the variables are known but their values are unknown while ambiguity refers to the scenario where the variables or their relationships are itself unclear. However the authors did not make an attempt to link the findings to project management approaches. Studies have also classified uncertainty into different types and have related them to specific project structures [25]. In this study, we adopt the categorization of uncertainty into variation, foreseen, unforeseen and chaos as proposed in De Meyer et al. [11, 12]. A brief description of each is presented below:

- **Variation:** This refers to a range of possible values over which a certain variable characterizing the information can vary. An example of the same could be variation of the time duration of a certain activity between say 10 and 15 days. Such changes can result in change of the project critical path and hence a change in the final project duration. Variations are generally too small to plan and manage individually.
- **Foreseen Uncertainty:** This refers to all previously identified uncertainties which may or may not occur during the course of a project. An example of the same could be occurrence of a specific side effect from administration a specific drug developed by a Pharmaceutical company [11, 12]. The side effect is the foreseen uncertainty in this case. However at start, it is not known whether the side effect will occur, and if it occurs what will be the magnitude of the occurrence. Management of foreseen uncertainty might require derivation of contingency plans, and constant monitoring of project progress.
- **Unforeseen Uncertainty:** This refers to uncertainties which are not identifiable during project planning. In this case, the project team is unaware of the possibility of the occurrence of this uncertain event, and as such there is no alternative plan of action. These kinds of uncertainties are also known as “Unknown unknowns” or “unk-unks”. Unforeseen uncertainty can characterize projects that symbolize new product development, etc.
- **Chaos:** In this case, the project itself lacks stable descriptions of objectives, assumptions, and goals. The basic structure of project plan is itself uncertain in this case. R&D projects suffer from this kind of uncertainty. In absence of stable structure, the final outcome of these projects is often different from what was intended at the start [11, 12].

Research of managing projects under uncertainty has emphasized the importance of matching project management approaches to the nature of uncertainty characterizing the project. McFarlan [24] in his paper concluded the same with the uncertainty profile of the project measured by three dimensions of project size, project structure, and technology experience. De Meyer et al. [11, 12] tried to differentiate project management style and stakeholder management among the four types of uncertainty identified above. Results suggested use of flow chart mode of decision making for uncertainty of the type variation, decision tree under foreseen uncertainty, evolving decision tree under unforeseen uncertainty, and iterative decision tree in case of chaos. Pitch et al [28] analyzed project management strategies under uncertainty, ambiguity and complexity measured based on information adequacy. The results of their study identified three project management strategies viz. learning, instructionism, and selectionism, and further demonstrated how successful application of these strategies is contingent on the uncertainty and complexity of the project outcome. Extending the work, Sommer and Loch [34] investigated the use of selectionism and learning strategies in projects characterized by complexity and unforeseen uncertainty. Their findings differentiated the usage of these approaches under unforeseen uncertainty and under complexity measured in dimensions of project complexity measured in terms of project size and number of interactions. Söderlund [33] in his review paper categorized the project management styles reported in the above mentioned papers under the 'contingency school' of thought. The contingency school based on the classical organizational contingency theory [7] postulates the needs to match project management approaches with the project environment.

We base our work on the uncertainty typology proposed in De Meyer et al. [11, 12] and discussed above. Their work has recommended the need to adopt a contingency style of management depending upon the type of uncertainty characterizing the project. Now considering each type of uncertainty, the level of uncertainty might also change with project progress. This raises the question of whether the adopted project management approaches also needs to account for such temporal variations in uncertainty in order to achieve the intended outcome.

In order to carry out the investigation, we focus on certain variations in the level of foreseen uncertainty during project development. Scope creep which has been acknowledged as a major risk factor affecting software projects [17], represents a form of foreseen uncertainty characterizing the project environment. However at the onset of the project, the extent of scope creep the project is likely to face cannot be ascertained. The variations in the level of foreseen uncertainty are represented in our study by considering requirement addition (generation) following different patterns (discussed later). The impact of such variations on suitability of project management policies and project performance is the subject of inquiry in this paper. A review of relevant literature of scope creep, project performance, and project management policies is given below.

Scope Creep

Scope creep has mostly been studied from the context of requirement volatility which also takes into account updates and deletions of the project's requirement set. Studies on requirement volatility in software projects have mostly taken an event oriented viewpoint in order to understand its nature and source; its effect; and its management strategies. Some notable findings of these studies are listed below.

- Requirement volatility can occur not only in terms of magnitude [3], but also in terms of pattern of requirement addition [40].
- The causes of requirement volatility have been attributed to presence of inconsistencies or conflicts among requirements; activities carried out during the project like defect fixing, functionality correction [26]; evolving user/customer knowledge and priorities; technical, schedule or cost related problems, change in work environment [10], and process model selection decisions [22]. In this context, a process model (also known as 'systems development life cycle model' or SDLC) describes the various stages involved in an information system development project, and provides a mechanism to plan for and manage project execution [30].
- The maximum effect (of requirement volatility) is observed on developer productivity [13]. Late additions in requirements significantly impact the proportion of high severity defects resulting in deterioration of product quality [45].
- Suggestions to managing project under varying magnitude of requirement volatility include adoption of specific frameworks (like formation of change control boards [16], and specifying the project execution strategy upfront like selecting the process model for the project [38], and adoption of specific techniques during project development (for example usage of joint application design (JAD), and configuration management [16], base lining requirements [43], proper change management planning [44], etc.).

