RCAUSE – A ROOT CAUSE ANALYSIS MODEL TO IDENTIFY THE ROOT CAUSES OF SOFTWARE REENGINEERING PROBLEMS

Er. Anand Rajavat [1] and Dr. (Mrs.) Vrinda Tokekar [2]

[1]Department of Computer Science & Engineering, Shri Vaishnav Institute of Technology and Science Indore, M. P., India
anandrajavat@yahoo.co.in

[21]Department of Information Technology, Institute of Engineering and Technology (DAVV), Indore, M. P., India

Vrindatokaker@yahoo.co.in

ABSTRACT

Organizations that wish to modernize their legacy systems, must adopt a financial viable evolution strategy to gratify the needs of modern business environment. There are various options available to modernize legacy system in to more contemporary system. Over the last few years' legacy system reengineering has emerged as a popular system modernization technique. The reengineering generally focuses on the increased productivity and quality of the system. However many of these efforts are often less than successful because they only concentrate on symptoms of software reengineering risk without targeting root causes of those risk. A subjective assessment (diagnosis) of software reengineering risk from different domain of legacy system is required to identify the root causes of those risks. The goal of this paper is to highlight root causes of software reengineering risk. We proposed a root cause analysis model RCause that classify root causes of software reengineering risk in to three distinctive but connected areas of interest i.e. system domain, managerial domain and technical domain. .

KEYWORDS

RCause, Reengineering, legacy system

1. INTRODUCTION

Over the last few years' economical and political conditions of business environment resulted in numerous rising challenges for software organizations. Expectation levels of modern business environment grow in rapid succession. One important factor that affects how quickly an organization can change its working practices and policies is the ease with which it can modify its legacy systems, to support the new way of working. A legacy system is a system which was developed sometime in the past and which is critical to the business in which the system operates. Typically, legacy systems were developed before the widespread use of modern software engineering methods and have been maintained to accommodate changing requirements. [1] Unfortunately, many organizations are encumbered with legacy systems that are extremely difficult to change.

A legacy system may evolve in a number of ways, continued maintenance, reengineering [2] and replacement are the general evolution strategies of which one or a combination may be an appropriate way of evolving a legacy system. [3].The research indicates that reengineering of

DOI : 10.5121/ijcsit.2011.3319

programs and data reduces the development cost and reengineering projects must also be completed within a much shorter time frame. However software reengineering risk from different domain and their impact on software quality causes reengineering efforts to fail

The purpose of this paper is to highlight some of the most important root causes for software reengineering risk .Evolution of legacy systems through reengineering requires attention to root cause of software reengineering risk. Proposed root cause analysis model classifies root causes of software reengineering risk in to three distinctive but connected areas of interest i. e. system domain, managerial domain and technical domain. Successful reengineering requires attacking all root causes so that we are not only in position to modernize the system, but also we are in a better position to provide quality to legacy system in a repeatable and productive manner.

2. RELATED WORK

Reengineering is a disciplined approach for the evolution of legacy systems. Reengineering process involves applying reengineering principles to an existing system to meet new requirements. However, development of successful reengineering effort needs to consider reengineering problems from a number of different perspectives.

Paul Bride in [4] provides detail activities of reengineering process of software systems but he could not touch issues related to root causes of software reengineering risk. Several authors have focuses on technical aspects of reengineering [5]. However, system, managerial, and quality aspects of a legacy system play an important role in the successful implementation of reengineering efforts.

Harry M. Sneed in [6] estimates risk of reengineering project in terms of project completion rates and cost overruns [7]. However reengineering effort also required to considering other factors like performance improvement, resource utilization, quality goals, user satisfaction etc. Eric K. Clemons in [8] suggests that two principal reasons for failure of reengineering efforts are functionality risk and political risk. Though there is other serious risk such that technical risk, process risk, development environment risk, architecture risk, and risk related to stakeholders are also need to be considered.

Proposed work firstly summaries and categories risk of legacy system reengineering in to three different but connected areas of interest i.e. system domain, managerial domain and technical domain. The purpose of this work is to highlight some of the most important root causes of software reengineering risk.

3. RCAUSE (ROOT CAUSE ANALYSIS) MODEL

Root cause analysis (RCA) is a problem solving methods aimed at identifying the root causes of problems. The practice of RCA is predicated on the belief that problems are best solved by attempting to correct or eliminate root causes, as opposed to merely addressing the immediately obvious symptoms. By directing corrective measures at root causes, it is hoped that the likelihood of problem recurrence will be minimized.

Proposed root cause analysis model RCause identify root causes of software reengineering risk by considering system, managerial and technical domain of legacy system in accordance with requirements of target system. RCause model identifies and categories software reengineering risk in to three different areas of interest i.e. system domain, managerial domain and technical domain.

