

Exploring Risk Factors in ERP Maintenance Projects

Cristina López, *University Pablo of Olavide*, Jose L. Salmeron, *University Pablo of Olavide*

Abstract –In recent decades, companies across the world have implemented Enterprise Resource Planning (ERP) systems. The proper ERPs implementation has been a much explored issue. In contrast, ERP systems maintenance has been treated with less depth. ERP maintenance is necessary to achieve the system’s maximum performance. Nevertheless, these projects are often managed intuitively and without taking into account the existing risks. To address this need, we have studied ERP maintenance risks. In this way, we have identified and classified the risks affecting the performance of ERP maintenance projects. Moreover, we have applied a multicriteria decision making methodology to assess them. The results will help practitioners to manage ERP maintenance project risks in a better way.

Keywords –Enterprise system, Multicriteria decision making methodology, Software maintenance, Software project risks.

I. INTRODUCTION

ERP projects are never finished: the maintenance of the system starts after the implementation process. ERP maintenance process is composed of activities undertaken from the time the ERP goes live until it is retired from production. Generated benefits derive means for carrying out more efficient and effective business activities [1].

To achieve the ERP’s expected results, ERP and business process have to be completely aligned. If this fails, more damaging consequences may occur. Therefore, the ERP system maintenance is a critical issue, because if this is not carried out properly by professionals, the system will soon become useless.

Interest in ERP maintenance in scientific literature has increased in recent years [2]. In fact, the framework classifies ERP maintenance research [3]. However, ERP maintenance projects are ambiguous, because there is no clear framework that indicates the means or goals. In addition, the risks that threaten these kinds of projects are dealt with intuitively. In fact, there is no certain and structured methodology or application to avoid pitfalls and omissions [4].

Risk management shows us the real situation in which our project is, being able to manage the threats in a proactive way. If the project team identifies, assesses and treats existing risks in such projects, the probability of failure decreases. However, many companies have not maintained their ERP applications successfully. In fact, maintenance teams usually treat intuitively project risks. In contrast, ERP maintenance risks need to be properly managed. Otherwise, ERP will not attain all its potential benefits. ERP might even become useless.

Despite this, little effort to analyze ERP maintenance risks has been made in the literature. The application developed allows carrying out autonomous and continuous diagnostic analysis to identify oversight and malfunction in ERP

maintenance [5]. Nevertheless, it does not consider the existing risks in these projects. Considering this, we think that a formal study about ERP maintenance risks is valuable.

Accordingly, our goal is to rank ERP maintenance risks to establish the relative importance of each risk. To do this, we have utilized a multicriteria decision making methodology (MCDM). Prior to that, we have identified the risk that threatens ERP maintenance projects. The results indicate where the project team must focus on mitigating the risks and threats.

II. ERP MAINTENANCE RISKS

Risk management in ERP maintenance is needed due to the high risk and cost of those projects. Poor risk management of ERP maintenance projects often leads to failure. This causes huge loss for companies. The first steps in risk management is identifying and analyzing the risks in the maintenance project [6]. To support professionals’ work, we have conducted a formal study of the risks affecting the ERP maintenance. Fig. 1 shows the framework of the research.

