

# The Classification and Analysis of Risks in Reengineering System

Josphin Jasaline Anitha.X  
Assistant Professor  
MCA Department  
N.M.S.S.Vellaichamy Nadar College  
Madurai, Tamil Nadu, India.

M.Karthika  
Assistant Professor  
MCA Department  
N.M.S.S.Vellaichamy Nadar College  
Madurai, Tamil Nadu, India.

K. Alagarsamy  
Associate professor  
Computer Centre  
Madurai Kamaraj University  
Madurai , Tamil nadu, India.

## ABSTRACT

Software engineering has become an important field of computer science and an active research field. Due to new trend and technology most of the software is in need of change. Any software that has crossed a decade are incapable of satisfying customer need with current technology is named legacy system. To overcome this hazard and to be cost benefited, in facing the new trends the software has to be reengineered in a benefiting way. The legacy system, otherwise called existing system has to be reengineered. In Most of the reengineering system, the legacy transformation is the process of modernizing an operational system to retain and extend the value of investment in that system. It involves both infrastructure and application modernization. The primary benefit of legacy transformation is to enhance the business process and improve functionality of business objective. Legacy transformation projects are frequently challenged, because a set of risks will threaten the project success of legacy transformation. This paper presents a set of risks and their classification. From the analysis of risks, some mitigation that helps to make the reengineering projects more beneficial is suggested.

## Keywords

Reengineering, Legacy system, Reverse Engineering, Forward Engineering, mitigation.

## 1. INTRODUCTION

Software reengineering is reorganizing and modifying existing software systems to make them more maintainable. That is restructuring or rewriting a part or all of a legacy system without changing its functionality [1]. Reengineering involves adding effort to make them easier to maintain, the system may be restructured and redocumented. The purpose of reengineering is to modernize the existing system over a newer one. Aging and unreliable system components, when system becomes outages frequently. Changes to business processes become too complex, difficult, and/or costly to implement. Solutions are less expensive than legacy maintenance. The advantages of reengineering are reduced risk at reduced cost.

The reengineering process may also face various types of risks as software engineering. The risk identification is an art. The risk identification is more important for effective risk assessment, risk analysis, and management. In the proposed work, the potential risks are analyzed and categorized. A monitoring method is explained for the categorized risks. It will help a reengineering system towards an ease of maintenance and cost benefit with reduced risk at reduced cost.

## 2. REENGINEERING- AN OVERVIEW

The examination and alteration of a system to reconstitute it in a new system is known as reengineering [9]. It is also a process of modification of a software system that takes place after it has been reverse engineered, generally to add new functionality, or to correct errors. The reengineering process

typically encompasses a combination of other processes such as reverse engineering, re-documentation, restructuring, translation, and forward engineering.

The main objective of reengineering is shown in Figure.1, which helps to understand the existing software system, and to enhance, redesign, re-implement the original or old functionality with the current trend of technology, and with added subsystem to get more profit with latest technology.

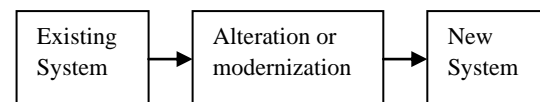


Fig.1 Reengineering process

The current system in use is called as *legacy system*. When the legacy system becomes old and weak in its design, structure, and coding, new modules can be added to strengthen the legacy system. This process makes the system thornier and it won't be cost benefited.

Whenever a new system is evolved, the system development life cycle consists of the following stages:

- Feasibility study
- Requirement and analysis
- Design
- Coding
- Testing
- Implementation and
- Maintenance

But, all these stages are not needed in reengineering. When we engineer the legacy system definitely it might take time, money, and people to buildup. The invested legacy system cannot be simply "*wasted or thrashed*". The legacy system has to be redefined for reengineering. Then the system will be more economical with the current trend and new technologies and will be at par.

The core objective of the redefined reengineering system is, fitting the modules with improved technology, functionality and explicit interfaces. The documentation will also match with the current trend and technology. Requirement definition of reengineering projects first requires the assessment of the

current legacy application and portfolio and infrastructure. This assessment should identify at minimum:

- Core applications that support the business and application that overlap or are redundant
- Obsolete or unused application with current business rules and processes
- Technical value and operational efficiency of application
- Infrastructure components and constraints
- Potential improvements in efficiency and reduction of redundancy in both system and current business processes.
- Latest trend and software technology that match the current business process.

A thorough understanding of all these items is essential to specify requirements that definitely meet the business need within technical constraints.

