

Journal of Information Technology Management

ISSN #1042-1319

A Publication of the Association of Management

RESOURCE MANAGEMENT EFFICACY AT SPECIFIED QUALITY LEVELS – AN INVESTIGATION UNDER LINEAR RISE AND TRIANGULAR REQUIREMENT VOLATILITY PATTERNS

RAHUL THAKURTA

XAVIER INSTITUTE OF MANAGEMENT BHUBANESWAR

rahul@ximb.ac.in

ABSTRACT

Effect of requirement volatility measured with respect of magnitude on project parameters like schedule, effort, etc has been well investigated. However, there are insufficient evidences on how occurrence of requirement volatility following different patterns can influence project management practices and project performance. To investigate the same, here using system dynamics, we study the impact of different resource allocation strategies on project quality under two experimental patterns of requirement volatility. Findings indicate variation in quality metrics depending upon the experimental scenario, thereby suggesting the need to adopt pattern-dependent project management practices.

Keywords: requirement volatility, quality assurance, resource management, system dynamics.

INTRODUCTION

Software projects have been plagued with problems of cost and schedule overruns since its inception. Requirement volatility which refers to the quantitative measure of the changes in requirements in terms of additions, deletions, and modifications during project progress has been hitherto blamed for these debacles. Evidences also cite requirement volatility to be significant risk influencing project performance [1, 2, 3].

Studies on requirement volatility have mostly focused on how the magnitude of requirement changes impact software development [4]. However, evidences indicate that requirement volatility can also take place following different patterns given the same magnitude [5]; with patterns referring to the various geometric shapes of requirements generation. It seems reasonable to assume that the different requirement volatility patterns will influence the efficacies of the project management approaches adopted.

This paper tries to evince the above assumption with the help of simulations carried out on an established system dynamics model of software project management [6]. By considering two different requirements volatility patterns viz. linear rise and triangular, we demonstrate the efficacies of some resource allocation policies as project management tasks on project quality. The study focused on the quality assurance (QA) activity, with quality being measured with the help of the metric QA effectiveness defined as the ratio of number of errors detected and QA effort expended. A low value of QA effectiveness indicates 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 [6].

This paper is organized as follows. The next section identifies some studies related to the work. Subsequently, the methodology adopted in our study is elaborated. The study results are presented next and analyzed. Finally, the last section concludes the work with a mention of future research opportunities. We use the term

‘project quality’ uniformly in the paper to refer to the quality of the product or service that is finally delivered to the users.

RELATED WORK

Research on software requirement volatility has mainly addressed its measurement, cause-effect, and mitigation strategies. It has been observed that the variation in requirements occur both in terms of magnitude [7, 8], and in terms of patterns [5]. Some of the factors that have been identified as antecedents to requirement volatility are requirement errors and inconsistencies [9]; users’ and developers’ increased understanding of the project [10]; technical, schedule or cost related problems [11] and selection of systems development life cycle or SDLC [12]. The primary impact of requirements volatility has been acknowledged as degradation in project quality [13]. In terms of requirement volatility management, emphasis has been given on development of change order control board [14], prudent selection of SDLCs [15]; and use of techniques line base-lining [16], joint application design (JAD) [14], etc. There is however no evidence of how pattern-wise occurrence of requirement volatility can be effectively managed.

Literature on resource allocation in software projects is limited. This could be because every software project is considered to be a unique scenario [17]. The available literature discusses different resource allocation policies with optimal effect on project performance like use of proportional and foresighted resource forecasting techniques [18], altering resource allocation order to project tasks [19], overstaffing the project from the onset [20], and keeping the level of workforce constant [20].

We focus on project QA activity given its obvious importance. Studies on software QA have primarily focused on the different quality improvement approaches to enhance project quality. For example, Basili and Rombach [21] provide a five step methodology for software process improvement based on analysis of defect related data. Liu et al. [22] present an approach of integrating formal specification, review, and testing activities with a view to remove errors and identify missing requirements. Wagner et al. [23] presents 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. [24] investigate the effectiveness of three

types of QA activities (viz. review, process audit, and testing) and their overall contribution to QA Return on Investment (ROI). The current study looks into how different resource allocation alternatives affects the QA process, and hence assumes a different lens on the phenomenon.

