European Journal of Operational Research 220 (2012) 484–495

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Decision Support

Modelling and assessing ERP project risks: A Petri Net approach

Davide Aloini [*], Riccardo Dulmin, Valeria Mininno

Department of Energy and System Engineering, University of Pisa, Largo Lucio Lazzarino 1, 56126 Pisa, Italy.

ARTICLE INFO

Article history:
Received 10 February 2011
Accepted 31 January 2012
Available online 7 February 2012

Keywords:
ERP project
Risk assessment
Petri Nets
Risk interdependencies

ABSTRACT

Risk management of Enterprise Resource Planning (ERP) projects is largely recognized as a very complex task both by academics and practitioners. Strict interconnections among risk factors often occur so that indirect effects on the overall project performance are very likely. Unfortunately, the implications of interdependency are usually underestimated by project managers and decision makers since they are difficult to include in any risk assessment logic. This work shows how Colored Petri Nets (CPNs) can be used to model risk factors in ERP projects in order to deal with the problem of interdependence in risk assessment. The technique is presented through an application to a real case study. Findings highlight the importance of interdependence and the indirect links for an effective ranking of risks. Furthermore, results emphasize the valuable support of CPNs in risk factor modelling since they allow both a more structured and systematic risk analysis and a more accurate planning for effective risk treatment actions.

1. Introduction

Enterprise Resource Planning (ERP) programs are integrated computer-based systems used to manage company-wide business processes. Their aim is to support the flow of information between all the business functions inside/outside the boundaries of the organization (Bidgoli and Hossein, 2004).

Implementing an ERP software is usually a complex and risky project that usually impose significant changes on business processes and work practices (Mandal and Gunasekaran, 2003). Many factors contribute to the success of the introduction project (Wright and Wright, 2001; Aloini et al., 2007) but just a few of them are technology-related factors. Many others belong to different areas such as strategic planning, project management, communication and financial issues.

Risk management is considered as a possible way to support the introduction of complex information systems. Several authors (Wright and Wright, 2001; Kumar 2002; Tatsiopoulos et al., 2003; Kweku-Muata and Ngwenyama, 2006; Aloini et al., 2007) studied risk management approaches in the ERP field in order to support the implementation project and improve the success rate of these projects. According to their perspective, risk management in ERP projects can enable the achievement of an appropriate management of all the sources of uncertainty within the project.

This work presents an innovative application of Coloured Petri Nets (CPNs) to the risk assessment (RA) of an ERP project. It aims to provide a methodology to include risk interdependence in the RA process by modeling and analyzing causal relationships among the risk factors and between the risk factors and effects. To our best knowledge, this is still a major gap in literature. Chapman and Ward (2003) stated that the most common shortcomings in terms of potential cause of failure in risk management are often about a superficial risk analysis which misses risk interdependence analysis; this is also valid within the ERP case. The complex structure of an ERP project and the high number of risk factors indeed increase the magnitude of risks not only in relation to each single factor, but also to the interconnections between them. Such project risks are typically interdependent and the antecedent to failure may be quite distinct and distant from actual disaster, since interdependence does not require proximity. For these reasons, before any risk assessment process, first it is essential to understand and model risk factors including interdependencies, in order to finally allow a valuable ranking and an effective guide to project risk management activities.

Given this gap, the main objective of this work is to develop a quantitative approach using CPNs to model ERP project risks and quantify the project vulnerability for each risk factor including their interdependencies. We show how the use of CPNs can allow to include interdependences and indirect effect causalities into the final risk evaluation algorithm. This is a major benefit in the field of ERP risk management since it enables the elaboration of new quali-quantitative risk indexes for project management and control.

The paper is organized as follows: Section 2 introduces a brief background of ERP projects and related risks. CPNs and their application to ERP risk management field are discussed in Section 3. Section 4 presents an exemplificative case study; a Petri Nets model is here used to analyze project risks and their impact on

[*] Corresponding author. Tel.: +39 050 2217347; fax: +39 050 2217333.
E-mail addresses: davide.aloini@dsea.unipi.it (D. Aloini), riccardo.dulmin@dsea.unipi.it (R. Dulmin), valeria.mininno@dsea.unipi.it (V. Mininno).

0377-2217/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2012.01.062

the performance. Company and project description, data collection process, modelling of risk factors are here reported. Results are provided in Section 5. Finally, conclusions, limitations and potential improvements are described in Section 6.

2. Risks in ERP projects

2.1. Project life cycle

According to the target application, researchers have differently described the ERP project life cycle. Some models are characterized by few general phases such as the Deloitte Consulting's model (1999) while others are more analytic, such as Markus and Tanis (2000), Ross and Vitale (2000) or Rajagopal (2002). The following Implementation Roadmap is one of the most used framework (Fig. 1) (Monk and Wagner, 2006). For the purpose of this article, we adopted this latter roadmap.