The literature does not delve into how management approaches are likely to be influenced with requirement addition (generation) following different patterns. We expect that the project management approach will be contingent on such pattern-wise variation [39], thereby affecting project performance.

Project Performance

A measure of software project performance can be arrived upon based on well defined metrics such as effort (cost), schedule, quality, etc. In this study, we have chosen quality as the indicator of software project performance. We use the term quality or project quality interchangeably to refer to the quality of the product/service delivered to the end users at the completion of the project. Project cost is more a point of concern to the project organization that is entrusted with developing the software. Now considering project schedule, if the intended product/service is developed as per the schedule but does not meet the pre-specified quality requirements, the delivery of the product/service might get postponed ([18]; pp. 81). On the contrary, the delivery provided to the users/customers at desired quality level after deadline may still be acceptable given some penalty borne by the project organization as per the contractual negotiation.

Studies on software project quality have primarily focused on the different quality improvement approaches in order to increase acceptance of the project deliverables. In this regard, Basili and Rombach [4] provide a five step methodology for software process improvement based on analysis of defect related data. Liu et al. [21] present an approach of integrating formal specification, review, and testing activities with a view to remove errors and identify missing requirements. Wagner et al. [41] present the findings of a survey on quality models in practice conducted among four software companies in order to update on the usage, techniques, and associated problems encountered in practice. Li et al. [20] investigate the effectiveness of review, process audit, and testing and their overall contribution to return on investment (ROI).

Project Management Policies

Project management in the context of software projects encompasses a broader range of tasks like allocating resources, planning, assessing risks, selecting management policies, processes and tools, etc with multiple ways of accomplishing each of these [14]. In this study, we concentrate on certain resource management policies which have been reported being used in different studies. Discussion on resource allocation policies in software projects is found to be limited, probably driven by the fact that each software project represents a unique scenario [27]. The available literature discusses different resource allocation policies with optimal effect on project performance. Some of these policies are use of proportional and foresighted resource forecasting techniques [15], variation of resource adjustment times [19], use of high-skilled resources [5], and overstaffing the project from the onset [8]. The choice of these policies was found to be based on “heuristic knowledge, subjective perception, and instinct” [2].

METHODOLOGY

Task Environment

Project management is a complex and dynamic phenomenon involving interplay of a wide range of hard and soft factors [9] which prompted us to use the system dynamics (SD) [37] approach in our study. The basic premise in SD is that system behaviour results from interaction among its feedback loops. Model building begins with development of a causal loop diagram that consists of a collection of causal links, each having a certain polarity. A positive (negative) link implies a reinforcing (balancing) relation where a positive change in the cause results in a positive (negative) change in the effect. A double line intersecting a link represents delays in an effect. A causal loop is formed by a closed sequence of causal links. A negative feedback loop has an odd number of negative polarity links, while a positive loop has an even number of negative links. The causal loop graph can be mapped to a mathematical model consisting of a system of difference equations, which can be simulated under different parametric conditions.

We contextualize the project setting by considering a familiar in-house medium-sized project implementing the waterfall methodology. The choice of waterfall methodology was driven out of its observed predominance even in projects endangered because of scope creep [38]. Given the findings, here we opt for Abdel-Hamid’s [1] SD model based on waterfall methodology. The model effectively integrates all relevant processes of software development like development, quality assurance, testing, rework, etc. It also allows one to investigate for the effect of changes in project human resource, project size, and project plan on project progress; and in the process appreciate the dynamics involved in software development so as to better manage the changes. The model was constructed based on in-depth interviews with sixteen software project managers and supported by extensive literature review. The interviews were used to fill in gaps in the literature. The model was subsequently validated based on case studies conducted by the author, and supplemented by expert review techniques [1]. The model further assumes the following:

- The software tasks are divisible and can be carried out in parallel.

- The requirements once specified do not change. Only new requirements get added in course of the project.
- Quality assurance gets precedence over development activity during the course of the project.

The model uses a factor ‘Task Underestimation Fraction’ (not shown) that captures fraction of undiscovered tasks that get added to the project scope, and is a measure of the magnitude of scope creep during project development. Further, there is no imposed cap on the maximum allowable delay during project development.

Figure 1 presents the causal loop diagram of the problem embodied in the model structure. The causal loop diagram was arrived upon by identifying the structure representing the problem of interest from Abdel-Hamid’s SD model. The model behaviour can be understood based on how the different feedback loops influence the dynamics. A description of the behavior of the causal loop diagram is provided below, with the model parameters shown in italics.

