RESEARCHERID
THOMSON REUTERS

[Salman et al., 5(5): May, 2018] ISSN 2349-4506
Impact Factor: 2.785

Global Journal of Engineering Science and Research Management

MONITORING SOFTWARE MAINTENANCE PROJECT RISKS USING SINGULAR VALUE DECOMPOSITION

Hawraa Ahmed Salman
*Business Administration Department, AL ESRA'A UNIVERSITY, Baghdad, Iraq
DOI: 10.5281/zenodo.1255779

KEYWORDS: Enterprise System software maintenance; risk management, Risk Identification

ABSTRACT
Enterprise Systems are the main concern that have busied the organizational project managers. The risks in Enterprise Systems are performed by IT system such as Enterprise Resource Planning (ERP), Customer Requirement Management (CRM), and Product Data Management (PDM). The maintenance of software projects have different features. These include increased complexity and higher project failure rates. This paper present Singular Value Decomposition (SVD) correlated with the traditional risk factor calculations to estimate the software maintenance projects. The study proposed a risk functional model to estimate the Enterprise Systems performance in maintenance phase and the risk effect. The results specify the insufficient expertise and staffing due to software management and organization as the main risk factors in maintenance phase.

INTRODUCTION
Enterprise System Risk Management (ESRM) is the process of identifying, assessing and controlling threats to an organization's capital. These risks could arise from a wide variety of sources such as strategic management errors, accidents and natural disasters [1]. In Development of Life Cycle, Enterprise System is the process of improving their efficiency and operational flexibility. Quality improvement is one of the development techniques in this field [2]. The Quality improvement models and methodologies include Planning, analysis, design, implementation, and maintenance phases. Software of Enterprise maintenance System is a critical issue, because if it is not fit, the system will soon not be useful. Maintenance is an unavoidable activity required to keep systems synchronized with the reality [3]. The present work focuses on Enterprise Systems software maintenance phase which creates many risks that must be managed diligently. Early identification and recognition of risks can changes the course of actions to mitigate and reduce the risks. Webster et al, in 2005 proposed a taxonomy of possible risks for software management projects. It was created from the extensive survey of risk maintenance literature in order to list known problems that may occur during maintenance [4]. Lópeza and Salmerona in 2012 proposes numerous failures that occur in the software maintenance. They developed a framework based on IEEE 1074 which helps the practitioners to control risks factors in software maintenance projects [5]. Elzamly in 2014 presented new techniques for quantitative and mining techniques to compare the risk management techniques to each of the software maintenance risks to identify and model if they are effective in mitigating the occurrence of each software maintenance risk in

software development life cycle [2]. This paper focused on maintenance phase that includes any future updates or expansion of the system. The aim of the present paper is to control risks including the processes, methods, and tools for managing Enterprise Systems risks in the maintenance phase before they become problems.

RISK MANAGEMENT TECHNIQUES

All projects have risks, but high-technology projects have particular risks, such as their high variation [6]. If risks are not managed properly, the quality of the final product can be compromised; customer expectations go unmet; and staff, anxious and conflicted during the life of the project, may demonstrate reduced productivity. Conceptually, from an organizational perspective, the risk arises when organizations pursue opportunities in the face of uncertainty, and constrained by capacity and costs [7]. A proper risk management program with appropriate risk management and suitable strategies can reduce to the minimum the cost and stressful problems [8]. Kercheval in 2006 applied singular value decompositions (SVD) in portfolio risk forecasts. He wants to adjust a global covariance matrix encompassing several sub-markets by individually correcting the sub-market diagonal blocks [9]. Zhou in 2009 also used singular value decompositions (SVD) AS A main technique to reduce the computation without much loss in financial risk [10]. Elzamly and Hussin in 2011 proposed a regression model techniques based on the effect size technique [7]. They also improved quality of software projects of the participating companies while estimating the quality–affecting risks in IT software projects. The results show that there were 40 common risks in software projects of IT companies in Palestine. Also, they introduced the linear stepwise discriminant analysis model to predict software risks in software analysis development process.