4. ARCHITECTURE OF RCAUSE MODEL

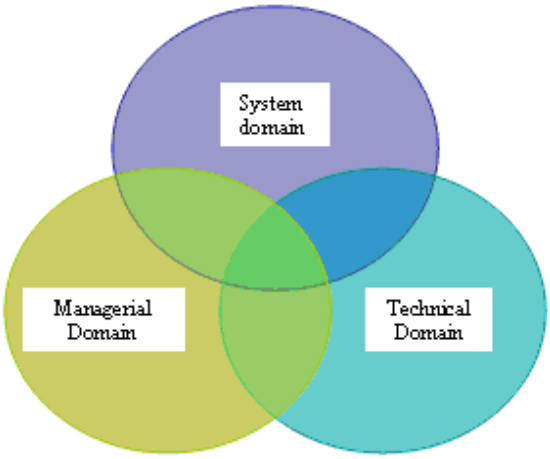

Figure.1 Architecture of RCause Model

Fig .1 represents the basic architecture of RCause model comprises with system domain, managerial domain and technical domain.

System domain represents the first subsection of Rcause model. System domain includes planning and structuring the system Infrastructure efforts, organizing the stakeholder's tasks and ensures that the products and services fulfill the organization's goals and objectives. The first subsection of Rcause model is shown in figure 2 which demonstrate major risk involved in system domain of legacy application.

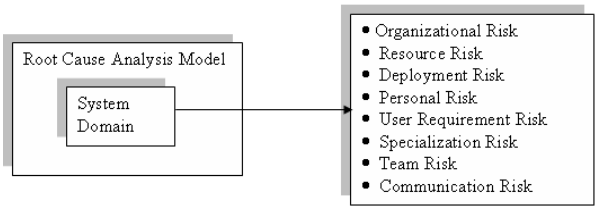

Figure.2 RCause subsection-1

Managerial domain represents the second subsection of Rcause model. Managerial domain identifies impact of market factors and effect of competitive products, on quality & cost of target system. Managerial domain covers issues of identification and measurement of organizational economic value to support system evolution activities. The second subsection of Rcause model is shown in figure 3 which demonstrate major risk involved in managerial domain of legacy application.

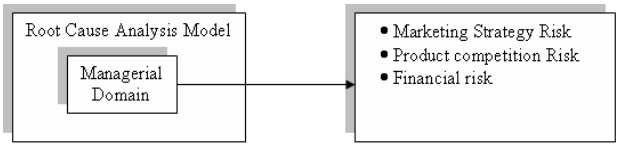

Figure.3 RCause subsection-2

Technical domain represents the third subsection of Rcause model. Technical domain covers issues related to software functionality and software quality. Technical domain identifies legacy systems functional capabilities and quality features and assesses the impact of proposed changes. The third subsection of Rcause model is shown in figure 4 which demonstrate major risk involved in technical domain of legacy application [9][10].

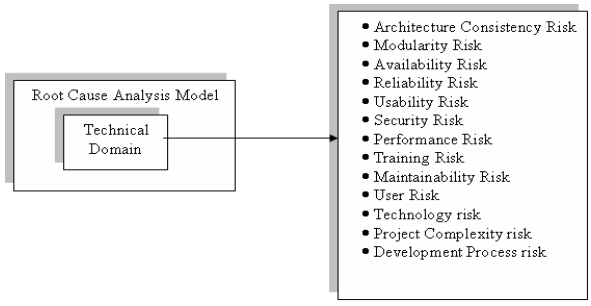

Figure.4 RCause subsection-3

RCause Model

RCause model is used to address different software reengineering risk, in order to get the "root cause" of those risks. It is used so we can correct or eliminate the cause, and prevent the reengineering risk from recurring. The RCause model firstly summaries the general risk associated with legacy system reengineering and then address the root causes of those risks. Rcause model identifies and analyze software reengineering risk by concentrating on three different areas of legacy system i.e. system domain, managerial domain and technical domain.

Table 1 articulate the root cause of risk associated with system domain of legacy application

Table 1 Root causes of system domain risks

S.No.	Risk Item	Root cause
1	Organizational risk	1. Unfinished organizational policies and standards. 2. Undefined statement of needs. 3. Complex Review and Approval process of organization. 4. Complex and unstable organizational environment. 5. Lack of management support and attention.
2	Resource Risk	1. Inadequate estimation of required resources. 2. Poor quality of available Resources. 3. Inadequate resources (personal and technical) to support the reengineering task. 4. Shortfalls in reusable components. 5. Unavailability of required information (analysis and design documents) about existing system.