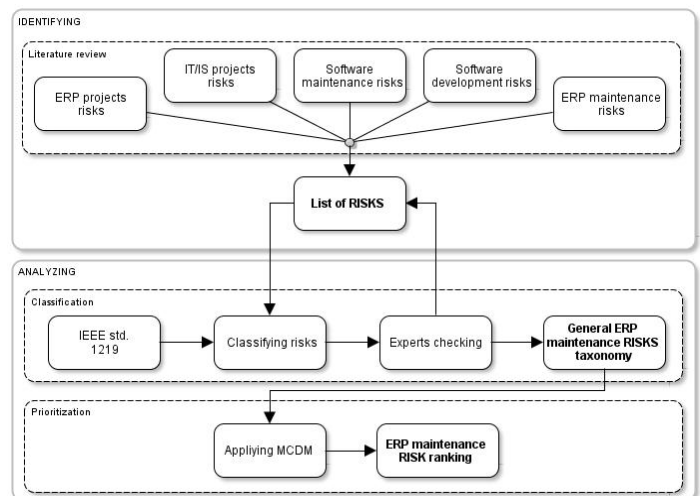


Fig. 1. Research framework

Different risks could affect the whole ERP maintenance project. The software project risk taxonomies are not completely fit to ERP maintenance because the above frameworks are very general and do not take into account the features of ERP systems. Therefore, it is necessary to create an ERP maintenance risk taxonomy. Despite the need, little effort has been made on the risks management in ERP maintenance. Indeed, ERP risks studies represent about 12% of the ERP research [7], but none of them is devoted to ERP maintenance risks. Specifically, ERP projects risks research focuses on the

pre-maintenance stage. Indeed, a model of existing risk factors in enterprise systems implementation projects has been presented [8]. It is based on Dow Corning Incorporated and Fox Meyer Drug Corporation's implementation cases. Also a risk management application has been created for the modelling, optimal adaptation and implementation of an ERP system [4].

Other studies about ERP implementation projects identify [7, 8, 9, 10] risk factors, describe [10], analyze each one based on the literature [7] and prioritize among them [9]. However, the literature only brings together three risks in the maintenance phase, and these derive from insufficient and inappropriate personnel. But, there are more risk factors that affect such projects. To increase the chances of attaining the aims of the ERP maintenance projects, it is necessary to know which risks threaten this kind of projects. Hence, we decided to identify the risks that affect these projects.

With this in mind, we carried out a broader literature review on software development, software maintenance, information technology and the ERP project risks studies. In this way, we detected that many risks are identified by several papers. So, the risk factors were checked removing duplicates. Moreover, we analyzed and eliminated those risks which do not affect ERP maintenance projects. Finally, we renamed and adapted to the study's scope the risk factors identified.

The next stage is the analysis phase. This is a process of evaluating attributes, classifying and prioritizing the risks identified [6]. The risk classification phase consists of

grouping the risks identified according to their characteristics. To do so, we have classified the risks, according to the IEEE (Institute of Electrical and Electronics Engineers) std. (Standard) 1219 [11]. This standard describes an iterative process that marks the stages in the software maintenance.

Once completed, five experts checked and validated the resulting taxonomy. The respondents are leading ERP maintenance experts. The optimal number of experts depends on the characteristics of the study itself. We can, however, say that the greater the heterogeneity of the group, the lesser number of experts is recommended; 5 being a good number [12]. A heterogeneous group is understood to be a group of people with the same knowledge but on a different social or professional scale, which is what took place in our study.

The selection of the respondents is also important. Multiple choices were contemplated. The main selection criterion considered was recognized knowledge in research topic and absence of conflicts of interest. All conditions were respected. The participating experts were not chosen just because they were easily accessible. We contacted the experts by e-mail. All experts' opinions were considered to be of the same importance.

After making corrections indicated and approval of the final taxonomy by all the experts, the general risk taxonomy was completed. In Section 3, we explain the grouping of risks. Table 1 shows the general taxonomy of ERP maintenance risks.

TABLE I
GENERAL ERP MAINTENANCE RISKS TAXONOMY

PHASE	KEY	RISK	SOURCE
Identification	R1	Changing scope/objectives	[7, 9, 13, 14, 16, 17]
	R2	Continuing stream of requirement changes	[9, 15, 18]
Analysis	R3	Evaluation of performance requirements	[15, 17]
	R4	Poor configuration control. Documentation, design, code, maintenance procedures	[13, 18]
Design	R5	Miscommunications, misunderstanding or conflicting requirements	[7, 8, 9, 13, 14, 17, 18]
	R6	Information asymmetry in development process	[10, 16]
	R7	Insufficient/inappropriate staffing and expertise	[7, 8, 9, 10, 14, 15, 17, 18]
	R8	Wrong project size estimates	[7, 8, 16, 17]
Implementation	R9	Conflict and no cooperation between user departments	[9, 13, 14, 16, 17, 18]
	R10	Absence of detailed plan of the IT project	[9, 16]
	R11	Cannot meet milestones due to management and organizational difficulties	[13,17]
	R12	The management ability of the project manager	[7, 8, 9, 10, 16, 17]
	R13	Inadequate plans or procedures	[9, 13, 17]
	R14	Lack of top management commitment to the project	[7, 8, 9, 10, 14, 16, 17, 18]