METHODOLOGY

We used the system dynamics (SD) [25] approach to carry out the study, driven by the fact that project management involves a dynamic interplay of a wide range of hard and soft factors [26]. System dynamics uses feedback structures to model a problem in order to understand the behaviour. Model building starts with developing a causal loop diagram consisting of a collection of causal links, each having a certain polarity. The causal loop diagram represents the hypothesis concerning the problem of interest. The causal loop graph can be subsequently mapped to a mathematical model consisting of a system of difference equations, which can be simulated under different parametric conditions.

In order to carry out the study, we adopted the project dynamics model of Abdel-Hamid [6] that implemented the waterfall SDLC. The rationale behind basing our experimentation on waterfall SDLC was its observed predominance even in the current context [15]. The model integrated all the relevant processes of software development like development, quality assurance, testing, rework, etc thereby allowing experimentation by changing resource levels, project size and observing the resultant effects. The model was extensively validated based on several case studies so as to increase confidence of the reader on the model results [6].

The model uses a factor ‘Task Underestimation Fraction’ that captures fraction of undiscovered tasks that get added to the project scope, and is a measure of the magnitude of requirement volatility during project development. QA activity is given more precedence over development during project duration. The effort allocation to QA gets adjusted based on the project schedule pressure. The model however does not impose any cap on maximum allowable delay of project schedule. Figure 1 represents the causal loop diagram of the problem embodied in the model structure. A description of the behavior of the causal loop diagram is provided below, with the model parameters shown in italics.

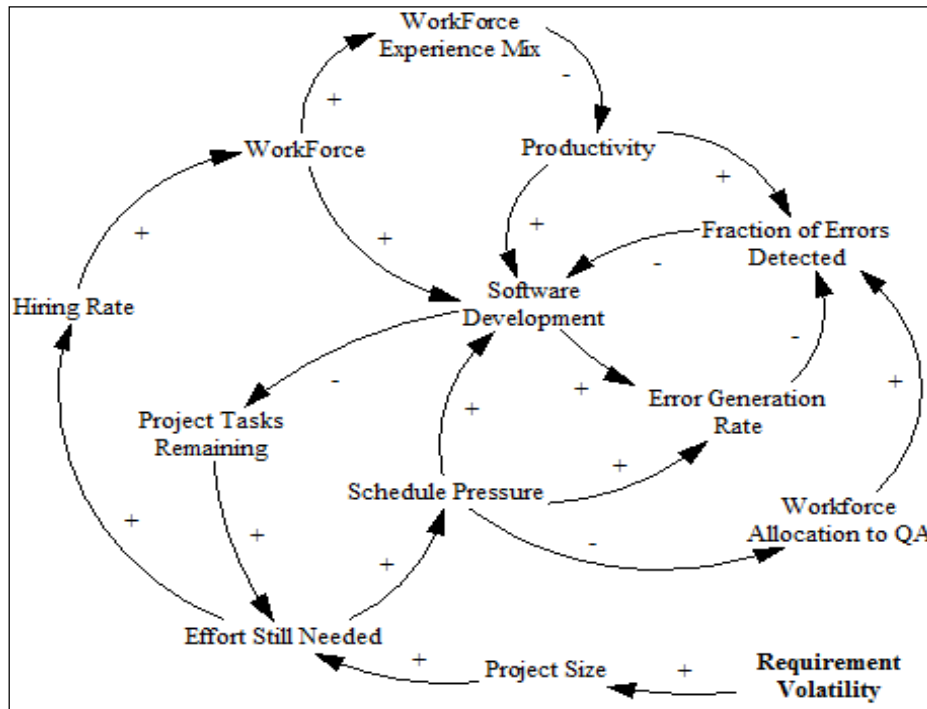


Figure 1: Model Causal Loop Diagram

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 [27], 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 [6]. Thus, under the circumstances, some reduction in percentage of *workforce allocation to QA* takes place. High *error generation rate* and reduced QA 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. However further complexity is introduced by the pattern of requirement volatility and the chosen resource allocation policy. 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 chosen for experimentation are based on the TRW Inc. case study [6]. The reported project is medium sized, having initial specified job size as 64,000 delivered source instructions (DSI) which corresponds to 1067 function points (FP). The initial estimates of effort and schedule were derived using COCOMO ('constructive cost model': [27]) is provided in Table 1.

