INVESTIGATING THE IMPACT OF RESOURCE MANAGEMENT POLICIES ON PROJECT QUALITY ASSURANCE UNDER REQUIREMENT VOLATILITY

Research-in-Progress

Rahul Thakurta¹
Xavier Institute of Management
Xavier Square, Bhubaneswar, India
rahul@ximb.ac.in

Suresh P
Infosys Technologies Limited
Hebbal Electronics City, Mysore, India
Suresh_P01@infosys.com

Abstract

Requirements of a project are found to change in various ways during the course of the same. The effects of varying magnitude of requirement volatility on different project parameters like effort, schedule, etc have been well investigated. However, there has been no analysis on how different ‘patterns’ of requirement volatility influence selection of project management approaches and the resultant effect on project performance. To address this exigency, here, using system dynamics modeling, we investigate the impact of different resource allocation strategies on project quality assurance activity under two experimental patterns of requirement volatility. Results indicate variations in quality metrics like error generation, error detection, quality assurance effort, depending upon how the two different requirement volatility patterns and resource allocation policies impact the project dynamics. These initial findings re-inforce the need to adopt pattern-dependent contextual project management approaches, and also pave the way for a detailed exploration to improve project performance.

Keywords: Requirement volatility pattern, quality assurance, resource allocation, system dynamics

¹ Work accomplished as part of his internship with Infosys Technologies Limited
Introduction

Despite advances in software engineering over the past 30 years, most software projects still experience numerous requirement changes during their life cycle, which is brought about by the dynamic nature of development activities (Srinivasan 2009; Wallace et al. 2004). Requirement volatility, which refers to the change in requirements (in terms of the number of additions, deletions, and modifications) during the software development life cycle, has been identified as a significant risk affecting software projects (Boehm 1991; Hoorn et al. 2007; Mathiassen et al. 2007).

The impact of the magnitude of requirement volatility on different project performance parameters have been reported by several authors (a brief review is present in Thakurta and Ahlemann 2010). However, for a given magnitude, requirement volatility can be observed to happen following different patterns. These patterns refer to the different geometric shapes of requirements generation, based on how change orders/requests are issued during project development. Examples of such patterns are exponential decay (here the rate of change order generation is highest at the beginning, and the rate drops down non-linearly with time) (Zowghi and Nurmuliani 2002), exponential rise (here the rate of change order generation is lowest at the start; the rate increases non-linearly with time) (Abdel-Hamid and Madnick 1991), and triangular (here the rate of change order generation increases upto a point near the mid-stages of the project, and then starts to decrease) (Houston et al. 2001). This raises the question of what could be the influence of such patterns on project management and the resultant effect on project performance, which as per our knowledge has not been researched or reported in the current literature.

We anchor our work on the project contingency theory (Shenhar and Dvir 2007) which highlights the need for contextual project management practices to increase chances of project success. By considering pattern-wise change in project requirements, we investigate the efficacy of different resource allocation policies as project management tasks on project quality. The study focuses on the quality assurance (QA) activity, with quality being measured with the help of the metric QA effectiveness which is defined as the ratio of number of errors detected and QA effort expended. The selection of QA activity is driven by the fact that error detection and correction at this stage is relatively easy and also less expensive (Abdel-Hamid 1988). Further, a low value of QA effectiveness will imply either more errors in the final product or service delivered to the users, or a higher expenditure of project effort arising out of error detection and correction during the later stages of the project (Abdel-Hamid and Madnick 1991). This additionally has implications on the success of the project.

We adopt the system dynamics approach and carry out our investigation on a validated software project dynamic model, tailored to a specific project development context. Results based on experimentation with two contrasting patterns of requirement volatility (discussed later) depict variations in QA effectiveness under usage of different resource allocation policies employed. This assumes the first step towards a more holistic investigation, taking into account the different requirement volatility patterns that can be observed in reality.

The next section presents the theoretical background of our work. The subsequent section describes the methodology we have adopted in this study. Study results are then presented and discussed. Finally, in conclusion, we summarize the findings of this investigation and delineate the next steps. The term 'project quality' has been used uniformly in the paper to refer to the quality of the product or service that is finally delivered to the users.