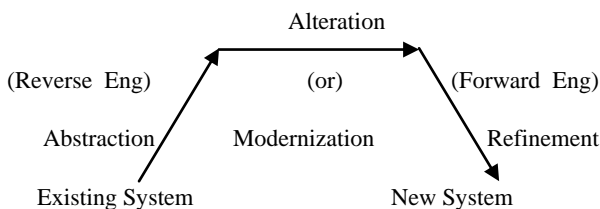
### 2.1 Reengineering life cycle:

When hardware or software support becomes obsolete and when tools to support re-structuring are available, a system can be taken up for reengineering. The advantages of reengineering are reduced risk at reduced cost. The reengineering life cycle has two parts.

- Reverse Engineering*
- Forward engineering*

**i.Reverse Engineering:** Reverse Engineering is the process of analyzing a software system in order to

- identify its components and their interrelationships and
- Create representations of the system in another form or at a higher level of abstraction. [1,4]



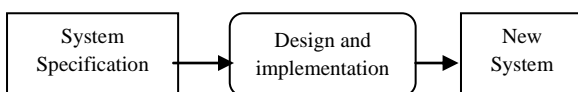
**Fig.2 Reverse Engineering**

Reverse engineering is a key to the rest of the process of reengineering, because it enables us to take an existing software system that is being reengineered (e.g., in terms of its source code) and to recover an abstract representation that can be used for subsequent reengineering or even reimplementation.

**ii. Forward Engineering:** The Forward Engineering is the

- Process that applies software engineering principles, concepts, and methods to re-create an existing application.

Simply it is the process of legacy system to new system. Figure.3 shows the functions of forward engineering [8].



**Fig. 3 Forward Engineering**

The forward engineering evolves in redocumentation, restructuring, and remodularization.

### 2.2 Benefits of Reengineering

For many reasons, especially economical, any system or project needs redevelopment. In the case of reengineering, reuse of legacy components will help to make a new system. Some of the benefits are as follows:

- *Improves the quality and consistency of business plans.*
- *Improves the financial reporting.*
- *Makes fair of bonus plans possible.*
- *Filters the macroeconomic factors*
- *Extends the planning process' relevance.*
- *Lowers Costs*
- *Lowers Risks*
- *Better use of existing staff*
- *Reveals business rules*
- *Incremental development*
- *Reduces operational costs*
- *Improves system performance, accessibility, system documentation*
- *Preserves legacy investments*
- *Minimize impacts to business operations*
- *Add or enhance software functionality*

As a system evolves in reengineering, the purpose and success secret of the system are rediscovered. Reengineering can be done by different stages, such as budget and available resources. The functional organization has a working system in all time, and the users are able to gradually become accustomed to the reengineered system as it carries in the incremental way. Always reengineering is an incremental improvement of systems, rather than a drastic change in the system. This will improve the system as per user's need. Moreover the existing staff members can be train in the reengineered system without losing them. Which will help to evolve the skills of the existing staff member's as well as system. With the reengineering system the benefits can be classified as

- *Portability*
- *Ease of maintenance*
- *Migration*
- *Reliability*

Legacy transformation projects have many of the same risks as other types of system development projects. The impacts of those risks are more severe than the traditional projects, because legacy transformation projects involve changes in functional operations that may be integral with the current trend and technology to business operations.

### 3. RISK:

Risk is a likely shortfall between the desired level of performance and the actual performance at a future time. [16].

#### 3.1 Risk Assessment:

A systematic approach to observe the working methods, expectation of trouble and find out a suitable monitoring method to prevent loss in the business and hostility in the work place is known as assessment. The assessment should include the controls required to eliminate, reduce or minimize the risks.

#### 3.2 Controlling Risks:

Any system must be conscious about the risks which may occur at any time and at any place. To control the risks the following principles of measurement should apply in any system.

- The risks can be avoided or eliminated
- The affected part of the system has to be analyzed.
- The preventive method to control the risks
- Technical method can implement to control risk

Information, instruction and training, and providing personal protective equipment should always be considered as a last resort after the above controls have been considered. Employers should, wherever possible, give priority to those controls that best protect everyone exposed to the risk.

#### 3.3 Risk Mitigation

When the risks are identified in a system, the elimination process will take place. Over the period the practice of finding risks and alleviation is easy.

#### 3.4 Risk Analysis

Risk analysis has become a standard requirement for legacy process transformation projects in many organizations. The basic thing is to identify, analyze, classify, and categorize the risks involved. When the risks are systematically listed out and categorized, the mitigation process become easier and economically sound in profit. Any reengineering system faces some risks when the process transformation takes places. Such risks are as follows.

