Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 14 Issue 2 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Stochastically Simulating the Effects of Requirements Creep on Software Development Risk Management

By P. K. Suri & Shilpa Rani

Kurukshetra University, India

Abstract- One of the major chronic problems in software development is the fact that application requirements are almost never stable and fixed. Creeping user requirements have been troublesome since the software industry began. Several empirical studies have reported that volatile requirements are a challenging factor in most information systems development projects. Software process simulation modeling has increasingly been used for a variety of issues during software development. The management of software development risks is one of them. This study presents an approach for simulating and analyzing the effect of Requirements Creep on certain software development risk management activities. The proposed algorithm is based on stochastic simulation and has been implemented using C.

Keywords: requirements creep, requirement volatility, requirement management, stochastic simulation, software risk management.

GJCST-C Classification: K.6.3

Strictly as per the compliance and regulations of:

Stochastically Simulating the Effects of Requirements Creep on Software Development Risk Management

P.K. Suri [α] & Shilpa Rani [σ]

Abstract- One of the major chronic problems in software development is the fact that application requirements are almost never stable and fixed. Creeping user requirements have been troublesome since the software industry began. Several empirical studies have reported that volatile requirements are a challenging factor in most information systems development projects. Software process simulation modeling has increasingly been used for a variety of issues during software development. The management of software development risks is one of them. This study presents an approach for simulating and analyzing the effect of Requirements Creep on certain software development risk management activities. The proposed algorithm is based on stochastic simulation and has been implemented using C.

Keywords: requirements creep, requirement volatility, requirement management, stochastic simulation, software risk management.

I. Introduction

Software process simulation modeling is increasingly being used to address variety of issues from the strategic management of software development, to supporting process improvements, to software project management training. One of the proposed purposes for software process simulation is the management of software development risks, usually discussed within the category of project management [1]. There have been various (but quite a limited) studies which have used modeling and simulation for software development risk management for example: Madachy's Model [2], Houston's Model [3]. The present study also describes an approach for managing software development risks using simulation.

In the present work, implementation of a simulator has been done for modeling the effects of Requirements Creep on various risk management factors during software development using stochastic simulation.

This paper has been has been organized into various sections including the present one. An overview of software development risk factors has been provided in section II while 'requirements' as a major risk factor during software development have been discussed in section III, followed by potential effects of requirements creep(section IV). The proposed algorithm has been provided in section V, the results of which have been demonstrated and interpreted in section VI with the help of charts representing the relationships between various risk management factors.

II. Risk Factors during Software Development

Top 10 software risk items identified by Boehm [4] for software development projects:

- Personnel shortfalls
- Unrealistic schedules and budgets
- Developing the wrong functions and properties
- Developing the wrong user interface
- Gold plating(adding more functionality/ features than is necessary)
- Continuing stream of requirements changes
- Shortfall in externally furnished components
- Shortfalls in externally performed tasks
- Real-Time performance shortfalls
- Straining computer-science capabilities

Jones [5] has presented the following three key software areas:

- Risks associated with inaccurate estimating and schedule planning
- Risks associated with incorrect and optimistic status reporting
- Risks associated with external pressures, which damage software projects.

Some investigators have even presented software development risks on the order of 150 or more. Twenty nine of these risk factors have been cited by Houston [3] as most important Software development risk factors.

III. Requirements: A Major Software Development Risk Area

A requirement is the condition or capacity that a system that is being developed must satisfy [6]. Requirement management in general is mainly concerned with three tasks: Requirement Elicitation, Requirement Analysis and Requirement specification.

Author α: Dean (R&D), Chairman & Professor (CSE/IT/MCA), H.C.T.C.M., Technical Campus, Kaithal, Haryana, India.
e-mail: pksurikuk@gmail.com
Author σ: Research Scholar, Department of Computer Science & Applications, Kurukshetra University, Kurukshetra, Haryana, India.
e-mail: shilparanikuk@gmail.com

One of the major chronic problems in software development is the fact that application requirements are almost never stable and fixed. Creeping user requirements have been troublesome since the software industry began. Several empirical studies have reported that volatile requirements are a challenging factor in most information systems development projects [7], [8], [9], [10]. There is no quick, perfectly effective cure. Various factors have been considered to be behind the creeping user requirements [3], [7], [11], [12], [13], [14], from which the following have been modeled in the presented study:

- Excessive Schedule Pressure
- User-Practitioner Relationship Level (which accounts for the User's involvement level and Practitioner's level of knowledge).