These methods were used to measure and predict risks by using control techniques [7]. Elzamly et al, in [7] identify software risks and controls in the software development lifecycle. They ranked the software risks factors according to their importance and occurrence frequency based on the data source. The survey questionnaire is used to collect data and method of sample selection referred to as ‘snowball’ and distribution personal regular sampling was used. Elzamly in 2014 proposed new techniques that were performed using quantitative and mining techniques to compare the risk management techniques to each of the software maintenance risks to identify and model if they are effective in mitigating the occurrence of each software maintenance risk in software development life cycle [2]. Hojjati and Noudehi in [8] applied Monte Carlo simulation in assessment of the risks. They used the quantitative evaluation of the risks of the project for Information Technology domain, and so to have a case study and use the Primavera Risk Analysis software in the quantitative Management. Paraschivescu in [11] proposed an integrated concepts in quality and risk management correlations. Gandhi in [12] developed Artificial Neural Networks (ANNs) to predict the amount of risks involved in a software project. The risks in this method can be detected well in advance and proper steps for mitigation that can be taken to increase the success rate of project. Also, Andreas in [13] investigated the applicability of Artificial Neural Networks (ANNs) to analyze survey data on the effectiveness of risk management practices in product development (PD) projects, and its ability to forecast project outcomes. They presented the relations between risk management factors affecting the success of a PD project, such as cost.

RESEARCHERID
THOMSON REUTERS

[Salman et al., 5(5): May, 2018] ISSN 2349-4506
Impact Factor: 2.785

Global Journal of Engineering Science and Research Management

METHODS OF ANALYZING THE RISK

Risk analysis is the understanding of when, where, and why risk might occur through the probability and impact of risk elements [12]. It's examining how project outcomes and objectives might change due to the impact of the risk event. Risk analysis is a component of risk management which applied to projects, information technology, security issues and any action where risks may be analyzed on a quantitative and qualitative basis. The quantitative assessment of the risk requires calculating two risk factors: the severity of the outcome of the occurrence and the probability of its occurrence. In the quantitative analysis of the risk, the whole project is simulated and the effects of each of the critical risks on the time and the overall costs will be studied [8]. In this analysis, the project activities in the scheduled plan and the risk programs are all available. Each of the activities can be probable, and a distribution function must be determined for it, and it is highly important to notice in selecting the distribution probability. The other method in risk analysis is the qualitative risk analysis. It is a project management technique concerned with discovering the probability of a risk event occurring and the impact the risk that will have if it does occur probability is the likelihood that a risk event will occur, and impact is the significance of the consequences of the risk event.

SELECTION OF MAINTENANCE RISK FACTORS

There are several risks factors in Enterprise Systems software maintenance projects which affect the software development process. Webster et al, in 2005 presented a taxonomy, listing the risk factors contained in each element of each risk class and the references where those risk factors were found. They concerned Maintenance Environment class with the project environment, where the software is being maintained. The risk factors are divided in five elements [14][4]:

i. Maintenance process: Lists the risk factors related to the definition, planning, documentation, suitability, enforcement, and communication of the methods and procedures used to maintain the product. This element contains seven risk factors.
ii. Maintenance system: Lists the risk factors related to the tools and supporting equipment used in product maintenance, such as CASE tools, simulators, compilers, and host computer systems. This element contains six risk factors.
iii. Management process: Lists the risk factors related to the planning, monitoring, and controlling of budgets and schedules; controlling factors involved in defining, implementing, and testing the product; the project managers experience in software maintenance, management, and the product domain; and the managers expertise in dealing with external organizations including customers, senior management, and other contractors. This element contains nine risk factors.
iv. Management methods: Lists the risk factors related to the methods, tools, and supporting equipment that will be used to manage and control the product maintenance, such as monitoring tools, personnel management, quality assurance, and configuration management. This element contains 14 risk factors.

v. Working environment: Lists the risk factors related to the general environment within which the work will be performed, including the attitude of people and the level of cooperation, communication, and morale. This element contains six risk factors.