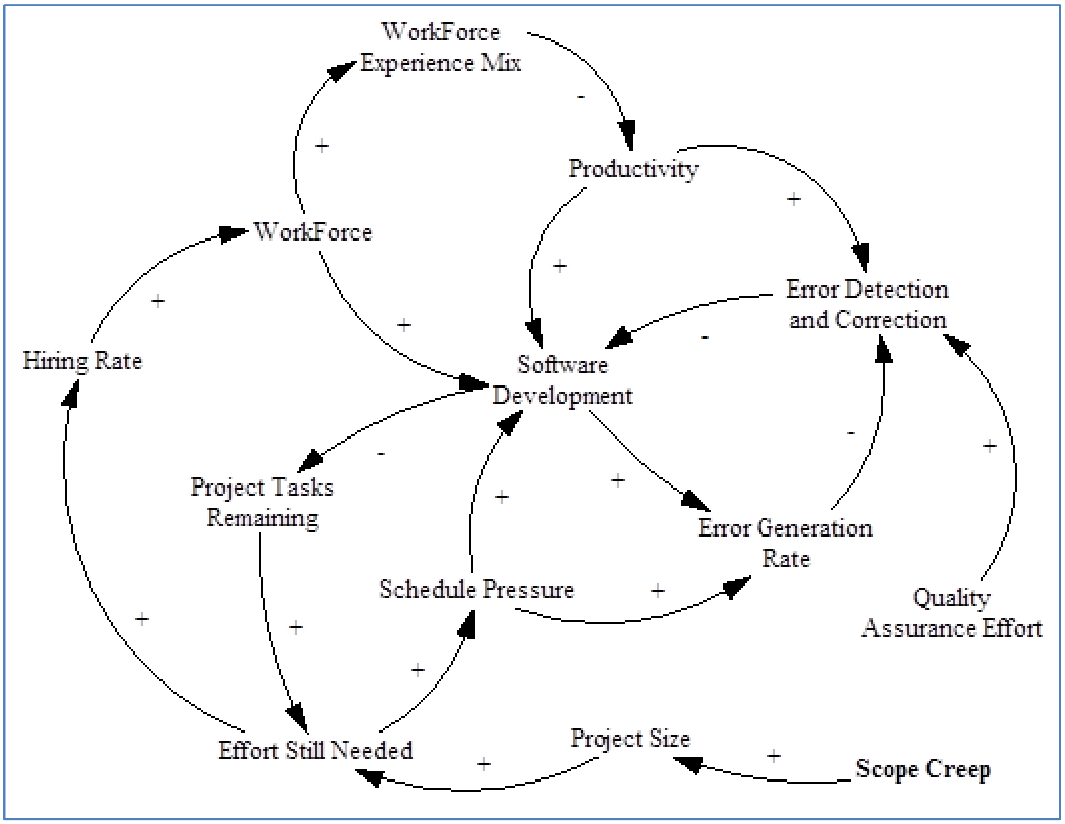


Figure 1: Model Causal Loop Diagram

Scope creep during project development leads to augmentation of project size. With increase in project size, the estimate of effort still needed to complete the project which is a function of project size [6], also increases. This increased effort requirement positively affects the schedule pressure, and leads to generation of more errors because of higher error generation rate. High error generation rate in turn negatively impacts error detection and correction, thereby hampering project quality. On the other hand, the increased effort requirement (effort still needed) arising because of scope creep also induces hiring (hiring rate), which increases the project workforce. Presence of a higher workforce boosts up software development resulting in more number of tasks pending for testing. Tasks processing at a higher rate bring down the effort still needed (because of reduction in project tasks remaining), and thus helps to reduce the schedule pressure. The decrease in schedule pressure reduces the error generation rate and thereby positively influences project quality.

The dynamics is further complicated by the pattern in which change orders are generated during project development (i.e. requirement generation pattern), and the resource allocation policy adopted. The later changes the workforce experience mix (i.e. ratio of rookies and experienced professionals in the workforce) and thus affects the software process owing to the fact that rookies are less productive and also more error-prone compared to their experienced counterparts.

The model parameters (Table 1) were set as per the TRW Inc. case study [1], which matches our project context. The project environment portrayed in the case study represents the development environment of a familiar in-house organic medium-sized software project. The reported project is medium sized, having initial specified job size as 64,000 delivered source instructions (DSI) which corresponds to 1067 Tasks. The initial estimates of effort and schedule were derived using COCOMO (‘constructive cost model’: [6]) as follows:

Table 1: Initial Parameter Estimates

Parameter	Estimate
Initial Specified Job Size	1067 Function Point
Initial Estimated Effort	3594 Person-Days
Initial Schedule Estimate	348 Days
Project Average FTE	10.3 Persons

The value of project average full-time-equivalent (FTE) professionals, was arrived at 10.3 persons; implying ten persons to be working fulltime on the project, and one person to be devoting 30% of his/her daily work-hour on the project.

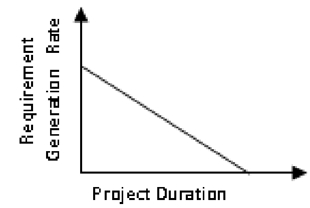
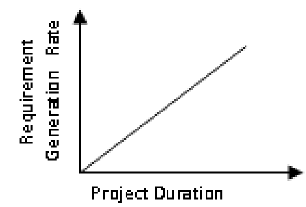
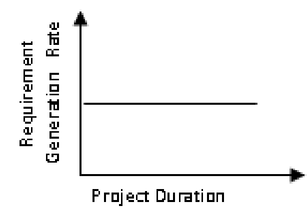
Experiment Design

In a real life scenario, generation of change orders contributing to scope creep appears more as a random phenomenon. Taking a closer look, such random variations can be seen to be composed of different geometrical patterns that are observable in reality. The basic structures of such patterns are depictions of increasing trend (linear rise), decreasing trend (linear decay), and uniform trend, which are described below for the purpose of experimentation. The evidences of occurrence of requirement changes following these patterns has been provided in Stark et al. [36] (pp. 8) in a different context (maintenance projects).