4	Personal Risk	1. Personals inconsistency with the organizational culture. 2. Inadequate managers and leaders. 3. Lack of motivation. 4. Unbearable Schedule. 5. Unrelated job assignment and unsatisfied reward. 6. Lack of technical knowledge in advanced technology to support system evolution task.
5	User Requirement Risk	1. Requirement changes frequently. 2. Inadequate, ambiguous and unusable requirements. 3. Users negative attitude towards system evolution objectives. 4. Misinterpretation of system evolution objectives. 5. Lack of user involvement. 6. Conflicts between Users and developers.
6	Specialization Risk	1. Lack of user experience in specialized tools and Techniques to support system evolution activities. 2. Inexperience and amateur workforce for system evolution task. 3. Lack of specialized skills in team members required by system evolution task. 4. Inexperience team members.
7	Team Risk	1. Complex team structure. 2. Lack of cooperation among team members. 3. Inadequate managers and leaders. 4. Frequently turnover of team members. 5. Unwillingness to work together to resolve conflicts. 6. Non-existence of coordination plan.
8	Communication Strategy risk	1. Complex process for information exchange. 2. Communication gap between stakeholders involve in system evolution process. 3. Mismatch between characteristics of stakeholders and requirements of the reengineering process. 4. Non-existence of coordination plan. 5. Change in organizational management during the project.

Table 2 articulates the root cause of risk associated with managerial domain of legacy application.

Table 2 Root causes of managerial domain risks

S. No.	Risk Item	Root cause
1	Development Process Risk	1. Uncontrolled development process. 2. Lack of life cycle model support. 3. Unavailability of quality assurance plan. 4. Complex review and approval process. 5. Undefined success factors and measures. 6. Ineffective project management plan. 7. Poorly designed business process. 8. Unrealistic schedule and budget. 9. Inadequate measurement policy for process component.
2	Marketing strategy Risk	1. Frequently changes in marketing strategy towards new product. 2. Poor system evolution planning to satisfy market demand. 3. High user expectation from target system. 4. Product demand decreases with time.
3	Product competition Risk	1. Availability of high quality competitive product. 2. Availability of low cost competitive product. 3. Availability of many types of competitive product.
4	Financial Risk	1. Unavailability of required funds to support system evolution task. 2. Delayed in fund release by upper levels. 3. Complex process for financial approval. 4. Low degree of ROI (returns on investment).

Table 3 articulate the root cause of risk associated with technical domain of legacy application

Table 3 Root causes of technical domain risks

S. No.	Risk Item	Root causes
1	Architecture consistency risk	1. Inconsistency between architecture of legacy and target system. 2. Heterogeneous legacy system framework to support sub-system control and communication. 3. Complex legacy system architecture to understand stakeholders needs. 4. Low degree of reusability within legacy system architecture. 5. Low performance and maintainability due to complex legacy system architecture.
2	Modular Risk	1. Inconsistency between system organization and modular decomposition.

		2. Unmodularized complex legacy system structure. 3. High degree of coupling between the modules of legacy system. 4. Complex control structure among modules.
3	Availability Risk	1. Unavailability of required analysis and design documents of legacy system. 2. Unavailability of required resources on time. 3. Unavailability of user requirements to support target system. 4. Low value of legacy system availability.
4	Reliability Risk	1. High degree of coupling between the modules of legacy system. 2. Percentage of reused code is very low. 3. Weak fault tolerance. 4. High failure rate which increases mean time between failures. 5. Large amount of time required for repairing which increases mean time to repair.
5	Usability Risk	1. The usability rate of functions provided by legacy system is very low. 2. Complex User interface of legacy system.
6	Security Risk	1. Required Information lost or theft. 2. Inadequate policy for information security. 3. Ineffectiveness of security regulations.
7	Performance Risk	1. Inadequate Infrastructure (Technical and information system) to support the target system. 2. Poor organizational structure. 3. Lack of specialized skills in team members required by the system evolution task. 4. Poor project planning towards reengineering. 5. Unrealistic Schedule and Budget. 6. Inadequate management of process component i.e. human, physical and financial. 7. Requirements changes frequently. 8. Complex system evolution process. 9. Inexperience and immature team members.
8	Training Risk	1. Aging and unionized work force to support operation of target system. 2. Long-term maintenance contract with an outside vendor. 3. Unavailability of adequate training Program for the existing work force.

		4. Key members of the project team required training on new technology used in target system. 5. High cost and schedule impact of applying the new technology due to extreme training program.
9	Maintainability Risk	1. Unavailability of maintenance documents. 2. System progresses not monitored close enough. 3. Revolutionize development team during system evolution.
10	User Risk	1. Nonexistence of coordination plan between user and developers. 2. Lack of user involvement. 3. Unavailability of user on required time. 4. User resistance to change. 5. Lack of cooperation from user. 6. Users with negative attitude towards the target system.
11	Technology Risk	1. Incompatability of legacy system with the concept of new technology. 2. Technology newness. 3. Technical Infeasibility. 4. Continuous changing system requirements. 5. Immature technology.
12	Project complexity Risk	1. Mismatch of the target system with the organizational strategy. 2. Large number of components in legacy system. 3. Large number of links to existing system and external entities. 4. Nonexistence of coordination plan among team members. 5. Undefined success factors and measures.