Elzamly in [2] proposed the top software maintenance risk factors in software development project life cycle that most commonly used by researchers when studying the risk in software projects. The risk factors list consists of the 10 most serious risks to a project ranked from one to ten, each risk's status, and the plan for addressing each risk. These factors need to be addressed and thereafter need to be controlled. These software maintenance project risks illustrate in table 1.

Table 1. The top ten software maintenance risk factors [2]

	Software maintenance risk factors Frequency	
R1	Inadequate knowledge/skills.	11
R2	Inadequate change management.	6
R3	Corporate politics with negative effect on software project.	5
R4	Lack of resources and reference facilities.	4
R5	Lack of top management commitment and support and involvement.	4
R6	Shortfalls in externally furnished components, COTS.	3
R7	Legacy software project.	1
R8	Acquisition and contracting process mismatches.	1
R9	User documentation missing or incomplete.	1
R10	Harmful competitive actions.	1
	Total frequency	37

López and Salmeron in [5] developed a framework of software maintenance risks. This is presented in Table 2. It contains 34 risks classified according to the maintenance phase that they influence. Implement problem reporting method is the phase that contains the highest numbers of factors.

Table 2. Framework of risks in software maintenance projects [5]

A1	Be short of detailed of plan of the IS/IT.	B4	Conflict between user departments.
A2	Insufficient expertise.	B5	Inadequate plans or procedures.
A3	Insufficient/inappropriate staffing	B6	Information asymmetry.
A4	Inadequate measures, assessment tools, and simulations tools.	B7	Insufficient expertise.
A5	Changing cope/objectives.	B8	Lack of top management commitment to the project.
A6	Continuing stream of requirement changes	B9	Managing project complexity.
A7	Misunderstanding the requirements.	B10	No cooperation between departments.
A8	No explicit definition about the standard of product quality.	B11	No explicit object of the IS/IT project.
A9	The management ability of the project manager.	B12	Poor configuration control.
A10	Miscommunications or conflicting requirements.	B13	Poor implementation of safety measures.
A11	Evaluation of performance requirements.	B14	The limitation of the existing technique.
A12	Failure to manage end user expectations.	B15	Wrong IS/IT project size estimates.
A13	Ignore the matching of the technique and function.	C1	Evaluation of performance requirements.
B1	Adding unnecessary characteristics.	C2	Managing IS/IT project complexity.
B2	Be short of detailed of plan of the IS/IT project.	C3	Misunderstanding the requirements.
B3	Cannot locate or effectively manage external software development.	C4	Not explicit definition about the standard of product quality.

SINGULAR VALUE DECOMPOSITION (SVD)

Singular Value Decomposition (SVD) is a mathematical method used to analyze matrices. The risk matrix will be denoted as n×n matrix. The unique property of the SVD transform is the N2 degrees of freedom. The SVD works as an optimal matrix decomposition technique which presents the maximum signal energy in case of few coefficients.

It has the ability to adapt to the variations in local statistical form. The SVs (Singular Values) have stability, i.e. the strength of the variance (spread) on a specific axis does not vary rapidly, and sorted in decreasing order. That means the strength of each state of questionnaire risk list answer can be observed. The effects of this technique are to eliminate the unwanted data. The unwanted data problem with questionnaire is that respondents may lie due to social desirability. Most people want to present a positive image of themselves and so may lie or bend the truth. Using SVD based on questionnaires can observe the effective means by measuring the behavior, attitudes, preferences, opinions and, intentions of relatively large numbers of subjects. This technique allows to deal with the problem of large questionnaire variety of answers. The results of this technique are the eigenvectors of the optimal answer, which represent the pattern of certain questionnaire [10]. The matrix decomposition known as the Singular Value Decomposition (SVD) is one of the most powerful and useful outcome from linear algebra. Equation 1 is matrix A and is decomposed into the product of three matrices [15] [16]:

$$A = USV^T \tag{1}$$

Where U an orthogonal matrix a diagonal matrix S, and V^T are orthogonal matrices with eigenvectors of AA^T respectively. The main idea of SVD is to select only the powerful and useful outcomes from linear algebra. The orthogonal matrix shown in Figure 1 observe the first eigenvector points containing the most variance coming from other vectors jointly and represents a scalier variance of corresponding eigenvectors.