- **Linear Decay:** Here, initial high rate of change order generation decreases linearly with time. Close collaboration with users generates high rate of requirement generation initially. With time, the requirements gradually stabilize and the rate of change declines.
- **Linear Rise:** Here the rate of change order generation increases linearly with time. Users’ and developers’ learning curves make project tasks grow at an increasing rate.
- **Uniform Variation:** Constant rate of change order generation throughout the project’s duration which causes project tasks to grow linearly.

The above experimental patterns of requirements generation further represents three different categories of foreseen uncertainty. Linear decay pattern of requirement generation symbolizes the case where the level of uncertainty can be visualized to decrease with time (decreasing uncertainty). Conversely, linear rise pattern of requirement generation represents the reverse case with the level of uncertainty increasing with time (increasing uncertainty). Lastly, uniform variation represents the case with the level of uncertainty not showing any significant fluctuations over time (constant uncertainty). Table 2 provides a description of these categories, along with the representation of the requirement generation pattern that characterises the uncertainty in each case.

Table 2: Categories of Foreseen Uncertainty and Representation

Categories (Nature of Foreseen Uncertainty)	Description	Representation
Decreasing Uncertainty	Here the level of uncertainty is assumed to decrease with project progress.	
Increasing Uncertainty	Here the level of uncertainty is assumed to increase with project progress.	
Constant Uncertainty	Here the level of uncertainty is assumed to remain constant over the project development period	

Workforce management policies represent the intervention mechanisms that influence the underlying control structure governing the project workforce. For the purpose of experimentation, we chose the following three policies from the literature given their relevance in the context of software projects:

- **POLICY 1: Augmenting development team skill**

  This represents a scenario where members recruited into the project have a higher skill set. This is possible if the organization undertaking the project have previous experience in carrying out similar sort of projects. In this case, the employees of that organization will be having domain expertise, and the assimilation time associated with project related training is expected to reduce. In order to implement the same, we set the value of 'assimilation delay' to 10 (working days). This is equivalent to two work weeks (a work week consists of 5 working days). The value is also close to the one (12 days) reported in Sengupta et al. [32].

- **POLICY 2: Overstaffing the project at start**

  Here the project maintains additional bench strength based on the expectation of requirement generation arising out of scope creep during project development. Usage of this overstaffing strategy can be noticed in Collofello et al. [8]. Projects which have high business impact or face huge time constraint can employ this strategy. Since scope creep represents a case of foreseen uncertainty, here we implement overstaffing by setting starting workforce at the value taking into account expected size of additional requirements arising out of change order generation during project development. The calculation led to the size of starting workforce as 7.9 (expected project average FTE: 15.8 based on COCOMO).

- **POLICY 3: Reducing resource allocation delay**

  Resource allocation delays represent the average time required to hire in extra personnel from outside the organization. Past research has indicated that tuning resource allocation delays to the project characteristics helps to improve the project performance [19]. Reduction in hiring delay is possible through pre-hiring of desired competency, or maintaining bench strength for the project. Based on reported evidence in Lee et al. [19], here we set the hiring delay as 5 working days (one work week).

Table 3 lists the parameter values relevant to implementation of the stated policies. The values of other parameters are same as that of the 'Base' case (i.e. the behaviour as depicted by the model structure without implementation of any of the resource allocation policies).

Table 3: Parameter Changes for Policy Implementation

Policy #	Parameter Values
1.	Assimilation Delay = 10 Working Days
2.	Starting Workforce Level = 7.9
3.	Hiring Delay = 5 Working Days

In order to carry out simulation, the cause-and-effect model shown in Figure 1 was converted into a simulation model represented as a system of difference equations. Such a representation, also known as a stock and flow diagram in the language of SD [37] was implemented using commercially available iThink software. The software uses visual diagramming for constructing dynamic models, provides level of abstraction, allows for graphical representation of table functions, and provides utilities for sophisticated model components beyond previous implementation of system dynamics [22].

In the simulation model, we set a quality objective of 75% implying project in concern has high quality requirements which appropriately matches our study objectives. This will ensure that at least 75% of the errors generated during software development get detected and rectified; the final figure however contingent on how the dynamics unfolds under various run conditions. Project quality is measured with the help of the metric defect density (defined as the ratio of the number of defects contained in the delivered product/service and project size). The task underestimation fraction is set at 0.67 implying that the initial project size can grow by 50% during project development because of scope creep. The growth of project tasks under the three different requirement generation patterns (i.e. linear decay, uniform, and linear rise; see Figure 2) is shown in Figure 3. Since the task underestimation fraction is same in all cases, the same amount of tasks always gets delivered at the end. However, different change order generation patterns modulate the growth of project tasks in different ways. The next section elaborates on the results.