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 last entry in the table indicates a fractional value for the project average full-time-equivalent (FTE) professionals. The value can be interpreted as 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.

For experimentation purpose, we focused on the following two requirement volatility patterns which approximates some real project scenarios

- **Linear Rise:** The exponential rise pattern of change order generation rate provided in a case study [6] was approximated using the linear rise pattern in this case. Here the rate of change order generation increases linearly with time, driven by users' and developers' learning curves.
- **Triangular Variation:** Project tasks grow initially at an increasing rate as developers learn about the domain. But with few requirements left to be identified, the rate of requirements change drops towards the latter stages of the project.

We experimented with the following workforce management policies to investigate their effect on the QA activity:

- **POLICY 1:** Controlling the level of workforce over the development period
This can arise when a project manager tries to maintain the level of workforce at some desired value [20]. Projects having a 'fixed price' contract can often encounter such a scenario. Here, for experimentation, we assume that the estimate of the desired workforce is arrived upon based on informed guess of the size of additional tasks arising out of change order generation.
- **POLICY 2:** Overstaffing the project from the start
Here there are reserves available with the project in expectation of occurrence of re-

quirement volatility during project development. Projects having high business criticality or time constraint can employ this strategy. For experimentation purpose, here we implement overstaffing by setting the value of starting workforce equal to twice the average full time equivalent (FTE) as given in Table 1. Use of overstaffing strategy can be noted in Reference [20].

- **POLICY 3:** Using resource allocation strategy based on forecasting techniques depending upon requirement change expectations

The policy assumes that project managers have a hunch of the expected pattern of requirement change according to which they have planned resource deployment in their projects. For experimentation purpose, 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. This forecasting technique has been used in Joglekar and Ford [18].

- **POLICY 4:** Altering resource allocation depending upon priority ordering of tasks

Here we alter the priority ordering of various tasks. In the model, allocation of resources to project tasks takes place following the order: 'Training → QA → Rework → Software Production (Development/Test)'. This indicates that QA and rework are given higher priorities compared to project development activities. While many other possible resource allocation schemes exist, here we test the efficacy of the following resource allocation scheme: Training → Development → QA → Rework → Testing. In this case, we assign higher priority to project development activities, followed by QA, rework and testing. We want to test how this policy performs under the different requirement volatility patterns.

The parameter values affected by implementation of the stated policies are listed in Table 2. The other model parameters assume the 'Base' case values (i.e. the behaviour as depicted by the model structure without implementation of any of the resource allocation policies). In case of Policy 4, the parameter values are all identical to the 'Base' case; only the workforce allocation sequence was altered.

Table 2: Parameter Changes for Policy Implementation

Policy #	Parameter Values
1.	Desired Workforce Level = 15.8
2.	Starting Workforce Level = 20.6 (Twice the Average FTE, refer to Table 1)
3.	Forecasted Hiring Rate ~ Change Order Generation Rate
4.	-

Model simulation was achieved by developing the executable version of the model, also known as the stock and flow diagram [25] using the iThink1 software. 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 arising out of requirement volatility. The growth of project tasks under linear rise and triangular requirement volatility patterns (Figure 2) is shown in Figure 3. Results are provided in the next section.