$$\begin{bmatrix} \lambda_1 \ldots & .0. & \ldots & 0 & \\ 0 \ldots\ldots & \lambda_2 \ldots & & 0 & \ldots \\ 0 \ldots\ldots & 0 & \ldots & .\lambda_n & \end{bmatrix}$$

Figure 1: The matrix of eigenvector

By this method, the exhibition of total variance will represent the data sum of all eigenvalues. The singular values are the square root of the eigenvalues. SVD is used to study the system dynamically according to the variance, and this is done by sampling system states in terms of the vector of attributes and constructs space as a two-dimension matrix [A]. The matrix A contains a set of questionnaire answers. These questionnaire include the common answers which represent the common reasons that can be used as an indicator to the risk factor. This common risk reasons will be selected as a risk pattern will do by SVD technique. The process of SVD will detect the higher eigenvalues of the inserted matrix A and risk reasons, the matrix A by removing the low eigenvalues this procedure will keep the risk factor of high eigenvalues to be corresponding patterns [15]. In this study the SVD technique was used to frequently determine the effect of each risk reason, then from this information the description for each sketch will be extracted to consider it as a pattern related to this risk factor. Equation 2 describes the array of each risk pattern:

$$A\vec{v} = \lambda\vec{v} \tag{2}$$

where A = matrix has a sequential value of risk repetition,
V = vector which represents the risk factors.
λ = the eigenvalue
The dimension of the array is the number of risk factors by the number of used questionnaire.

EVALUATION OF THE MAINTENANCE RISK FACTORS

A questionnaire was created which consisted of questions related to the selected 34 risks maintenance risk factors presented by López and Salmeron in [5]. These questions were selected using the related works by Marvin in [14] and Webster in [4]. The calculation of risk factor value involves two types of questions, positive questions which represent the questions that answered (*yes*). Also, the negative questions represent the questions that answered (*no*) [12]. It can be considered the sum of the questionnaire list of questions for each risk type is formulated as:

$$RF = \sum_{i=1}^{N} (Q_i W_i) \tag{3}$$

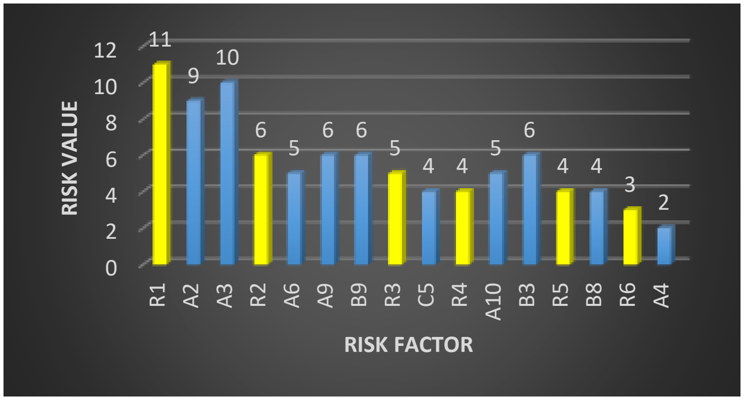
where RF is the Risk Factor Value, Q is the value of each question, W is the weight. The boundary condition is:

$$Q = \begin{cases} positive\ questions & \begin{smallmatrix} if\ yes = 1 \\ if\ no = 0 \end{smallmatrix} \\ negative\ questions & \begin{smallmatrix} if\ yes = 0 \\ if\ no = 1 \end{smallmatrix} \end{cases} \tag{4}$$

Data collection was achieved through the use of the questionnaire in order to determine risks that were common to the majority of software projects in the analyzed software companies. The 34 software risk factors in Maintenance phase were presented to respondents. The fifteen project managers that participated in this survey are coming from specific organizations in Iraq. An integrated Singular Value Decomposition with Risk Factor Value SVD-RF scheme was designed and developed based on two stages. The first stage is to find the optimal answer of the questionnaire based on SVD technique. The second stage is to calculate the risk factor using the formula above.