- Mission and Goal
- Program management
- Decision drivers
- Organization management
- Customer/User
- Project parameters
- Product content
- Deployment
- Development Environment
- Project Management
- Project team
- Technology
- Maintenance

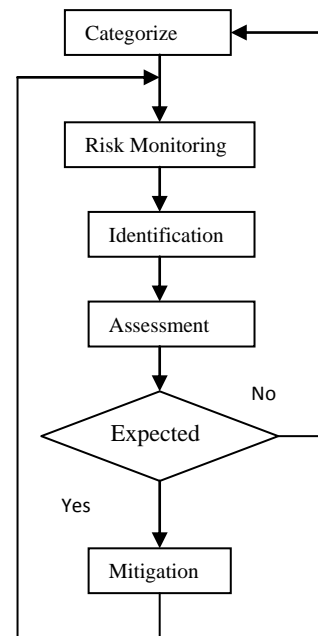
### 4. THE RISKS IN RE-ENGINEERING SOFTWARE

Although re-engineering is often used as a means to reduce the risk, reduce costs and maintain the operation of deriving the software, but re-engineering cannot avoid risk. The risk of early identification programs to support and project management in the preparation of estimates, risk assessment of software re-engineering and provide a practical framework for the expectations. Identifying risks is essential for effective

risk assessment, risk analysis and risk management. This paper shows the categorization and identification of risks and an evolution of mitigation methods for reengineering projects. While monitoring the project in a continuous manner, the identification of risks can be done at the initial stages. Once the risk is identified the mitigation process is easy with the expert team and we can avoid the problem in functional operations. The maintenance part also becomes very easy.

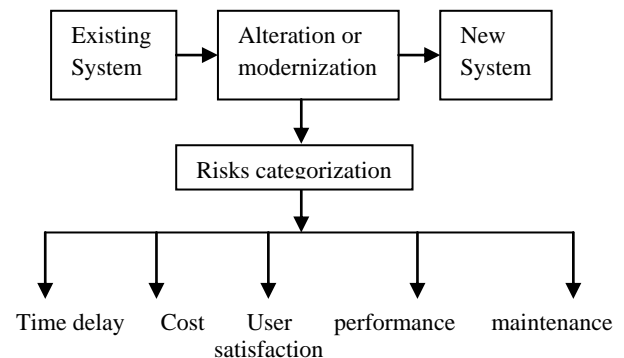
This will make the project a profitable one. The main advantage of reengineering is reduced risk at reduced cost. This can be easily proved. The risk mitigation is shown in Fig.4 consists of the following stages. *Classification, Identification, Assessment, Mitigation, Monitoring.* Whatever the risks may arise we categorize them into the limited divisions. With continuous monitoring of risk we can identify the risk when it occurs. The assessment can also be done and check whether the risk is known or unknown risk. If the risk is known and expected, mitigate it with expert team otherwise, the unknown risk should be categorize and go back to risk monitoring system.

The process can be done in the following manner,



**Fig 4. Risk monitoring methodology**

The purpose of the study is to suggest a method to monitor the reengineering software, to identify the risks when it occurs and categorize them and mitigate with appropriate solution. This will lead the reengineering software to reduce risk and make it most cost benefited. The risks can be categorized in to the following five types as depicted in Figure.4.



**Fig.5 Classifications of Risks**

**Time delay:** Time is the prime risk factor for any project. Time is needed to anticipate and prevent problems. Time is the great equalizer, since every day that is made available to the project is an additional day to deal with the consequences of risk. By managing risk, time can be used to an advantage, rather than being wasted. The risks related to time factor are

- *Lack of knowledge about legacy systems.*
- *Lack of understanding the concepts of legacy system.*
- *The vital role played by organizational hierarchy in making decision.*
- *Political reasons.*
- *Delivering the project on time.*
- *Budget cost overflow.*
- *Migration of team people*

**Cost benefit:**

Any project is reengineered for business needs to face the market with new trend and technologies and to make it more cost benefited. Risks involved in cost benefit are as follows:

- *Less benefit/profit*
- *High maintenance cost*
- *Expensive backup*
- *High cost to finance report*
- *Poor quality and inconsistency of business plans*
- *Loss of legacy investments*

**Performance:** Performance Risk is the degree of uncertainty in the development and deployment process that may keep the system from meeting its technical specifications or that may result in the system being unsuitable for its intended use.

The reengineering projects depend on the performance of new system. The new system performance should definitely be better than the legacy system. The Performance risks are:

- *Non portability*
- *Result mismatching*
- *Unsupported legacy system*
- *Reliability mismatch*
- *Incompatible Result*
- *Improper selection of reengineering approach*
- *Inappropriate Data restructuring*

**User/Customer Satisfaction:** In a competitive marketplace where businesses compete for customers, customer satisfaction is seen as a key differentiator and increasingly has become a key element of any business strategy.