	R15	Managing project complexity	[13, 16]
Regression	R16	Ignore the matching of the technique and function	[7, 16, 18]
	R17	The limitation of the existing technique	[7, 9, 16]
	R18	Inadequate measures, assessment tools and simulations tools	[9, 13, 17]
Acceptance	R19	No explicit definition of the standard of the product quality	[10, 16, 17]
	R20	No explicit object of the IT project	[16]
	R21	Poor implementation of safety measures	[13]
Delivery	R22	Failure to manage end user expectations	[9, 14, 15, 16, 17, 18]
	R23	Adding unnecessary characteristics	[15, 18]

The risk prioritization phase consists of ranking the risks identified according to their importance. A risk is more or less important depending on its level of the risk of exposure [15]. According to this, we ordered the risks included in the taxonomy according their level of danger. For this purpose, we used a multicriteria decision-making methodology. The results show which risks are more important at each maintenance project stage, which stages are more critical and which risks are more dangerous in ERP maintenance. Therefore, the findings will indicate to practitioners which risks should be treated first.

III. RESEARCH MODEL

A. The AHP methodology

AHP methodology was presented by Saaty in 1970s [19, 20]. This is a method that uses a hierarchical structure to face a complex decision problem. AHP method has been widely used to reflect the importance, or weights, of the factors associated with priorities [21]. The MCDM represents the problem through identification of goals, criteria, and subcriteria. AHP methodology enables to prioritize the criteria included in the hierarchical model. To attain this, the hierarchical elements are evaluated and compared binarily by assigning weights.

AHP method encompasses three basic steps. Firstly, the decision problem has to be broken down into a hierarchy of interrelated elements. Secondly, the data has to be collected by pairwise comparisons of former elements and the attributes' weights on each level have to be computed using the eigenvalue method. Finally, the categories' weights have to be calculated and the final ranking obtained.

The choice of this tool has been conditioned by its multiple advantages. In the beginning, this allows breaking down and analyzing ERP maintenance risks in parts. In the application process, the tool is supported by mathematical concepts. Moreover, this permits the participation of different individuals and creates a consensus. Finally, the AHP generates a summary and offers the possibility of conducting a sensitivite analysis.

B. The AHP methodology in the research

The main goal of this research was to prioritize ERP maintenance risks. To do so, we applied AHP method. Fig. 2

shows the activities carried out within each method step. In this way, five leading ERP maintenance experts took part in the study. They checked the preliminary hierarchy and compared pairs' elements in the final hierarchy. The instrument used to collect this data was the survey. Specifically, we consulted the experts via web-based surveys.