EXPERIMENTAL RESULTS

The surveys were completed based on 34 maintenance software risk factors presented by López and Salmeron as shown in Table 2, observe the outputs that contain the bar graph. A comparative analysis was performed with two groups, López and Salmeron [5] (in blue bars) and Elzamly [2] (in yellow bars). The plotted results shown in Figures 2 indicate that results observe the same responses of the risk factor effects. The bar graph shows that the Enterprise Systems Risk Management (ESRM) identification due to the present method and the previous studies is 96.97%. These valid results highlight the largest problem on the Enterprise Systems are (R1) risk factors which represents inadequate knowledge/skills, (A2) insufficient expertise and (A3) Insufficient/inappropriate staffing.



Figures 2: comparative analysis of risk factors

A similar analysis conducted with the 34 maintenance software risk factors presented by López and Salmeron observe the outputs as in Figure 3, 4 and 5.

ISSN 2349-4506
Impact Factor: 2.785

Global Journal of Engineering Science and Research Management

Figure 3: Identifying the Software Improvements Needs

Figure 3 shows the response of identifying the software improvements needs in Enterprise Systems Risk Management (ESRM). A2 which represent the insufficient expertise and A3 which represent the insufficient/inappropriate staffing observe the highest risk factors. These two factors concerned with establishing both the dynamics of the software development environment, which can quickly outdate the work, and the culture of the country where the study is conducted. Figure 4 and 5 represent the results of problem reporting method and Reapply of the Software Project Life Cycle (SPLCP).

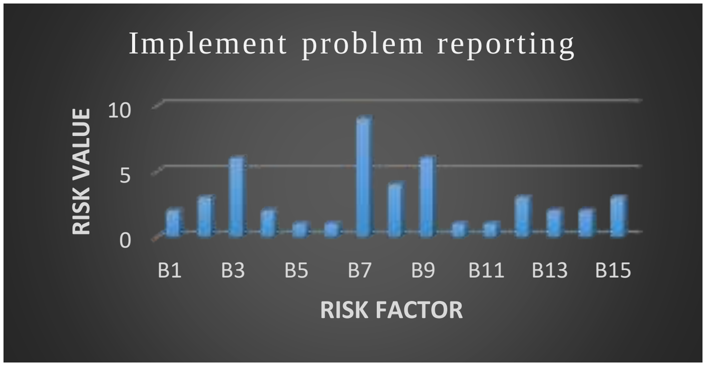


Figure 4: reporting method results

ISSN 2349-4506
Impact Factor: 2.785
Global Journal of Engineering Science and Research Management

Figure 5: Reapply of the Software Project Life Cycle (SPLCP)

The results observe a big effect by B7 and C5 which represent the insufficient expertise and difficulties in the applied software management and organization. These two factors are critical in development the Enterprise Systems Risk Management (ESRM). These two factors can provide important information regarding that ESRM's improvement and risk management practices. The higher risk in maintenance phase based on the three groups of maintenance phases came from identifying software improvements needs (A) which observe 45%, while the other two risk groups (implement problem reporting method (B), Reapply SPLCP (C)) observe 31% and 24% respectively as shown in Figure 6.