Theoretical Background

Project contingency theory derived from the classical organizational contingency theory (Burns and Stalker 1961) argues that the best approach to managing a project depends on context, and project performance is related to the extent of fitness between project characteristics and project management. With the focus on projects as a temporary organization (Packendorff 1995), the NTPC diamond framework (Shenhar and Dvir 2007) suggests a context-free distinction in four dimensions i.e. novelty (related to product/service newness to customers and users), technology (related to newness of the technology used), complexity (related to product location on a system-subsystem hierarchy), and pace (related to urgency and available timeframe), in order to facilitate project management. The novelty
dimension relates to requirement volatility addressed in this paper. The extent of product/service
newness to market influences the level of market research in identifying user preferences, thereby
contributing to requirement volatility.

Classifying projects based on the four dimensions in order to prescribe project management approaches
has three significant shortcomings. First, its organizational scope is too narrow and the project outcome
might be contingent more on the soft factors influencing a project rather than by use of any project
management approach (Engwall 2003). Second, the four axes in NTPC diamond framework indicate
 corresponding changes in terms of magnitude. Changes can also take place in terms of patterns (with
magnitude constant). Third, project management encompasses a broader range of tasks like allocating
resources, planning, assessing risks, selecting management style, processes and tools, with multiple ways
of accomplishing each of these tasks (Hughes and Cotterell 2009). The second limitation further suggests
that the choice of these tasks is likely to be affected not only by the positioning of a project on the NTPC
diamond framework, but also by the pattern of influence in each dimension.

In order to better appreciate the shortcomings, we focus on a single dimension of the framework (i.e.
novelty), and assess the influence of a specific project management task (resource allocation), and the
resulting effect on an aspect of project performance (quality). Pattern-wise change along novelty
dimension is achieved by considering two contrasting requirement volatility patterns addressed later. A
review of relevant literature of each is provided below.

**Requirement Volatility**

Studies on software requirement volatility have mostly taken an event oriented viewpoint in
understanding its nature and source (Barry et al. 2006; Costello and Liu 1995; Nidumolu 1996),
comprehending its effects on project performance parameters (Ferreira et al. 2009; Houston et al. 2001;
Jiang et al. 2009), and investigating mitigation strategies (Maruping et al. 2009; Thakurta and Ahlemann
2010). Evidences from literature suggest contrasting effects on project performance parameters arising
out of change order generation following different patterns (Thakurta et al. 2009). We expect that the
project management approach will be contingent on such pattern-wise variation, thereby affecting project
performance.

**Resource Allocation**

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 (Otero et al. 2009). Authors have
suggested different resource allocation policies with optimal effect on project performance like variation
of resource adjustment times (Lee et al. 2007), use of proportional and foresighted resource forecasting
techniques (Joglekar and Ford 2005), altering resource allocation order to project tasks (Black and
Repenning 2001), overstaffing the project from the onset (Collofello et al. 1998), and keeping the level of
workforce constant (Collofello et al. 1998). The choice of these policies was found to be based on
“heuristic knowledge, subjective perception, and instinct” (Acuña et al. 2006). Accordingly, an
investigation of the efficacy of each in different project contexts can act a rule-of-thumb, facilitating
improved resource allocation decisions.

**Quality**

Going by the project contingency theory, we select quality as the project performance parameter, and
study the impact of different resource allocation policies on project QA. Variation in quality levels is
expected to influence customer satisfaction levels, a key determinant of project success (Shenhar and Dvir
2007). Instead of focusing on different quality improvement approaches which has been the focus of QA
literature (Basili and Rombach 1987; Leszak et al. 2000; Li et al. 2010), here we investigate how choice of
a specific project management task (different resource allocation policies) influences the process, thereby
affecting the quality levels.
Methodology