The model includes five phases:

"*Project Preparation*" is about the project approval and funding process; it starts defining the project scope (goals and objectives). Tasks usually include: the kick-off meeting, defining the project team/manager, selecting the vendor/package, identifying and prioritizing the interested business processes, communicating the objectives and impacts of the new system, evaluating the investment and fixing the budget, starting a risk management process.

"*Business Blueprint*" develops a detailed documentation about how the business processes should be managed and supported by the ERP system. Typical tasks include: developing a project plan, defining the key users (KUs), educating the KUs and the project team, gaining the supporting skills, completing a Gap Analysis in order to decide if changing the way the firm leads the business to fit the software (*compromising*) or customizing the ERP to suit the business processes (*customization*), managing the legacy systems. During this phase, the technical team also determines the management strategy for the legacy systems (interfaces, data transfer, applications to be replaced).

"*Realization*" covers several activities: system configuration, hardware – network connection, reengineering of processes, execution of the change management plan. Typical tasks are: system integration, data clean-up and conversion, education of KUs/IT staff/executive, development of a system prototype.

"*Final Preparation*" includes: testing the system on critical processes, conducting end-users training, setting up the Help Desk, bugs – fixing, tuning and optimization of data and parameters, ending the data migration from legacy systems, setting the Go Live. Test sessions are usually conducted in order to identify any discrepancies or bugs and solve them. Legacy system data are converted and migrated to a new database. The end-user training is started during the last testing phase, in order to prepare the Roll-out phase. The discrepancies and bugs are solved and the solutions checked. Finally, user profiles are established and end-user training completed.

"*Go Live & Support*" starts from system activation and ends when "normal operations" have been achieved. Typical tasks are: Help Desk, final bug fixing, monitoring the operative performances, modifications to hardware capacity and software system. It includes the cut-off of any legacy systems. In this phase, all the transactions are usually stopped in order to preserve data integrity until data migration is complete. A support system is also provided to help users in the post-implementation activities. System performance is systematically monitored and continuous improvement actions are planned and carried out.

2.2. Risk factors and effects

Factors affecting an ERP implementation project spread around all the project phases (Al-Mashari et al., 2003). Many empirical researches have focused the attention on risk identification and classification (Sumner, 2000; Tatsiopoulos et al., 2003; Wright and Wright (2001); Scott and Vessey, 2002a; Yang et al., 2006). Aloini et al. (2007) especially reviewed a large number of articles about ERP system implementation from a risk management perspective. They identified 19 risk factors (Table 1) and 10 project effects (Table 2) to which we mainly refer to in this work.

2.3. Risk management

Applying a risk management process to ERP projects has a strategic value since it can achieve to formulate the appropriate risk treatment strategies and actions early in the project. This can be really critical for a successful implementation.

According to the Project Management Association, a typical risk management process goes through the following four steps: context analysis; risk assessment, risk treatment; and risk control. A broader consensus exists on this subject. We consider this approach suitable for ERP project as well.

1. *Context analysis* – aims to define the boundaries of the Risk Management processes in order to support the definition of the correct risk model approach.
2. *Risk assessment* – allows the organization to early determine the potential threats (*risk identification*) and their impacts (effects) on the project. It also includes prioritizing risk factors according to their risk levels (*risk quantification*) by a *risk analysis* – which provides the inputs to the process (e.g. the occurrence probability of a risk factor, the factor interdependences, the links with potential effects, the severity of these effects and the detection difficulty) and a *Risk Evaluation* – which defines risk classes.
3. *Risk treatment* – targets towards the selection of an effective strategy to manage the risks related to the different identified risk classes.
4. *Risk control* – includes *monitoring and review risks* and *communication and consulting* (effectively communicate hazard in order to support the managerial actions).

Formal, structured risk assessment methods are seldom if ever applied to risk management in complex IT projects, such as the introduction of ERP systems. In a specific review on ERP "risk management", Aloini et al. (2007) stated that most of the contributions were focused on the risk identification and risk analysis in a rather

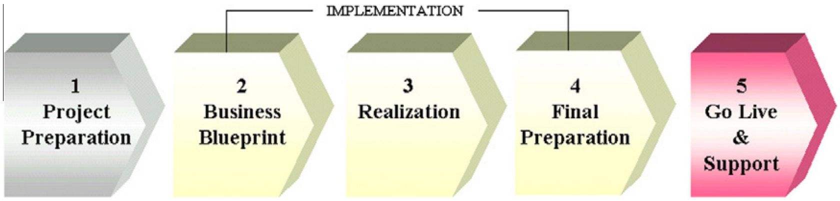

Fig. 1. ERP project life cycle (SAP implementation roadmap).