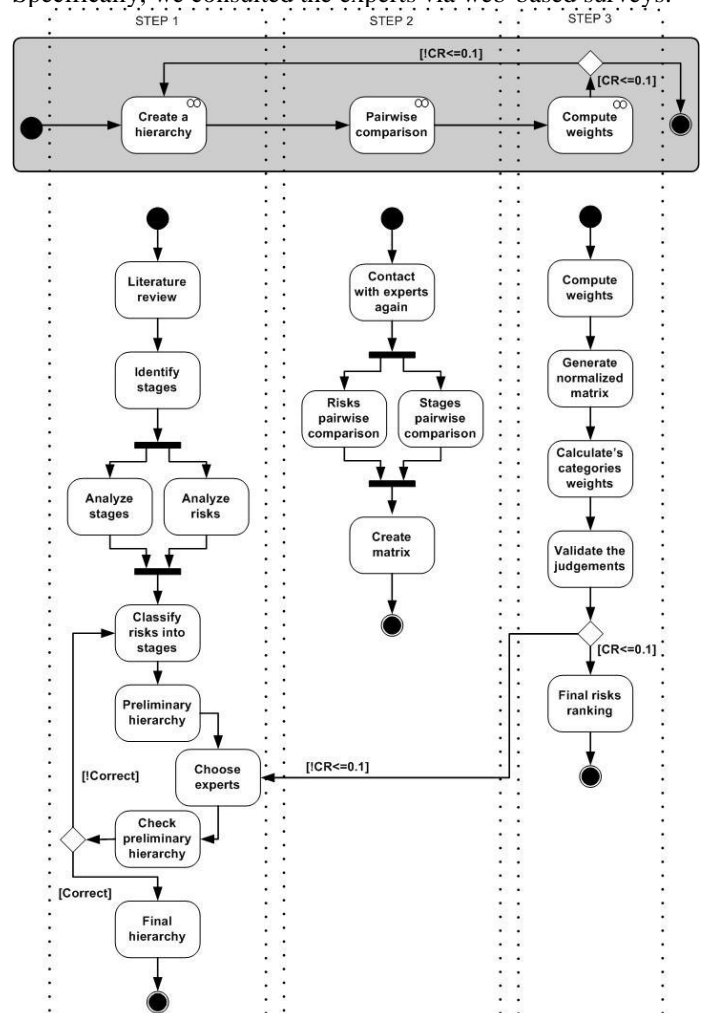


Fig. 2. Tasks in each step of the research

Constructing the hierarchy

The determination of the degree of importance associated with the risk factor can be resolved by decomposing it into subproblems within a hierarchy structure. Elements on the middle levels are the categories for evaluating those risk factors. We establish seven categories for ranking ERP

maintenance risks. Each category represents one software maintenance project phase. The phases were extracted according to the IEEE std. 1219 [11]. The third level shows the specific risks within each category.

According to the literature, we created the ERP maintenance risks hierarchy. This is represented in Fig. 3. Subsequently, this was validated by five experts. Finally, these experts indicated the importance of the ERP maintenance risks in order to rank them. This is valuable, since ERP users and experts have significantly different perception about the requirements [22].

In the following sub-sections, we explain how the risks identified make ERP maintenance projects difficult.

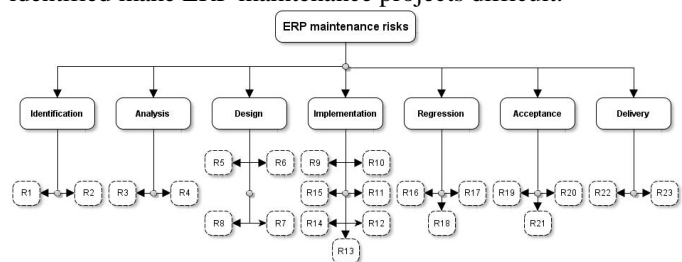


Fig. 3. Hierarchy of risks

Identification category

This stage starts with classifying the maintenance request into categories. According to the company policy, these requirements could be accepted or rejected. If these are accepted, their impact and necessary resources are estimated. Usually, changes in the environment and / or changes in business objectives generate modifications of greater impact. Higher impact requests imply higher risks. Therefore, these requirements should be assessed more carefully.

Finally, the accepted requirements are ranked and scheduled. If the flow requirements are continuous, this makes the prioritization of accepted maintenance requests difficult. Furthermore, the final planning is subject to constant change. This hinders the proper implementation of modifications and/or improvements in the ERP system.

Analysis category

In this phase of the ERP maintenance projects, ERP maintenance requests ranking are received to study their scope and feasibility. After that, a preliminary plan for design, implementation, test and delivery phase is prepared for each one.

There is a high risk of improper analysis of the requirements. This is due to the difficulty of measuring their performance and the analysis stage starts with doing this. To evaluate the requirements, a feasibility analysis is performed. This must indicate impact, safety and security implications, cost and value benefits of the modification or/and improvement.

After the feasibility analysis, a further more detailed analysis is carried out. It consists of identifying the elements of modification, firm requirements for the modification, and safety and security issues to generate an implementation plan. The conclusion of the analysis may be wrong due to a poor

configuration control, documentation, design, and code maintenance procedures.