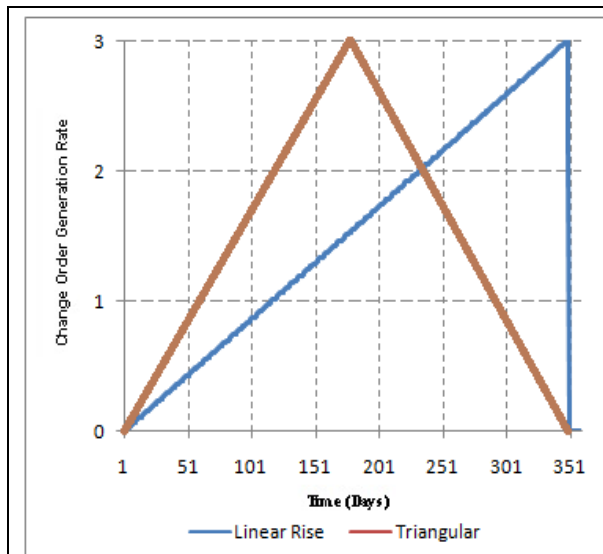


Figure 2: Change Order Generation Rates

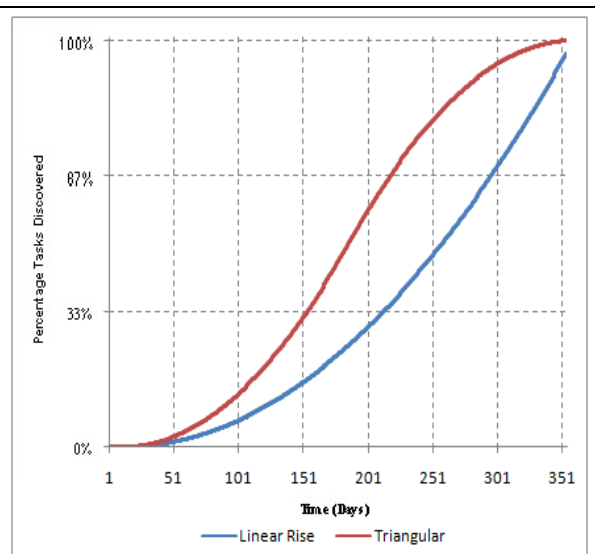


Figure 3: Growth of Project Tasks

¹ Available at:
<http://www.iseesystems.com/software/Business/IthinkSoftware.aspx>

RESULTS

Table 3 provides a comparison of project performance for ‘Base’ and the different policies as discussed in Table 2 for the linear rise pattern of requirement

volatility. The values in each cell in Table 3 indicate the simulation result and a percentage (%) figure. The percentage figure indicates where the values of each parameter stand with respect to the ‘Base’ (taken as 100%) corresponding to the different policies used. In all cases a total of 1592 tasks were processed.

Table 3: Effect of Different Policies under Linear Rise

	Base	Policy 1	Policy 2	Policy 3	Policy 4
QA Effort (Person-Days)	15572 (100%)	11632 (75%)	9294 (60%)	17973 (115%)	3677 (24%)
Rework Effort (Person-Days)	1883 (100%)	1403 (75%)	1472 (78%)	2446 (130%)	149 (8%)
Completion Date (Days)	789 (100%)	1262 (160%)	601 (76%)	1015 (129%)	815 (103%)
FTE Manpower (Person)	31.5 (100%)	14.6 (46%)	26.5 (84%)	28.9 (92%)	11.3 (36%)
No. of Errors Generated	2053 (100%)	1898 (92%)	1824 (89%)	2079 (101%)	1983 (97%)
No. of Errors Detected	1662 (100%)	1503 (90%)	1406 (85%)	1669 (100%)	294 (18%)
QA Effectiveness (No. of Errors Detected/Person-Days)	0.11 (100%)	0.13 (118%)	0.15 (136%)	0.09 (82%)	0.08 (73%)

Table results indicate usage of Policy 2 to be the most effective towards achieving maximum QA effectiveness. To interpret the results, let’s compare simulation runs of Policy 2 and Policy 3 (maximum QA effort expenditure). Policy 3 is characterized by usage of the forecasting technique with project workforce adjustment driven by 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 cause some decrease in productivity during the initial stages (Figure 4 (b)). With time, there are perceived delays in project progress arising out of productivity losses. The delays doesn’t trigger hiring as here hiring is not driven by the project status. The schedule pressure also does not increase in absence of a finite schedule completion limit. Hence the QA activity is not curtailed and continues as long as task remains pending for QA. The elongation of the QA duration contributes to higher QA effort expenditure in comparison to ‘Base’ (Table 3). Error detection is affected by both the pool of errors present and

the productivity. In absence of late hiring, at the final stages of the project, exhaustion results in deterioration of the workforce productivity. This causes the error detection rate to decline towards the end (not shown).

In comparison, Policy 2 uses a higher workforce to start with (Figure 4 (a)). In absence of upfront hiring needs, the productivity depicts an increasing trend over the initial period (Figure 4 (b)). This ensures that tasks get processed and assigned for QA at a relatively higher rate. High productivity also causes the error generation rate to be low (not shown). As visibility increases with project progress, the delays in project schedule become apparent. This triggers off hiring process, which in turn results in augmentation of workforce size. Presence of a higher workforce ensures 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).