Task Environment

Project management is a complex and dynamic phenomenon involving interplay of a wide range of hard and soft factors (Crawford and Pollack 2004) which prompted us to use the system dynamics (SD) (Sterman 2000) approach in our study. The basic premise in SD is that system behavior 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. 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 (Royce, 1987). The choice of waterfall methodology was driven out of its observed predominance even in projects endangered because of requirements volatility (Thakurta and Ahlemann 2010). We implement the same using Abdel-Hamid’s (Abdel-Hamid and Madnick 1991) SD model that is based on the 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 extensively validated based on case studies conducted by the author, and supplemented by expert review techniques (Abdel-Hamid and Madnick 1991). The model parameters (Table 1) were set as per the TRW Inc. case study (Abdel-Hamid and Madnick 1991), which matches our project context. Also, the advantage of working with real project data are that the values can be trusted and it further allows for comparison with the actual results. The value of project average full-time-equivalent (FTE) professionals, derived using COCOMO (Constructive Cost Model, Boehm et al. 1995) 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.

Abdel-Hamid’s SD model uses a factor ‘task underestimation fraction’ that captures fraction of undiscovered tasks that get added to the project scope. Effort allocation to QA gets adjusted, having based on the project schedule pressure. However 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. A description of the behavior of the causal loop diagram is provided below, with the model parameters shown in italics.

Requirement volatility 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, (Boehm et al. 1995) also increases. This increased effort requirement positively affects the schedule pressure, and leads to generation of more errors because of higher error generation rate. With increase in schedule pressure, some readjustment in the software team engaged in the project is expected to take place. In order to meet the agreed upon delivery schedule and keep the project costs under control, the project managers facing schedule pressure might give more priority to development related activities compared to QA. In the process, they might completely abandon or do some curtailment in the team assigned for QA (Abdel-Hamid and Madnick 1991). Thus, under the circumstances, some reduction in percentage of workforce allocation to QA takes place. High error generation rate and reduced QA

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Specified Job Size</td>
<td>1067 Function Point</td>
</tr>
<tr>
<td>Initial Estimated Effort</td>
<td>3594 Person-Days</td>
</tr>
<tr>
<td>Initial Schedule Estimate</td>
<td>348 Days</td>
</tr>
</tbody>
</table>
manpower in turn negatively impacts fraction of errors detected, thereby hampering QA effectiveness. The increased effort requirement (effort still needed) arising because of requirement volatility 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 QA. 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. Under the circumstances and with availability of a larger workforce, percentage of workforce allocation to QA also increases. The net result is an increase in QA effectiveness.

The dynamics is further completed by the pattern in which change orders are generated during project development (i.e. requirement volatility 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.

**Experiment Design**

For the purpose of our experimentation, we approximated two contrasting patterns of requirement volatility experienced in real project setting in the following ways:

**Linear Decay**: We approximated the exponential decay pattern of change order generation rate (Zowghi and Nurmuliani 2002) by the linear decay pattern. Here, initial high rate of change order generation decreases linearly with time. Close collaboration with users generates high rate of requirements change initially. With time the requirements gradually stabilize and the rate of change declines.

**Linear Rise**: The exponential rise pattern of change order generation rate (Abdel-Hamid and Madnick 1991) was approximated using the linear rise pattern in this case. Here the rate of change order generation increases linearly with time. Users’ and developers’ learning curves make project tasks grow at an increasing rate.

We experimented with the following resource allocation policies discussed in the literature to investigate their effect on the QA activity.

**POLICY 1**: Controlling the level of workforce over the development period
This can be visualized as project management, trying to maintain the level of workforce at some desired value (Collofello et al. 1998). The situation can arise, for example, when the project is executed in a ‘fixed price’ contract and the management is unwilling to change the workforce level. Here we considered two scenarios: scenario A with the level of desired workforce estimated based on initial project scope, and scenario B with desired workforce level estimated based on the hunch of the size of additional tasks arising out of change order generation.

POLICY 2: Overstaffing the project from the start

Here the project maintains additional bench strength based on their expectation of requirement volatility during project development. Usage of this overstaffing strategy can be noticed in Collofello et al. (1998). Projects which have high business impact or face huge time constraint can employ this strategy. For experimentation purpose, we implement overstaffing by setting the value of starting workforce equal to twice the average full time equivalent (FTE) workforce as given in Table 1. This would provide us with an understanding of how this policy influences project dynamics under the described experimental settings.