Design category

This stage starts with identifying which ERP modules are affected by the maintenance modifications. If the requirements are miscommunication, misunderstandings or there are certain conflicts between them, the probability of identifying the modules correctly decreases. This is also affected by absence of sufficient and/or appropriate staffing and expertise in the task. Subsequently, software modules documentation that will be affected by the changes is modified. The fact that the software development process is based on information asymmetry may dull the adequate performance of this task. Finally, the list modification and plan are identified and updated. The wrong project size estimates make this task difficult.

Implementation category

Each modification assessed and accepted provokes changes in the system. These changes are introduced in the software through coding and unit testing. These tasks will be affected if the plan of the IT project is short of details. That is why the plan is the guide that marks the goal and schedule. Also the complexity of the project makes those tasks difficult, because this increases the number of elements to be taken into account in the process of programming.

When the modification is programmed, the practitioners have to integrate it in the ERP. To do so, the project manager's ability is crucial because they must detect whether the impacts of the changes are appropriate or not. This task also requires cooperation of other departments of the organization. To stimulate this support, the top management commitment is very important. Finally, a test-readiness review must be performed.

In this phase, the risk reviews are continuous, that is why the project managers must perform adequate plans and utilize adequate procedures. This task must be guided by the milestones indicated in the implementation plan. However, these cannot be met due to management and organizational difficulties.

System test category

At this point, the modified ERP is validated. Specifically, the system test checks the interface and functionality of the ERP and the practitioners try to identify possible faults in the code. Therefore, we need to know the adequate tools, techniques, functions and measurements to develop this stage successfully.

Acceptance system category

Acceptance tests can be performed by customers or ERP packages users. The check requires knowing the object of the project. This object will be compared with the final result. The execution of a proper acceptance test requires that this task is based on the standard of the product quality. This indicates the minimum requirement of the ERP. Furthermore, the

maintenance activities must be controlled with the establishment of safety measures. If these guidelines are not followed, users may accept a system that is not valid.

Delivery category

At this point, the practitioners carried out the entire task necessary to deliver the final ERP. However, it is possible that the system created does not meet users' needs or it has unnecessary characteristics. This is caused by failure to manage end user expectations.

Pairwise comparisons and computation of the factors' weights

Two stages specifically describe this phase [22]. Firstly, the computation of different weights by asking about the importance of each attribute with respect to each of the others through pairwise comparisons. The second step is computation of a vector of priorities.

In the first step, the risk factors are compared in terms of their importance within a given category. The attributes of all categories have to be compared (within their own category). We have used the widely accepted nine-point scale which is the original scale suggested by Saaty [19]. The numerical values representing the judgements of the pairwise comparisons are arranged in the upper triangle of the square matrix. For example, it represents how much criteria i is preferred over criteria j . This means that (1):

$$a_{ij} = w_i / w_j \quad (1)$$

The elements in the main diagonal of A are all equal to 1 and the elements of the down triangle are the inverse of the elements in the upper triangle. Each of its elements, a_{ij} , is the ratio of the absolute weight relative to the importance of criteria i over the absolute weight relative to the importance of criteria j . Note that the matrix is provided directly by the results of the questionnaire. Therefore, the matrix becomes (2) (3):

$$A = (a_{ij}), (i, j = 1, \dots, n) \quad (2)$$

$$A = \begin{pmatrix} 1 & \dots & a_{1j} \\ \dots & 1 & \dots \\ a_{ji} & \dots & 1 \end{pmatrix} \quad (3)$$

That is (4):

$$A = \begin{pmatrix} 1 & \dots & w_i/w_j & \dots \\ \dots & 1 & \dots & \dots \\ w_j/w_i & \dots & 1 & \dots \end{pmatrix} \quad (4)$$

Note that the elements of this matrix reflect the importance of each attribute with respect to another. However, we are interested in knowing the value of the weight of each attribute itself (the vector of priorities), not the weights when compared

to the other attributes. This is done in the next step of the analysis. Note also that this matrix verifies that (5):

$$Aw = nw \quad (5)$$

where w is the vector of the actual absolute weights and n is the number of criteria.

The whole process can be computed using any mathematical software. We used expert choice (EC) software for computing the category weights and generating a normalized matrix. EC is an AHP-based multi-objective decision support tool. This is designed for the analysis, synthesis and validation of complex individual or group decisions.