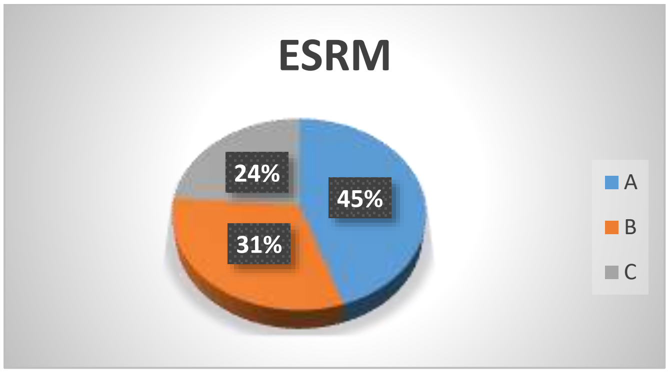


Figure 6: Identifying Risks in Maintenance Phase

CONCLUSION

In this work we apply Singular Value Decomposition (SVD) data analysis techniques to support Enterprise Systems Risk Management (ESRM). The results show that that SVD technique exhibited of total variance in common questionnaire of each of the maintenance software risk factors to model if they are effective in mitigating the occurrence of each risk factor. The comparison with the previous studies approved that this method can be used effectively to identify the risk effect in maintenance phase. The used method specify the Enterprise Systems Risk in three reasons. They are the insufficient expertise, insufficient or inappropriate staffing and difficulties in the applied software management and organization. They represent 45%, 31% and 24% respectively from total risk in the Enterprise System.

REFERENCES

[1] R. S. Kenett and O. Raphaeli, "Multivariate Methods in Enterprise System Implementation , Risk Management and Change Management," no. 2004, pp. 1–22, 2005.

[2] A. Elzamly, "Evaluation of Quantitative and Mining Techniques for Reducing Software Maintenance Risks," vol. 8, no. 111, pp. 5533–5542, 2014.

[3] N. Hrgarek, "Integrating Six Sigma into a Quality Management System in the Medical Device Industry," vol. 33, no. 1, pp. 1–12, 2009.

[4] P. B. Webster, M. De Oliveira, N. Anquetil, B. Av, and W. Norte, "A Risk Taxonomy Proposal for Software Maintenance."

[5] C. López and J. L. Salmeron, "Monitoring software maintenance project risks," vol. 5, pp. 363–368, 2012.

[6] S. M. Neves and C. E. S. da Silva, "Risk management applied to software development projects in incubated technology-based companies: literature review, classification, and analysis," *Gest. Prod., São Carlos*, vol. 23, no. 4, pp. 798–814, 2016.

[7] A. Elzamly, B. Hussin, and N. Salleh, "Top Fifty Software Risk Factors and the Best Thirty Risk Management Techniques in Software Development Lifecycle for Successful Software Projects," vol. 9, no. 6, pp. 11–32, 2016.

[8] S. N. Hojjati, "The use of Monte Carlo simulation in quantitative risk assessment of IT projects," vol. 2621, pp. 2616–2621, 2015.

[9] A. N. Kercheval, "OPTIMAL COVARIANCES IN RISK MODEL AGGREGATION."

[10] S. Project and T. Division, "All About Risk ...," 2009.

ISSN 2349-4506
Impact Factor: 2.785

Global Journal of Engineering Science and Research Management

[11] A. O. Paraschivescu, “Risk Management and Quality Management an Integrate Approach,” vol. 19, no. 1, pp. 55–61, 2016.

[12] A. Gandhi, A. Naik, K. Thakkar, and M. Gahirwal, “Risk Management in Software Development using Artificial Neural Networks,” vol. 93, no. 19, pp. 1–7, 2014.

[13] N. Andreas, “Neural network-based survey analysis of risk management practices in new product development NEURAL NETWORK-BASED SURVEY ANALYSIS OF RISK MANAGEMENT PRACTICES IN NEW PRODUCT,” 2017.

[14] M. J. Carr and S. L. Konda, “Taxonomy-Based Risk Identification,” no. June, 1993.

[15] E. Kostelich and D. Kuhl, “Introduction to SVD and Applications,” *Practice*, no. 1, pp. 1–5, 2008.

[16] J. Vandewalle and B. De Moor, “A Variety of Applications of Singular Value Decomposition in Identification and Signal Processing,” *SVD and Signal Processing: Algorithms, Applications and Architectures*. pp. 43–91, 1988.