This study demonstrates the use of stochastic simulation as a flexible vehicle for effectively assessing and managing risk by measuring the effect of requirements creep on various software risk management factors using stochastic simulation.

IV. Potential Effects of Requirements Creep

The requirements creep level may be affected by the high schedule Pressure and User Practitioner Relationship level which in turn may affect the Defect generation rate, rework and job size [15]. The present algorithm simulates the effect of requirements creep by sampling the distribution of variables and continuously recalculating them after each run.

V. Algorithm

Symbol Used	Interpretation
CL	Creep Level
CCL	Cumulative Creep Level
INCREASE	Increase in Creep Level
CINCREASE	Cumulative Increase in Creep Level
SHPL	Schedule Pressure Level
CSHPL	Cumulative Schedule Pressure Level
IJS	Increased Job Size
CIJS	Cumulative Increased Job Size
UPRL	User-Practitioner Relationship Level
IncIJS	Increase in Job Size per unit rise in Creep Level
RWC	Rework Cost
CRWC	Cumulative Rework Cost
IncRWC	Increase in Rework Cost per unit rise in Creep Level
SRUNS	Number of Simulation Runs

STEP 1: Read Input data.
[Read SRUNS and UPRL]

STEP 2: Do the initialization:
[Set CL=0, CCL=0, INCREASE=0, CINCREASE=0, SHPL=0, CSHPL=0, UPRL=0, CUPRL=0, RWC=0, CRWC=0, IJS=0, CIJS=0, IncIJS=0, IncRWC=0]

STEP 3: Generate Schedule Pressure Level (from a random distribution)

STEP 4: CL=CL+SHPL-UPRL
RUN=RUN+1

STEP 5: If ((SHPL-UPRL)<CL) THEN {
INCREASE= CL- SHPL-UPRL
CINCREASE=CINCREASE+INCREASE
(Generate random values of IncRWC and IncIJS)
CIJS=CIJS+ INCREASE*IncIJS
CRWC= CRWC+INCREASE*IncRWC}
Compute defect generation percentage w.r.t. requirements creep level each time.

STEP 6: Compute Average Creep Level, Average Schedule Pressure Level, Average Rework Cost and Average Increase in Job Size.

STEP 7: Compute percentage of Defect Generation with respect to requirements creep level=((CCL-CSHPL)/CCL)*100

STEP 8: Print the computed statistics.

STEP 9: If RUN < SRUNS then go to STEP 3.
(Run for a large value of SRUNS)

STEP 10: END.

VI. Results & Interpretation

Table 1

User-Practitioner Relationship Level	Average Requirements Creep Level	
	Avg. Schedule Pressure=5	Avg. Schedule Pressure=8
1	220.607	343.166
2	173.958	296.061
3	128.358	250.334
4	83.534	204.735
5	38.884	159.708
6	6.5289	115.058
7	1.487	70.408
.	.	.
.	.	.

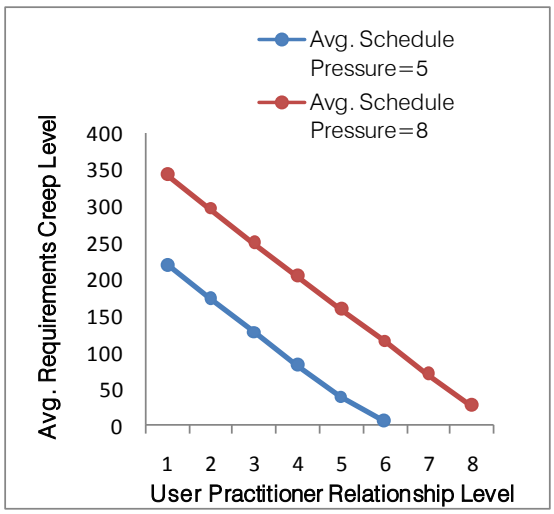

Figure 1 : Variation of Average Requirements Creep with User-Practitioner Relationship level

From the table1, it can be analyzed that a boost in User-Practitioner Relationship Level (i.e. User Involvement Level and Practitioner's Level of Knowledge) lowers the average level of Requirements Creep at a given schedule pressure.