These weights must verify [21] (6):

$$Aw = \lambda_{\max} w \quad (6)$$

where λ_{\max} is the largest eigenvalue of A and w is the eigenvector associated with that eigenvalue.

The last stage is to measure the consistency of the judgements of the answers. The value $\lambda_{\max} = n$ should always be the largest eigenvalue of A . However, inconsistencies in the answers of the people interviewed may lead to a different value. The closer it is to n , the greater the consistency of the answer.

A normalized consistency ratio (CR), based on the divergence of the largest eigenvalue to n , is commonly used in the literature [21]. The closer the CR is to zero, the greater is the consistency.

The answers are consistent if the equality $a_{ij} \times a_{jk} = a_{ik}$ holds for all attributes. That is, if the transitive property holds (the preference of A over B is equal to the preference of A over C times the preference of C over B). If this equality does not hold for a given decision-maker, it means that the decision-maker is not consistent in his/her statements and the interview should be conducted again. The maximum accepted upper value for the consistency ratio is 0.1 [21]. Applying the AHP technique in this research, the value reached was less than 0.1.

IV. FINDINGS

The results obtained are shown in Tables 2 and 3. Table 2 summarizes the local weights risks for each stage in the software maintenance projects. In other words, this shows ERP maintenance risks ranking in each software maintenance projects phase. This indicates those risks affecting each stage and in which order the risks should be treated.

In the identification category, changing scope/objectives was the most critical risk with a local weight of 0.800. It was about four times greater than the continuing stream of requirement changes (0.200).

The most critical factor with a local weight of 0.667 was poor configuration control, documentation, design, code, and maintenance procedures in the analysis category. This was about two times greater than that of the evaluation of performance requirements (0.333).

Miscommunications, misunderstandings or conflicting requirements got the highest value with a local weight of 0.484 in the design category. It was about two to five times greater than insufficient/inappropriate staffing and expertise (0.231), wrong project size estimates (0.197) and information asymmetry in the software development process (0.088).

Conflict and no cooperation between user departments was the most critical risk with a local weight of 0.332 in the implementation category. This was about two to ten times greater than absence of a detailed plan of the IT project (0.062), inability to meet milestones due to management and organizational difficulties (0.156), the management ability of the project manager (0.047), inadequate plans or procedures (0.105), lack of top management commitment to the project (0.263) and managing project complexity (0.034).

Ignoring the matching of the technique and function got the highest value in the regression/system testing category. Its local weight was 0.705. This was about three to eight times greater than inadequate measures, assessment tools and simulation tools (0.211) and the limitation of the existing technique (0.084).

In the acceptance category, no explicit object of the IT project was the most critical risk with a local weight of 0.742. This was about four to nine times greater than no explicit definition about the standard of product quality (0.183) and poor implementation of safety measures (0.075).

Failure to manage end user expectations got the highest value in the delivery category. Its local weight was 0.889. This was eight times greater than adding unnecessary characteristics (0.111).

TABLE 2
SUMMARY OF LOCAL WEIGHTS

CATEGORY	KEY	LOCAL WEIGHTS
Identification Global weight: 0.398 (1)	R1	0.800 (1)
	R2	0.200 (2)
Analysis Global weight: 0.094 (4)	R3	0.333 (2)
	R4	0.667 (1)
Design Global weight: 0.128 (3)	R5	0.484 (1)
	R6	0.088 (4)
	R7	0.231 (2)
	R8	0.197 (3)
Implementation Global weight: 0.216 (2)	R9	0.332 (1)
	R10	0.062 (5)
	R11	0.156 (3)
	R12	0.047 (6)
	R13	0.105 (4)
	R14	0.263 (2)
	R15	0.034 (7)
Regression/ system testing Global weight: 0.061 (6)	R16	0.705 (1)
	R17	0.084 (3)
	R18	0.211 (2)
Acceptance testing Global weight: 0.065 (5)	R19	0.183 (2)
	R20	0.742 (1)
	R21	0.075 (3)
Delivery Global weight: 0.037 (7)	R22	0.889 (1)
	R23	0.111 (2)

Table 2 also shows a global weights-based software maintenance project phases ranking. This indicates which stages are most critical. The managers thus have a guide that shows where they should focus their strengths.