**Table: 1 Risk Verses Development life cycle**

Therefore, it is essential for any business to effectively manage customer satisfaction.

To do this, firms need reliable and representative measures of satisfaction. The reengineering system's customer /user satisfaction risks are listed below.

- *Budget overflow*
- *Unexpected result*
- *Less User friendliness*
- *Unsupported to referential model*

**Maintenance:** Maintenance is really an evolutionary development and maintenance decisions are aided by understanding what happens to any software systems over time. The key software maintenance issues are both managerial and technical. Key management issues are: alignment with customer priorities, staffing, which organization does maintenance, estimating costs. Key technical issues are: limited understanding, impact analysis, testing, and maintainability measurement. The maintenance activities are categorized into four classes:

- *Adaptive – dealing with changes and adapting to the software environment*
- *Perfective – accommodating with new or changed user requirements which concern functional enhancements to the software*
- *Corrective – dealing with errors found and fixing it*
- *Preventive – concerns activities aiming on increasing software maintainability and prevent problems in the future.*

The maintenance risk factors are

- *Backup*
- *Recovery of legacy systems*
- *Improper Data restructuring*
- *Improper Redocumentation*
- *Alternative system solution*

The risks which may occur in reengineering are tabulated according to people, project, process, product in the software development life cycle.

Identified Risks	4 P's Development life cycle			
	People	Project	Process	product
User satisfaction	Communication	Technology	Degrade	Alternative system
Cost	High salary	Budget overflow	Poor quality	Loss
Performance	Hazard	Deployment	Maze	Backup
Time delay	Incompetent	Mission and Goal	On time submission	Delivery time
Maintenance	Migration	Enhancement	Incompatible	Reliability

Thus we can categorize the risks that we expect and meet in legacy and in a new system. If the methods discussed either to be followed, the monitoring of risk is very easy. When we identify the risks we can mitigate them with the experts without affecting the functions of the new system. In such a way, we can make the reengineering system more cost benefited.

## **5. CONCLUSION**

Whenever reengineering takes place a variety of risks might occur. This paper proposes various risks which are classified and a monitoring method is taken for discussion. It will help to identify the risks and to categorize the risks to mitigate them properly. The main outcome is, the organization must feel easy to make the project with reengineering at a low budget, with current trend and technology, the customer satisfaction. It will lead the project more cost benefited with good performance and with an ease of maintenance. Further, risks involved in process transformation can be identified, analyzed and mitigation methods can be evolved. An evolutionary method will be done in a comprehensive manner in my future work related with process transformation, risk analysis and methods of mitigation in reengineering can become effective and efficient.

## **6. REFERENCES:**

- [1] Arnold.R., “ Software reengineering” IEEE, Computer society press, 1993.
- [2] Capers jones, “Early sizing and early risk analysis of software projects”, 2011.
- [3] Chia-chu chang “Software stability in software reengineering”. Information Reuse and Integration, 2007.
- [4] Chikofsky.E.J; Cross.J.H.,” Reverse Engineering and Design Recovery : A Taxonomy”, IEEE Software, Vol.23, 1990,pp.13-17.
- [5] Gerardo canfora, Aniello cimitile, “Software maintenance”, Journal of Software Maintenance 13(1): 1-2 (2001)
- [6] Harry.M.Sneed, “Economics of Software reengineering”, Journal of Software Maintenance, Vol.3, 1991,p.163.
- [7] Harry M.Sneed. “Risks involved in reengineering projects.,proceedings of WCRE” , IEEE Computer Society Press, Atlanta, 1999, pp -204-211.
- [8] Ian Sommerville.,2000. Software engineering .,
- [9] Dr.Linda and Rosenberge , “Software reengineering”.
- [10] .Moghaddas, Y., & Rashidi, H. (2009). A novel approach for replacing legacy systems. Journal of Applied Sciences, 9(22), 4086–4090.
- [11] Dr.Roger.S.pressman, “Software engineering” , fifth edition.
- [12] M. Solvin, and S. Malik. “Reengineering to reduce system maintenance: A case study”, Software Engineering, pp.14-24,2011.
- [13] Dr. ying zou, “Software reengineering evolution”.
- [14] Jakub Miler, Janusz Górski. “Identifying Software Project Risks with the Process Model”, ICSSEA2004.
- [15] John bergey, Dennis smith, Scott titley, Nelson Weiderman, Steven wonds. “Why reengineering projects fail”.
- [16] Planning mentor [www.layrib.com](http://www.layrib.com)