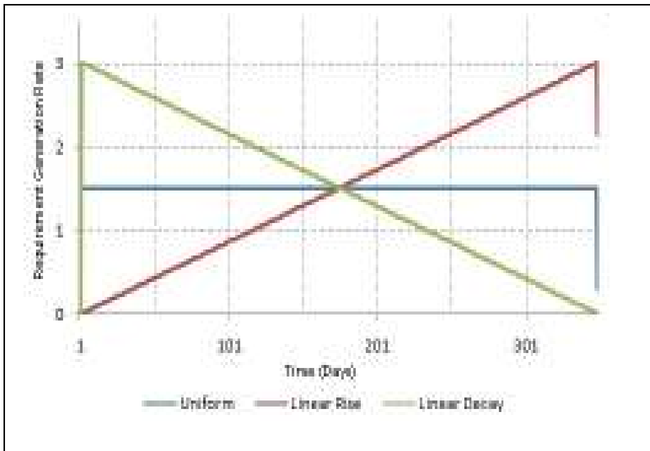


Figure 2: Change Order Generation Rates

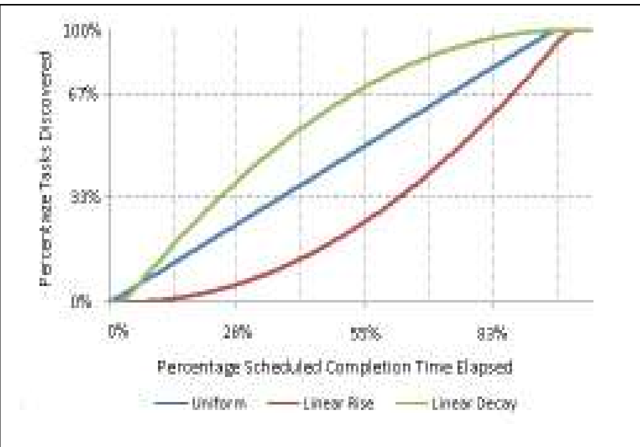


Figure 3: Growth of Project Tasks

RESULTS

Project Performance at Decreasing Uncertainty

Here the level of uncertainty is found to decrease with project progress, and is represented in our model by the linear decay pattern of requirement generation. Table 4 shows comparison of project performance for 'Base' and the different policies (Table 3) under the linear decay pattern of requirement generation. The values in each cell in Table 4 shows the actual result of simulation and a percentage (%) figure given within brackets. The percentage figure indicates where the values of each parameter stand with respect to the 'Base' (taken as 100%) for each of the different policies. In all cases a total of 1592 tasks were processed (i.e. 50% above the initial specified as given in Table 1). Defect density is calculated at the end of each simulation run by dividing the number of errors escaped by KDSI (i.e. 1000 DSI). Obviously, higher value of defect density indicates low quality of final delivery and vice versa.

Project Performance at Increasing Uncertainty

Here we look at the reverse scenario with the level of uncertainty increasing with project progress. This is represented in our model by the linear rise pattern of requirement generation. Table 5 presents the corresponding results for the 'Base' case, and when the different policies were used (same representation as Table 4 has been used).

Table 4: Effect of Different Policies under Linear Decay

	Base	Policy1 (Better Skill)	Policy 2 (Over Staffing)	Policy 3 (hiring delay)
Total Effort (Person-Days)	6009 (100%)	5933 (99%)	6000 (100%)	6173 (103%)
Completion Date (Days)	463 (100%)	414 (89%)	459 (99%)	440 (95%)
FTE Manpower (Person)	12.97 (100%)	14.33 (110%)	13.06 (101%)	14.02 (108%)
No. of Errors Generated	2249 (100%)	2157 (96%)	2157 (96%)	2274 (101%)
No. of Errors Escaped	536 (100%)	499 (93%)	432 (81%)	479 (89%)
Defect Density (No. of Errors Escaped/KDSI)	5.70 (100%)	5.31 (93%)	4.60 (81%)	5.10 (89%)

Table 5: Effect of Different Policies under Linear Rise

	Base	Policy1 (Better Skill)	Policy 2 (Over Staffing)	Policy 3 (hiring delay)
Total Effort (Person-Days)	5772 (100%)	5783 (100%)	5643 (98%)	6042 (105%)
Completion Date (Days)	542 (100%)	454 (84%)	529 (98%)	475 (88%)
FTE Manpower (Person)	10.66 (100%)	12.73 (119%)	10.68 (100%)	12.71 (119%)
No. of Errors Generated	2166 (100%)	1865 (86%)	2073 (96%)	2073 (96%)
No. of Errors Escaped	504 (100%)	383 (76%)	467 (93%)	376 (75%)
Defect Density (No. of Errors Escaped/KDSI)	5.36 (100%)	4.07 (76%)	4.96 (93%)	4.00 (75%)

Project Performance at Constant Uncertainty

Lastly we look at the scenario where the level of uncertainty remains constant over the project development period. This is represented in our model by the uniform pattern of requirement generation. Table 6 presents the corresponding results for the 'Base' case, and when the different policies were used (same representation as Table 4 has been used).