The identification category was the most critical phase. Its global weight was 0.398. This was about two to ten times greater than analysis (0.094), design (0.128), implementation (0.216), regression/system testing (0.061), acceptance testing (0.065) and delivery (0.037).

Table 3 shows the global weights-based risks ranking. In other words, the risks are ranked from highest to lowest in terms of causes of the failure. The ranking is based on global weights. They have been calculated by multiplying the local weights of each risk by the global weight of each category. By doing this, each local risk is balanced by the importance of the category to which it belongs.

Changing scope/objectives was the most critical factor with a global weight of 0.319. This was from four to eighty times greater than the rest. The second risk was continuing stream of requirement changes (0.080). The third was conflict and no cooperation between user departments (0.072). The fourth was poor configuration control; documentation, design, code, maintenance procedures (0.063). The fifth was miscommunications, misunderstandings or conflicting requirements (0.062). These five risks add up to 60% of the global total weight.

TABLE 3
RISKS RANKING WITH GLOBAL WEIGHTS

RISKS	GLOBAL WEIGHTS	CATEGORY
1. R1	0.319	Identification
2. R2	0.080	
3. R9	0.072	Implementation
4. R4	0.063	Analysis
5. R5	0.062	Design
6. R14	0.057	Implementation
7. R20	0.048	Acceptance testing
8. R16	0.043	Regression/ system testing
9. R11	0.034	Implementation
1. R22	0.033	Delivery
1. R3	0.031	Analysis
1. R7	0.030	Design
1. R8	0.025	
1. R13	0.023	Implementation
1. R10	0.013	
1. R18	0.013	Regression/ system testing
1. R19	0.012	Acceptance testing
1. R6	0.011	Design
1. R12	0.010	Implementation
2. R15	0.007	
2. R21	0.005	Acceptance testing
2. R17	0.005	Regression/ system testing
2. R23	0.004	Delivery

V. CONCLUSIONS

The present research seeks to help practitioners to manage the existing risks in ERP maintenance projects. With this in mind, we propose a general ERP maintenance risks taxonomy. This indicates the risks faced by managers in each ERP maintenance projects stage. Practitioners can thus plan and carry out ERP maintenance, controlling the critical threats. Furthermore, we have ranked risks classified in the taxonomy. To do so, a multicriteria decision-making tool called AHP was used. We thus indicate which ERP maintenance project phases and risks are critical.

The maintenance stages have been ranked from more to less problematic. One the most highlighted findings is that the first four stages are the most critical. In fact, the first is the most risky. The most important risks in each maintenance phase have also been identified. For example, in the design stage, the most critical risk is miscommunications, misunderstandings or conflicting requirements.

Finally, ERP maintenance risks have been prioritized from more to less critical. The two most important risks are changing scope and objectives and the continuing stream of requirement changes. Both have been classified in the identification stage. Hence, managers should focus their efforts on controlling them. The final results also highlight that the proper management of ERP maintenance requirement is essential for the success of the ERP maintenance. In fact, we can observe that four out of five more critical risks (R1, R2, R9 and R5) are closely related with this issue.

Therefore, the findings of this study can also help the professionals to carry out effective risk management in the entire process of ERP maintenance. But they should obtain more information about the risks for this purpose. In this way, the professionals also need to know how the risks arise. In this sense, we believe that studies about the ERP maintenance risk dimensions are also necessary.

For academics, the taxonomy provides a ground for further studies because ERP maintenance risks have been identified, classified and prioritized reliably for the first time.

REFERENCES

1. **Ng, C.S.P., Gable, G.G., Chan, T.** A ERP client-benefits-oriented maintenance taxonomy. *Journal of Systems and Software*, 2002, vol. 64, p. 87-109.
2. **Botta-Genoulaz, V., Millet, P.A., Grabot, B.** A survey on the recent research literature on ERP systems. *Computer in Industry*, 2005, vol. 56, p. 510-522.
3. **Gable, G.G., Chan, T., Tan, W.-G.** Large packaged application software maintenance: a research framework. *Journal of software maintenance and evolution: research and practice*, 2001, vol. 13, p. 351-371.
4. **Zafeiropoulos, I., Metaxiotis, K., Askounis, D.** Dynamic risk management system for the modeling, optimal adaptation and implementation of an ERP system. *Information management & Security*, 2005, vol. 13, p. 212-234.