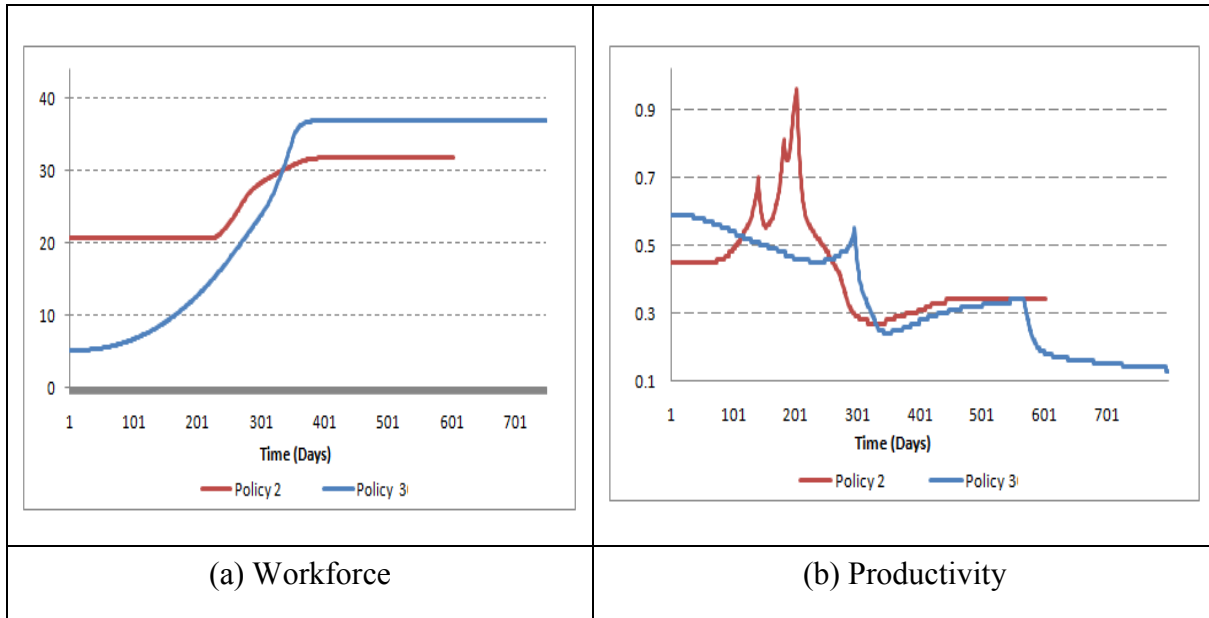


Figure 4: Model Parameter Variation Subjected to Linear Rise Pattern of Change Order Generation

Do we expect the results to differ when change order generation follows triangular pattern? The results corresponding to the simulation run pertaining to the different policies is included in Table 4. The effectiveness of QA activity could be observed to be the highest under Policy 1 (Table 4). Driven by a larger workforce (Policy

1), the schedule pressure was maintained at a lower level during the project initial stages. This ensured QA process to be executed as planned, and error detection was facilitated because of high productivity of the project workforce.

Table 4: Effect of Different Policies under Triangular Pattern

	Base	Policy 1	Policy 2	Policy 3	Policy 4
QA Effort (Person-Days)	14697 (100%)	11021 (75%)	11817 (80%)	14857 (101%)	3244 (22%)
Rework Effort (Person-Days)	1843 (100%)	1404 (76%)	1608 (87%)	1948 (106%)	128 (7%)
Completion Date (Days)	777 (100%)	1192 (153%)	584 (75%)	1120 (144%)	669 (86%)
FTE Manpower (Person)	30.4 (100%)	14.8 (49%)	33.3 (110%)	21.4 (70%)	13.7 (45%)
No. of Errors Generated	2058 (100%)	1908 (93%)	1844 (90%)	2071 (101%)	2048 (100%)
No. of Errors Detected	1665 (100%)	1512 (91%)	1432 (86%)	1658 (100%)	208 (12%)
QA Effectiveness (No. of Errors Detected/Person-Days)	0.11 (100%)	0.14 (127%)	0.12 (109%)	0.11 (100%)	0.06 (55%)

Results corresponding to Policy 4 indicate minimum value of QA effectiveness for all the experimental cases, and needs to be mentioned here. Policy 4 uses a resource allocation scheme following the order: Training → Development → QA → Rework → Testing. With task augmentation occurring in various ways depending upon the pattern of requirement volatility, the higher priority of the development activity in this case ensured greater workforce allocation to development and lesser to QA. This propelled the task processing rate, and in turn contributed to faster project completion and higher workforce utilization. QA activity however suffered, which gets reflected by the low QA effort expenditure and low error detection, as the table results indicate.