Table 6: Effect of Different Policies under Uniform Pattern

	Base	Policy1 (Better Skill)	Policy 2 (Over Staffing)	Policy 3 (hiring delay)
Total Effort (Person-Days)	5734 (100%)	4796 (84%)	5247 (92%)	7145 (125%)
Completion Date (Days)	565 (100%)	464 (82%)	540 (96%)	468 (83%)
FTE Manpower (Person)	10.15 (100%)	10.35 (102%)	9.71 (96%)	15.27 (150%)
No. of Errors Generated	2268 (100%)	1878 (83%)	2041 (90%)	2041 (90%)
No. of Errors Escaped	494 (100%)	388 (79%)	453 (92%)	403 (82%)
Defect Density (No. of Errors Escaped/KDSI)	5.25 (100%)	4.12 (79%)	4.81 (92%)	4.28 (82%)

DISCUSSION

Project manager's style and toolbox need to reflect the uncertainty profile of the project in order to ensure its successful completion [11, 12]. By focusing on the foreseen uncertainty category, we tried to investigate whether the adopted project management approaches also needs to be contingent on temporal variations of uncertainty in order to achieve the intended outcome. To carry out the investigation, variation of foreseen uncertainty was categorized into decreasing, increasing, and constant uncertainty as mentioned in Table 2. A discussion on the results is provided below:

Decreasing Uncertainty

We have characterized the same by considering linear decay pattern of change order generation during project development. The maximum amount of uncertainty is at the start of the project, with the level decreasing at the project proceeds towards completion. Simulation results indicate overstaffing (Policy 2) to lead to higher project performance assessed in terms of quality of the product/service delivered to the users. Having a larger workforce at one's behest is expected to assist the project manager to counter the increased schedule pressure arising out of high rate of change order generation early on.

Increasing Uncertainty

This has been characterized by considering linear rise pattern of change order generation during project development. In this case, the level of uncertainty increases as the project proceeds towards completion. Hence, in this project execution environment, the level of uncertainty is least at the start of the project, but increases with project progress, with the maximum level at the end of the project. Since increasing uncertainty is expected to jeopardize decision making, and more so during the middle, end stages of the project, approaches which are more flexible are expected to be successful in this context [11, 12].

Results based on simulation indicated augmentation of development skills (Policy 1) and reduction of hiring delay (Policy 3) to be contributing to improved project performance in this case. Reducing hiring delay is a flexible approach as in this case hiring needs to be triggered in the quickest possible time as and when the need arises during project development. Similarly having a skilled development team needs prior planning so as to ensure that the identified resource is not locked into some other engagements when needed. The later scenario can also be managed by usage of the resource sharing approach. In this case, the identified resources may be assigned certain commitment from the concerned project, on which the resources can work in parallel or as per the assignments. This will ensure that the project does not have to wait for the release of needed resources from other work assignments.

Constant Uncertainty

Constant uncertainty has been portrayed using uniform pattern of change order generation during project development. In this case, the level of uncertainty remains the same over the project duration. In absence of changes in the uncertainty level, its effect on team productivity might not vary by much (productivity changes are likely to happen because of other factors like exhaustion, communication overhead, etc). Thus it is expected that any policy that contributes towards greater productivity of the workforce is likely to proof useful. The results also indicate Policy 1 (better skilled team) to be the most effective, thereby conforming to the expectations.

CONCLUSION

It's expected that the project management approach needs to be contingent on the characteristics of the project. The research presented in this paper tried to investigate this contingent behaviour by studying project performance measured in terms of project quality under temporal variations in the level of foreseen uncertainty (viz. decreasing, constant, increasing). By modelling the uncertainty categories based on requirement generation patterns, simulation results indicate variations in project performance across the chosen management policies under the experimental scenarios. Given the project setting, overstaffing led to the best results in case of decreasing uncertainty scenario, while it was not that effective in case of other uncertainty categories considered in this study. The results emphasized the need for contingency planning in face of temporal variations of project uncertainty in order to achieve the intended objectives.

The findings of this study should be considered in the light of its inherent limitations. The extent of variations in project parameters across the policy choices was not very high given that the project did not have any imposed schedule penalty. Given the relatively small magnitude of variations for some parameters, it is also possible that results might change with the level of overstaffing as the amount of resources committed to the project is also dependent on the presence on bench strength or the project budget. Thus if a project starts with an extremely large workforce size than what has been assumed in the experiments here, the project performance can be expected to be different. These limitations don't undermine but rather emphasize the need to adopt contingency planning depending upon the expectation of uncertainty variation during project development. Failure in this regard is likely to contribute towards user dissatisfaction related to the quality of the final deliverable and the end result can be failure of the project.

In terms of the contributions, the study differs from several published studies on scope creep by taking a pattern oriented viewpoint of the phenomenon. The work also contributes to the project management framework proposed by De Meyer et al. [11, 12]. The framework demonstrated how the project management activities are likely to change depending upon the uncertainty profile of the project in dimensions of variation, foreseen, unforeseen, and chaos or turbulence. The present study adds to the findings by further categorizing foreseen uncertainty into three categories viz. increasing, decreasing, and constant. The effects of chosen resource management policies on the project performance were observed to be different in these categories. Thus extending De Meyer et al. [11, 12], we may say that the effect of project management activities will not only vary across the uncertainty dimensions, but also within the uncertainty dimensions as the findings here have shown.