5. **Kwon, O.B., Lee, J.J.** A Multi-agent intelligent system for efficient ERP maintenance. *Expert System with Applications*, 2001, vol. 21, p. 191-202.
6. **Dorofee, A.J., Walker, J. A., Alberts, C.J., Higuera, R.** Continuous Risk management guidebook. SEI, Carnegie Mellon University, Pittsburgh, 1996.
7. **Aloini, D., Dulmin, R., Mininno, V.** Risk management in ERP project introduction: Review of the literature. *Information & Management*, 2007, vol. 44, p. 547-567.
8. **Scott, J.D., Vessey, I.** Managing Risks in Enterprise Systems Implementations. *Communications of the ACM*, 2002, vol. 45, p. 74-81.
9. **Huang, S.-M., Chang, I.-C., Li, S.-H., Lin, M.-T.** Assessing in ERP projects: identify and prioritize the factors. *Industrial management & data systems*, 2004, vol. 8, p. 681-688.
10. **Sumner, M.** Risk factors in enterprise-wide/ERP projects. *Journal of Information Technology*, 2001, vol. 15, p. 317-327.
11. **IEEE std 1219.** *IEEE Standard for Software Maintenance*, 1998.
12. **Clayton, M.** Delphi: a technique to harness expert opinion for critical decision-making tasks in education. *Educational Psychology, Dorchesteron-thames*, 1997, vol. 17, p. 373-387.
13. **Sherer, S.A.** The three dimensions of software risk: technical, organizational, and environmental. **In:** Twenty-Eighth Hawaii International Conference, Hawaii, USA, 1995.
14. **Keil, M., Cule, P., Lyytinen, K., Schmidt, R.** A framework for identifying software project risk. *Communications of the ACM*, 1998, vol. 41, p. 76-83.
15. **Boehm, B.W.** Software risk management: principles and practices. *IEEE Software*, 1991, vol. 8, p. 32-41.
16. **Lu, X.N., Ma, Q.C.** Risk analysis in software development project with owners and contractors. **In:** Engineering Management Conference, 2004, p. 789-793.
17. **Zhou, L., Vasconcelos, A., Nunes, M.** Supporting decision making in risk Management through an evidence-based information Systems Project risk checklist. *Information Management & Computer Security*, 2008, vol. 6, p. 166-186.
18. **Batista, K.P., Olivera, K.M., Anquetil, N.** A risk taxonomy proposal for software maintenance. **In:** 21st IEEE International Conference on Software Maintenance, 2005, p. 453-461.
19. **Saaty, T.L.** A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 1977, vol. 15, 234-281.
20. **Saaty, T.L.** *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill, New York, 1980.
21. **Zahedi, F.** The analytic hierarchy process, A survey of the method and its applications. *Interfaces*, 1986, vol. 16, 96-108.
22. **Yang, C., Huang, J.B.** A decision model for IS outsourcing. *International Journal of Information Management*, 2000, vol. 20, 225-239.

Cristina Lopez is a Lecturer at the School of Engineering at the University Pablo of Olavide. She also participates in various research projects at the national level. Recently, she finished her PhD on risk management in software projects. Her papers have been published in IEEE transactions on software engineering and Journal of Systems and Software. Her research interests include decisions support systems, ERP systems and risk management. Contact her at clpvar@upo.es.

Jose L. Salmeron is an associate professor at the School of Engineering at the University Pablo of Olavide. Dr. Salmeron has co-authored eight books and has worked as a visiting scholar and researcher at the Texas Tech University. At present, he is leading research projects at the national and international level. His papers have been published in Communications of the ACM, Journal of Systems and Software, Computer Standards & Interfaces, Expert Systems with Applications and so on. He is currently carrying out research in the area of Computational Intelligence, Complex Adaptive Systems, and Technological Foresight. Contact him at salmeron@upo.es