POLICY 3: Appropriately managing the resource allocation delays

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 (Lee et al. 2007). Reduction in hiring delays is possible through pre-hiring of desired competency. Two scenarios were considered, the first having resource allocation delay of 5 days (scenario A), and the second having delay of 75 days (scenario B). These two scenarios are considered to be representatives of minimum hiring delay and maximum hiring delay based on evidences cited in Lee et al. (2007).

POLICY 4: Using resource allocation strategy based on forecasting techniques depending upon requirement change expectations

The assumption we make here is that the project managers, based on previous experiences or based on data of previous occurrences, have guessed the pattern in which the requirement is expected to change in their project, and have planned resource deployment accordingly. Based on Joglekar and Ford (2005), here we adopted a proportional forecasting policy where the hiring rate is adjusted in an identical fashion as the rate of change order generation in the project.

Table 2 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>
<thead>
<tr>
<th>Policy #</th>
<th>Parameter Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Scenario A: Desired Workforce Level = 10.3; Scenario B: Desired Workforce Level = 15.8</td>
</tr>
<tr>
<td>2.</td>
<td>Starting Workforce Level = 20.6 (Twice the Average FTE, refer to Table 1)</td>
</tr>
<tr>
<td>3.</td>
<td>Scenario A: Hiring Delay = 5 Days; Scenario B: Hiring Delay = 75 Days</td>
</tr>
<tr>
<td>4.</td>
<td>Forecasted Hiring Rate ~ Change Order Generation Rate</td>
</tr>
</tbody>
</table>

In order to carry out simulation, the cause-and-effect model (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 (Sterman 2000) was implemented using commercially available iThink software. The software allows one to simulate business processes and scenarios, and in the process carry out what-if analysis and derive managerial implications. 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. The task underestimation fraction is set at 0.67 implying that the initial project size can grow by 50% during project development because of requirement volatility. The growth of

---

2 Available at: http://www.iseesystems.com/softwares/Business/IthinkSoftware.aspx
project tasks under the two different requirement volatility patterns (i.e. linear decay 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. Corresponding to these two requirement volatility patterns, seven simulation runs were conducted as the column headers in Tables 3 and 4 provided in the next section indicate. Six of these runs correspond to the policies and scenarios (two scenarios for Policy 1 and two scenarios for Policy 3) listed in Table 2. At the end of each run, the parameter values (row entries in Tables 3 and 4) were noted down, and then all the values were reset to ‘Base’ case as defined above. The next section elaborates on the results.

Results

First, let’s consider the linear rise pattern of requirement volatility. Table 3 show comparison of project performance for ‘Base’ and the different policies as discussed above (Table 2). In all cases a total of 1592 tasks were processed (i.e. 50% above the initial specified as given in Table 1). QA effectiveness was measured at the end of each simulation run by dividing the number of errors detected through QA activity by the total person-days effort expended on QA activity.

Results indicate variation in parameters based on the policy used. QA effort could be observed to vary between 40% decrease (Policy 2: Starting Workforce Level = 20.6) and 15% increase (Policy 4: Forecasting Hiring Rate) compared to the ‘Base’ case. Error generation and detection were highest under Policy 3: Scenario A (Hiring Delay = 5 Days), and lowest under Policy 1: Scenario A (Desired Workforce Level = 10.3). QA effectiveness could be observed to be highest (42% increase) when Policy 2 was used, and lowest (13% decrease) under Policy 4. To understand these variations let’s compare results of Policy 2 and Policy 4.