Defect generation percentage calculated with repect to the level of rerquirement creep can be analysed with the help of figure2:

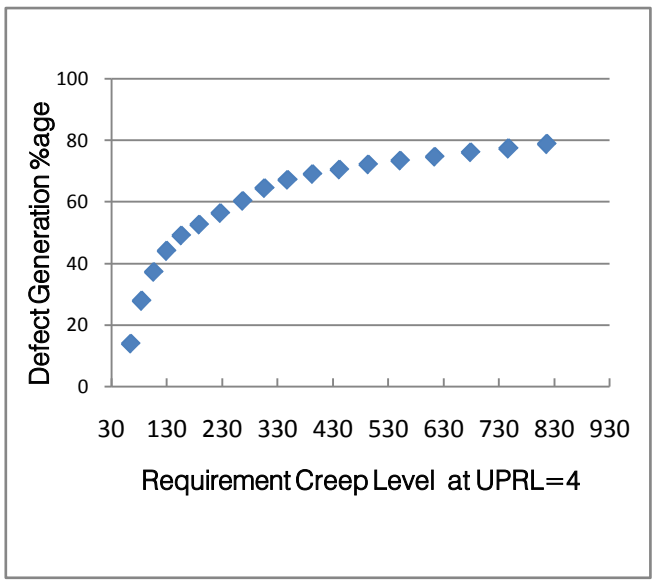

Figure 2 : Defect generation percentage with respect to requirements creep level at a certain User-Practitioner Relationship level

Increasing level of requirements creep may result into an increase in defect generation percentage.

User -Practitioner Relationship Level	AVERAGE REWORK COST at Avg. SHPL=5	Avg. SHPL=6	Avg. SHPL=8	Avg. SHPL=9
1	87.95	107.50	136.82	166.14
2	68.02	87.51	116.78	146.08
3	48.31	67.80	97.05	126.29
4	28.76	48.16	77.33	106.58
5	9.27	28.66	57.75	86.87
6	6.40	9.16	38.25	67.34
7	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Another simulation was done for different combinations of Schedule Pressure Level and User-Practitioner Relationship level which indicates that Average Rework Cost may increase with increase in Average Schedule Pressure Level but this happens at an increased User-Practitioner Relationship level (Table 2).

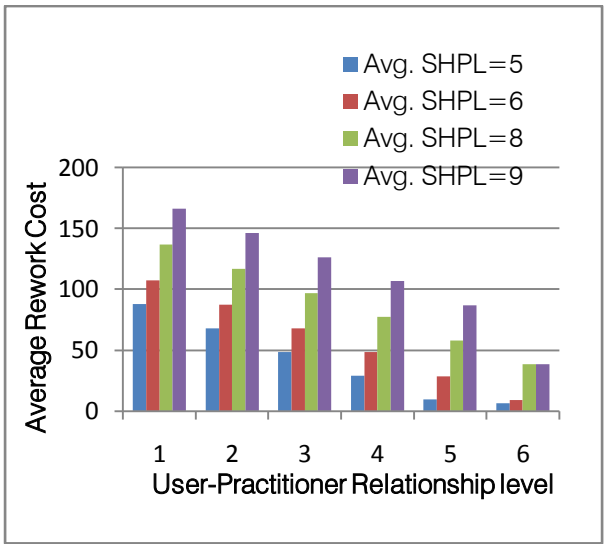

Figure 3 : Variation of Rework Cost with User-Practitioner Relationship Level at various schedule pressure levels