5. CONCLUSIONS

Over the past few years, organizations face with a very high competition and as a result they have to continuously improve their legacy system to satisfy current user and business needs. Legacy system reengineering is a well-known system modernizing technique which helps in effective cost control, quality improvements and time and risk reduction. The goal of reengineering is to increase productivity and quality of legacy system through fundamental rethinking and radical redesigning of system. Reengineering is the opportunity to develop optimal version of legacy system to satisfy all the stakeholders. However many reengineering projects are often less than successful because they concentrate on a narrow set of risk issues for that reason there is a great need to identify root causes of reengineering risk from different

domains of legacy system. It would be strategically advantageous for organizations if they could predict the root causes to tackle possible reengineering risk so that reengineering provides every attempt to seize and maintain true competitive advantages. Evolution of legacy systems through reengineering requires attention to root cause of software reengineering risk. Proposed RCause model identify software reengineering risk for three different domain of legacy system and analyze root causes of those risk in an efficient manner.

REFERENCES

[1] J. Ransom, I. Sommerville, I. Warren," A Method for Assessing Legacy Systems for Evolution", CSMR '98 Proceedings of the 2nd Euromicro Conference on Software Maintenance and Reengineering (CSMR'98), ISBN:0-8186-8421-6,PP :128.

[2] Peter H. Feiler, "Reengineering: An Engineering Problem," Technical Report Software Engineering Institute Carnegie Mellon university Pittsburgh Pennsylvania 15213, CMU/SEI-93-SR-5, 1993.

[3] Ransom J., Somerville I., Warren I. ," A Method for Assessing legacy systems for evolution," in Proceedings of the Second Euromicro Conference on Software Maintenance and Reengineering, 1998,ISBN: 0-8186- 8421-6, Digital Object Identifier 10.1109/CSMR.1998.

[4] Paul Briden, "Software Re-engineering Process," Tessella Support Services PLC Technical report, Issue V2.R1.M1, 2000.

[5] Peter H. Feiler, "Reengineering: An Engineering Problem," Technical Report Software Engineering Institute Carnegie Mellon university Pittsburgh Pennsylvania 15213, CMU/SEI-93-SR-5, 1993.

[6] Harry M. Sneed, "Risks Involved in Reengineering Projects," in WCRE: Proceedings of the 6th IEEE Conference on Reverse Engineering, PP.204, 1999.

[7] Sneed H., "Economics of Software Reengineering" in Journal of Software Maintenance, Vol. 3, No. 3,PP. 163, 1991.

[8] Eric K. Clemons Michael C. Row Matt E. Thatcher, "An Integrative Framework for Identifying and Managing Risks Associated With Large Scale Reengineering Efforts," Proceedings of the 28th Annual Hawaii International Conference on System Sciences, PP.960-969, 1995.

[9] Anand rajavat,Vrinda Tokaker," MngRisk –A Decisional Framework to Measure Managerial Dimensions of Legacy Application for Rejuvenation through Reengineering",International jurnal of computer application 2011 by IJCA Journal, Number 2 - Article 4 ,DOI 10.5120/1985-2674,2011.

[10] Anand Rajavat, Vrinda Tokekar, "ReeRisk –A Decisional Risk Engineering Framework for Legacy System Rejuvenation through Reengineering", Published in Proceedings of Second International Conference on Recent Trends in Information, Telecommunication and Computing – ITC 2011 by Springer LNCS-CCIS, March 10-11, 2011 in Bengaluru, India, CNC 2011, CCIS 142, pp. 152 – 158, 2011, © Springer-Verlag Berlin Heidelberg 2011

Authors

Er. Anand Rajavat
Assistant professor
Department of Computer Science & Engineering
Shri Vaishnav Institute of Technology and Science Indore, M. P., India
B.E. (Computer Science and Engineering) in 2000 from vikram university Ujjain
M.E. (Software Engineering) in 2007 from DAVV, Indore
Ph.D (Pursuing) (Computer Engineering) from DAVV Indore
Areas of Interest: Software Engineering, Object Oriented Analysis and Design, Software Architecture

Dr. (Mrs.) Vrinda Tokekar
Professor & Head,Department of Information Technology,
Institute of Engineering & Technology, Devi Ahilya University, Indore (M.P.) India
Ph. D. (Computer Engg.) in 2007 from DAVV, Indore
M.E. (Computer Engg.) in 1992 from DAVV, Indore
B.E. (Hons.) EEE, BITS Pilani in 1984,
Areas of Interest:Computer Networks, Distributed Computing, Security in Wireless Networks, Multimedia Communication, Software Engineering