Policy 4 uses the forecasting technique in order to adjust the project workforce depending upon the change order generation rate. The linear rise pattern of change order generation results in a progressive increase of workforce (Figure 4 (a)). The rookies’ coming in causes some decrease in productivity during the initial stages (Figure 4 (b)). With time, delays are perceived in project progress arising out of productivity losses. This does not lead to hiring, since in this scenario hiring is not driven by the project status. In the absence of finite schedule completion limit, the schedule pressure also does not increase (not shown). Hence the QA activity is not curtailed and continues as long as task remains pending for QA. This longer QA duration results in the QA effort expended to be higher compared to ‘Base’ (Table 3). Error detection is affected by both the pool of errors present and the productivity. In the absence of late hiring, at the final stages of the project, the workforce productivity gets hampered owing to exhaustion. This causes the error detection rate to decline towards the end, resulting in identification of about 1669 errors which is equivalent to the ‘Base’ result (Table 3).
Compared to this, in case of Policy 2, the project starts with a higher number of workforce (Figure 4 (a)). In the absence of upfront hiring needs, the productivity depicts an increasing trend over the initial period, minor variations being caused by communication related losses (Figure 4 (b)). Because of this, tasks get processed and assigned for QA at a relatively higher rate. High productivity of the project development team also causes the error generation rate to be low (not shown). With progress, as project delays become visible, hiring takes place. Hiring augments the workforce size and in turn the development and the QA process, despite productivity reductions because of additional training overheads. All these ensure an early completion of the project (Table 3). The shorter duration of the QA phase substantially reduces the QA effort expenditure. Low error generation rate and the assigned quality objective (75%) also lead to number of errors detected to be lower in this case (Table 3).

Table 4 presents the corresponding results for the ‘Base’ case, and when the different policies were used. Results indicate use of Policy 1: Scenario B (Desired Workforce Level = 15.8) to be the most effective (21% improvement over ‘Base’) even though maximum number of errors get detected under usage of Policy 3: Scenario A (Hiring Delay = 5 Days). In case of Policy 1: Scenario B, the project workforce is maintained constant at a high level (Table 2) which enables higher productivity. With change order generation taking place during the initial stages, the presence of a larger workforce does not allow the schedule pressure to build up. This ensures the QA process to go on as planned, and the QA effort expended is low owing to higher productivity of project workforce. Overall this leads to greater QA effectiveness under this policy.
Conclusion

Based on the simulation results we can conclude that the effectiveness of the QA process (a surrogate of project performance in our study) is contingent upon the pattern of change order generation and the resource allocation policies adopted. Given the project setting, overstaffing led to best results under linear rise pattern while working with a constant workforce emerged as the best choice under linear decay. The findings illustrate the shortcomings of the project contingency theory by positing that the same performance criteria might require different project management approaches, contingent on pattern of change in any of the dimensions of the NTPC diamond framework. The results also contradict with findings suggesting negative influence of overstaffing on project performance arising out of social loafing behavior (Alnuaimi et al. 2010; Srinivasan et al. 2010). The favourable results obtained in our study can be explained by the fact that overstaffing is expected to lead to uniform distribution of increased work load among the team members under requirement volatility. Thus, in the presence of the increased work responsibility, individual tendencies to engage in social loafing are expected to reduce.

The study is not without limitations. The extent of variations in parameters across the policy choices was not very significant given the fact that the project did not have any imposed schedule penalty. Usage of other resource allocation policies not considered in this research might contribute towards greater QA effectiveness. To proceed further, the authors intend to perform experimentation with other requirement volatility patterns encountered in practice, and further carry out investigation in other project contexts.

Our study is targeted at both researchers and practitioners. The study differs from several published studies on requirement volatility by taking a pattern oriented viewpoint of the phenomenon. The contribution to theory is a need to relook at the project contingency theory and contemplate refinements by incorporating pattern of change in the framework. From a practical point of view, the research informs the need to adopt contextual project management practices depending upon the project settings. If based on prior experience or supported by data of similar projects, a project manager is able to anticipate the pattern in which requirements are expected to change during project development; the study results serve as a guidance to what resource allocation policy to be adopted giving project priorities. The simulation model itself can act a decision support system allowing project managers to perform what-if analysis in order to understand the effect of requirement volatility on project quality and performance levels. In the present context of shorter time-to-market and stringent quality requirements, it would be interesting to see how our initial findings and the proposed work extension is able to influence the design of resource augmentation policies in organizations.

References