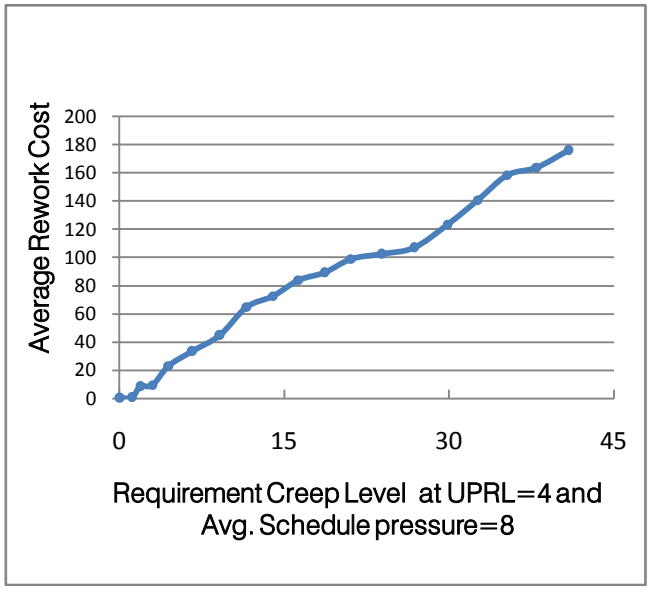

Figure 4 : Variation of Rework Cost with Requirements creep level at certain Schedule pressure level

An increase in requirements creep level may result into an increase in average rework cost. The increase becomes sharper at higher levels of requirements creep.

VII. Conclusion

The stochastic simulator presented here in this paper models the potential effects of requirements creep as a risk factor on various software risk management factors. This will enable software project managers to take decisions in planning and scheduling the various activities involved in software development and perform sensitivity analysis in order to achieve the desired risk mitigation goals.

References Références Referencias

1. Marc I. Kellner, Raymond J. Madachy, and David M. Raffo, "Software Process Modeling and Simulation: Why, What, How", Journal of Systems and Software, Vol. 46, No. 2/3, 1999.
2. Raymond J. Madachy, "A Software project dynamics model for process cost, Schedule and risk assessment", Ph.d. Dissertation, University of Southern California, 1994.
3. D. Houston, "A Software project simulation model for risk management", Ph.D. Dissertation, Arizona State University, Tempe, AZ, 2000.
4. B.W. Boehm, "Software risk management principles and practices", IEEE Software 8 (1), 32-41, 1991.
5. C. Jones, "Minimizing the risks of software development", Cutter IT Journal 11 (6), 13-21, 1998.
6. D. Gause and G. Weinberg, "Exploring Requirements", New York, NY, Dorset House, 1989.
7. D. Leffingwell and D. Widrig, "Managing Software Requirements - A Unified Approach", Addison Wesley Longman, Inc., 2000.
8. B. Curtis, H. Krasner and N. Iscoe, "A field study of the Software Design Process for Large systems", Communications of the ACM, 31(11), pp. 1268-1286,1988.
9. M. Lubars, C. Potts and C. Richter, "A review of the state of the practice in requirements modeling", Proc. of IEEE Symposium on Requirements Engineering, San Diego, Califonia,1993.
10. The Standish Group, The Standish Group Report - CHAOS, http://www.scs.carleton.ca/~beau/PM/Standish-Report.html, 2001-03-07, 1995.
11. K. R. Linberg, "Software developer perceptions about software project failure: a case study", The Journal of Systems and Software", 49(2-3):177–192, 1999.
12. J. Ropponen and K. Lyytinen, "Components of software development risk: How to address them? a project manager survey", IEEE Transactions on Software Engineering, 26(2):98–112, 2000.
13. R. Schmidt, K. Lyytinen, M. Keil, and P. Cule, "Identifying software project risks: An international delphi study", Journal of Management Information Systems, 17(4):5–36, 2001.
14. L.Wallace, M. Keil, and A. Rai, "Understanding software project risk: a cluster analysis", Information and Management, 42(1):115–125, 2004.
15. D. Houston, G. Mackulak and J. Collofello, "Stochastic simulation of risk factor potential effects for software development risk management", The Journal of Systems and Software 59 (2001) 247-257, 2001.