CONCLUSION

The results indicate the pattern of change order generation and the resource allocation policies adopted influence the effectiveness of the QA process. Corresponding to the experimental scenarios, overstaffing led to the best results under linear rise pattern but it was not as effective corresponding to the triangular pattern.

The study has limitations that need to be considered. In absence of any imposed schedule penalty, the variations in project parameters across the policy choices were not very high. The resource management policies not considered in the study can also contribute to greater QA effectiveness. Further, project characteristics like project size, project development methodology etc can contribute towards variation in results. These limitations withstanding, the results does call for a need to adopt contextual management approaches depending upon the expectation of change order generation in projects. Otherwise, the end result could be degradation of project quality or excess cost of the project, both of which has implications for project success.

In continuation of the research, future work is expected to analyze the effect of resource management policy on the total effort expended, as this ultimately translates as cost to the project organization. Impacts of additional constraints like competency of available workforce, cost penalty, schedule penalty etc on the results can be looked into. Experimentation with other possible change order generation patterns can also uncover additional dynamics associated with the process. Our research is expected to lead to many more exchanges that contribute towards project objectives in a meaningful way.

REFERENCES

[1] Boehm, B. "Software risk management: principles and practices," *IEEE Software*, 1991.

- [2] Hoorn, J.F. and Konijn, E. A. "Requirements change: fears dictate the must haves; desires the won't haves," *Journal of Systems and Software*, Volume 80, Number 3, pp. 328-355, 2007.
- [3] 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, pp. 570-598, 2007.
- [4] 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, pp. 1568-1577, 2009.
- [5] 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.
- [6] Abdel-Hamid, T.K. and Madnick, S.E. *Software Project Dynamics: An Integrated Approach*, Prentice Hall (Englewood Cliffs), 1991.
- [7] 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, pp. 448-464, 2006.
- [8] Costello, R.J. and Liu, D. "Metrics for requirements engineering," *Journal of Systems and Software*, Volume 29, Number 1, pp. 39-63, 1995.
- [9] Nurmaliani, 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.
- [10] Davis, A.M., Nurmaliani, N., Park, S., and Zowghi, D. "Requirements change: what's the alternative?" *Proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference*, July 2008, pp. 635-638.
- [11] Kontonya, G. and Sommerville, I. *Requirements Engineering Process and Techniques*. Wiley Publications Reprinted, February 2002.
- [12] 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.
- [13] Zowghi, D. and Nurmaliani, 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.
- [14] Jones, C. *Estimating Software Costs*. McGraw Hill, 1998.
- [15] 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.
- [16] Wiegers, K. *Software Requirements*, Microsoft Press, 1999.
- [17] 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.
- [18] Joglekar, N.R. and Ford, D.N. "Product development resource allocation with foresight," *European Journal of Operational Research*, Volume 160, pp. 72-87, 2005.
- [19] Black, L.J. and Repenning, N.P. "Why firefighting is never enough: preserving high-quality product development," *System Dynamics Review*, Volume 17, Number 1, pp. 33-62, 2001.
- [20] 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.
- [21] Basili, V.R. and Rombach, H.D. "Tailoring the software process to project goals and environments," *Proceedings of the 9th International Conference in Software Engineering*, California, USA, 1987.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] Sterman, J.D. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin/McGraw-Hill, 2000.
- [26] Crawford, L. and Pollack, J. "Hard and soft projects: a framework for analysis," *International Journal of Project Management*, Volume 22, Number 8, pp. 645–653, 2004.
- [27] 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, pp. 57-94, 1995.

AUTHOR BIOGRAPHY

Rahul Thakurta is an Assistant Professor of Information Systems at Xavier Institute of Management Bhubaneswar (XIMB), 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 Vilakshan journals published by XIMB.