Apart from addressing the study limitations, future research may analyze the effect of the chosen policies on other project parameters like the total effort expended, as this ultimately translates as cost to the project organization. Impact of project development constraints like competency of available workforce, cost penalty, schedule penalty etc on the project performance can be investigated. It is also possible to combine the policy choices, and investigate the effects: for example over-staffing at start (Policy 2), and also reducing the hiring delay (Policy 3) for subsequent hiring during project development. In the present contexts of shorter time-to-market and stringent quality requirements, it would be interesting to see how our research influences design of project control mechanisms in organizations.

REFERENCES

[1] Abdel-Hamid, T.K. and Madnick, S.E., *Software Project Dynamics: An Integrated Approach*, Prentice Hall, Englewood Cliffs, NJ, 1991.

[2] Acuña, S.T., Juristo, N. and Moreno, A. "Emphasizing Human Capabilities in Software Development," *IEEE Software*, 2006, pp. 94–101.

[3] Barry, E.J., Kemerer, C.F. and Slaughter, S. "Environmental Volatility, Development Decisions, and Software Volatility: A Longitudinal Analysis," *Management Science*, Volume 52, Number 3, 2006, pp. 448-464.

[4] Basili, V.R. and Rombach, H.D. "Tailoring the Software Process to Project Goals and Environments," *Proceedings of the 9th International Con-*

ference in Software Engineering, California, USA, 1987.

[5] Beaver, J.M. and Schiavone, G.A. "The Effects of Development Team Skill on Software Product Quality," *ACM Sigsoft Software Engineering Notes*, Volume 31, Number 3, 2006, pp. 1-5.

[6] Boehm, B., Clark, B., Horowitz, E., Christopher, J., Madachy, R. and Selby, R. "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0," *Annals of Software Engineering*, Volume 1, 1995, pp. 57-94.

[7] Burns, T. and Stalker, G. *The Management of Innovation*, Tavistock, London, 1961.

[8] Collofello, J., Rus, I., Chauhan, A., Houston, D., Sycamore, D.M. and Smith-Daniels, D. "A System Dynamics Software Process Simulator for Staffing Policies Decision Support," *Proceedings of the 31st Annual Hawaii International Conference on System Sciences*, Hawaii, USA, 1998.

[9] Crawford, L. and Pollack, J. "Hard and Soft Projects: A Framework for Analysis," *International Journal of Project Management*, Volume 22, Number 8, 2004, pp. 645–653.

[10] Davis, A.M., Nurmuliani, N., Park, S. and Zowghi, D. "Requirements Change: What's the Alternative?" *Proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference*, 2008, pp. 635-638.

[11] De Meyer, A., Loch, C.H and Pich, M.T. "A Framework for Project Management under Uncertainty," *Sloan Management Review*, Volume 43, Number 2, 2002, pp. 60-67.

[12] De Meyer, A., Loch, C.H and Pich, M.T. "Managing Project Uncertainty: From Variation to Chaos," *MIT Sloan Management Review*, 2002, pp. 60-67.

[13] Ferreira, S., Collofello, J., Shunk, D. and Mackulak, G. "Understanding the Effects of Requirements Volatility in Software Engineering by using Analytical Modeling and Software Process Simulation," *Journal of Systems and Software*, Volume 82, Number 10, 2009, pp. 1568-1577.

[14] Hughes, B. and Cotterell, M. *Software Project Management* (5th ed.), McGraw-Hill, London, 2009.

[15] Joglekar, N.R. and Ford, D.N. "Product Development Resource Allocation with Foresight," *European Journal of Operational Research*, Volume 160, 2005, pp. 72-87.

[16] Jones, C. *Estimating Software Costs*. McGraw Hill, Hightstown, NJ, 1998.

[17] Jones, C., *Assessment and Control of Software Risks*, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[18] Keil, M., Cule, P.E., Lyytinen, K. and Schmidt, R.C. "A Framework for Identifying Software Project Risks," *Communications of the ACM*, Volume 41, Number 11, 1998, pp. 76–83.

[19] Lee, Z.W., Ford, D.N. and Joglekar, N. "Effects of Resource Allocation Policies for Reducing Project Durations: A Systems Modeling Approach," *Journal of Systems Research and Behavioral Science*, Volume 24, Number 6, 2007, pp. 551-566.

[20] Li, Q., Shu, F., Boehm, B., and Wang, Q. "Improving the ROI of Software Quality Assurance Activities: An Empirical Study," Proceedings of the 2010 International Conference on New Modeling Concepts for Today's Software Processes: Software Process (ICSP'10), Berlin, Heidelberg, 2010.

[21] Liu, S., Tamai, T., and Nakajima, S. "Integration of Formal Specification, Review, and Testing for Software Component Quality Assurance," *Proceedings of the 2009 ACM symposium on Applied Computing (SAC '09)*, New York, USA, 2009.

[22] 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, Los Angeles, California, 1994.

[23] Mathiassen, L., Saarinen, T., Tuunanen, T. and Ross, M. "A Contingency Model for Requirements Development," *Journal of the Association for Information Systems*, Volume 8, Number 11, 2007, pp. 570-598.

[24] McFarlan, F.W. "Portfolio Approach to Information Systems," *Harvard Business Review*, 1981 (Sept-Oct).

[25] McLain, D. "Quantifying Project Characteristics related to Uncertainty," *Project Management Journal*, Volume 40, 2009, pp. 60-73. DOI=10.1002/pmj.20132.

[26] Nurmuliani, N., Zowghi, D. and Fowell, S. "Analysis of Requirements Volatility during Software Development Life Cycle," *Proceedings of the 2004 Australian Software Engineering Conference*, Innsbruck, Austria, 2004.

[27] Otero, L.D., Centeno, G., Ruiz-Torres, A.J. and Otero, C.E. "A Systematic Approach for Resource Allocation in Software Projects," *Computers & Industrial Engineering*, Volume 56, Number 4, 2009.

[28] Pich, M.T., Loch, C.H. and De Meyer, A. "On Uncertainty, Ambiguity and Complexity in Project Management," *Management Science*, Volume 48, Number 8, 2002, pp. 1008-1023.

[29] Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK®*

guide), Third Edition, Project Management Institute, Newtown Square, PA., 2004.

[30] Scacchi, W. Process Models in Software Engineering. In J.J. Marciniak (ed.), *Encyclopedia of Software Engineering* (2nd Edition), John Wiley and Sons, New York, 2001.

[31] Schrader, S., Riggs, W.M. and Smith, R.P. "Choice over Uncertainty and Ambiguity in Technical Problem Solving," *Journal of Engineering and Technology Management*, Volume 10, 1993, pp. 73-99.

[32] Sengupta, K., Abdel-Hamid, T.K and Bosley, M. "Coping with Staffing Delays in Software Project Management: An Experimental Investigation," *IEEE Transactions on Systems, Management, and Cybernetics*, Part A 29, 1, 1999, pp. 77-91.

[33] Söderlund, J. "Pluralism in Project Management: Navigating the Crossroads of Specialization and Fragmentation," *International Journal of Management Reviews*, Volume 13, 2011, pp. 153-176.

[34] Sommer, S. and Loch, C. "Selectionism and Learning in Projects with Complexity and Unforseeable Uncertainty," *Management Science*, Volume 50, Number 10, 2004, pp. 1334-1347.

[35] Standish Group International, Inc. "Chaos Summary", **http://www.standishgroup.com**, 2009.

[36] Stark, G.E., Oman, P., Skillicorn, A. and Ameele, A. "An Examination of the Effects of Requirements Changes on Software Maintenance Releases," *Journal of Software Maintenance*, Volume 11, Number 5, September 1999, pp. 293-309.

[37] Sterman, J.D. Business Dynamics: Systems Thinking and Modeling for a Complex World, Irwin/McGraw-Hill, Boston, MA, 2000.

[38] Thakurta, R. and Ahlemann, F. "Understanding Requirements Volatility in Software Projects – An Empirical Investigation of Volatility Awareness, Management Approaches and their Applicability," *Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS)*, Hawaii, USA, 2010.

[39] Thakurta, R. and Suresh, P. "Sensitivity of Project Quality to Different Patterns of Requirement Volatility," *Proceedings of the International Conference on Software Engineering (CONSEG-2011)*, Bangalore, India, 2011.

[40] Thakurta, R., Roy, R. and Bhattacharya, S. "Impact of Requirements Discovery Pattern on Software Project Outcome: Preliminary Results," *Proceedings of the 42nd Annual Hawaii International Conference on System Sciences*, Hawaii, USA, 2009.

[41] Wagner, S., Lochmann, K., Winter, S., Goeb, A. and Klaes, M. "Quality Models in Practice: A Preliminary Analysis," *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM '09)*, Washington, DC, USA, 2009.

[42] Wallace, L. and Keil, M. "Software Project Risks and their Effect on Outcomes," *Communications of the ACM*, Volume 47, Number 4, April 2004, pp. 68-73. DOI=10.1145/975817.975819, **http://doi.acm.org/10.1145/975817.975819**

[43] Wiegers, K. *Software Requirements*, Microsoft Press, 1999.

[44] Young, R.R. *Effective Requirements Practices*, Addison-Wesley, Boston, MA, 2001.

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

AUTHOR BIOGRAPHY

Rahul Thakurta is an Associate Professor of Information Systems at Xavier Institute of Management Bhubaneswar (XIMB), India. He received his Fellowship in Management Information Systems from Indian Institute of Management Calcutta. He teaches courses pertaining to Information Systems (IS) and Operations Management disciplines. His broad research interests include areas like planning and implementation of IS infrastructures, management of software ventures, design science, and technology adoption and diffusion. He has been the recipient of Infosys Research Fellowship and DAAD Short- Term Research Scholarship. He is also the Managing Editor of *Research World* and *Vilakshan* journals published by XIMB.