CEPIS UPGRADE is the European Journal for the Informatics Professional, published bi-monthly at <http://cepis.org/upgrade>

Publisher
CEPIS UPGRADE is published by CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>), in cooperation with the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*, <http://www.ati.es/>) and its journal Novática

CEPIS UPGRADE monographs are published jointly with Novática, that publishes them in Spanish (full version printed; summary, abstracts and some articles online)

CEPIS UPGRADE was created in October 2000 by CEPIS and was first published by Novática and INFORMATIK/INFORMATIQUE, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies)

CEPIS UPGRADE is the anchor point for UPENET (UPGRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
- inforewiew, magazine from the Serbian CEPIS society JISA
- Informatica, journal from the Slovenian CEPIS society SDI
- Informatik-Spektrum, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
- ITNOW, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
- Mondo Digitale, digital journal from the Italian CEPIS society AICA
- Novática, journal from the Spanish CEPIS society ATI
- OCG Journal, journal from the Austrian CEPIS society OCG
- Pliroforiki, journal from the Cyprus CEPIS society CCS
- Tölvumál, journal from the Icelandic CEPIS society ISIP

Editorial correspondence: Llorenç Pagés-Casas <pages@ati.es>
Advertising correspondence: <info@cepis.org>

Subscriptions
If you wish to subscribe to CEPIS UPGRADE please send an email to info@cepis.org with 'Subscribe to UPGRADE' as the subject of the email or follow the link 'Subscribe to UPGRADE' at <http://www.cepis.org/upgrade>

ISSN 1684-5285

CEPIS UPGRADE
The European Journal for the Informatics Professional
http://cepis.org/upgrade

Vol. XII, issue No. 5, December 2011

Farewell Edition

./..

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

CEPIS UPGRADE
The European Journal for the Informatics Professional
http://cepis.org/upgrade
Vol. XII, issue No. 5, December 2011

Farewell Edition

Cont.

UPENET (UPGRADE European NETwork)

CEPIS NEWS

Selection of Project Alternatives while Considering Risks

Marta Fernández-Diego and Nolberto Munier

The selection of projects consists in choosing the most suitable out of a portfolio of projects, or the most fitting alternative when there are constraints in regard to financing, commercial, environmental, technical, capacity, location, etc.. Unfortunately the selection process does not place the same importance on the various risks inherent in any project. It is possible however, to determine quantitative values of risk for each pair of alternative/threat in order to assess these risk constraints.

Keywords: Free Software, Linear Programming, Project, Risk Management, Threat.

1 Introduction

Failing to satisfy project objectives is a major concern in project management. Risks can generate problems with consequences that are not often considered, and indeed, in many cases risk management is not even taken into account [1]. However, the benefits of risk management are considerable. Risk management allows, at the beginning of the project, the detection of problems that could be otherwise ignored, and so effectively to help the Project Manager in delivering the project on time, under budget and with the required quality [2]. However, if risk management is not performed along the whole project, the Project Manager probably will not be able to take advantage of its full benefits.

This paper proposes a methodology that consists in building, from a final value of risk for each project pair (threat or alternative) and a decision matrix to determine, using Linear Programming (LP), which is the most effective alternative considering the risks. Of course, in a real case these same constraints, plus others, can be added to the battery of constraints that address environmental matters, economic, technical, financial, political, and so on. The result will reflect the best selection on the basis of all the constraints considered simultaneously.

The application of LP to this decision-making problem is new in the treatment of risk. It opens a series of possibilities in the field of risk management in such a way that this methodology represents more accurately than other methods a project's features, solving problems with all kinds of constraints, including those related to risk, and therefore placing risks at the same level as the economic, social and environmental constraints normally considered, with the idea of raising the discipline of risk management in projects. In short, although an even higher level of organizational maturity in terms of risk management would correspond to the integrated risk management of the portfolio of projects, it is expected that the outcome will be projects driven by risk management [3].

Authors

Marta Fernández-Diego holds a European PhD in Electronic and Telecommunications Engineering. After some research and development contracts in universities and multinational companies from France, Great Britain and Spain, she is currently a lecturer in the Department of Business Organization at *Universitat Politècnica de València*, Spain, where she teaches project risk management, among other subjects. <marferdi@doe.upv.es>.

Nolberto Munier is a Mechanical Engineer, Master in Project Management and PhD in Design, Manufacturing and Management of Industrial Projects. He has worked extensively in linear programming techniques and applied them to solving decision problems in urban projects in several cities in different countries. In addition, he developed a methodology called Simus for solving problems of a complex nature, with multiple objectives and with any type of constraints. He is currently an international consultant on issues of urban and regional planning. <nolmunier@yahoo.com>.

The paper presents in the next section an application example. The following describes in detail the characteristics of the problem to determine the choice of one alternative or another according to various criteria, along with its constraints. Finally, once the problem is solved by LP, the results are discussed.

> “ This paper proposes a methodology that consists in building the most effective alternative considering the risk ”

> “Failing to satisfy project objectives is a major concern in project management”

2 Application Example

2.1 Background

In the past decade, free software has exploded, even challenging the inertia that still exists in software engineering, mainly derived from proprietary software, resulting in new business models and product offerings that enable real choice for consumers.

To understand free software, let us begin by clarifying that the fundamental characteristic of proprietary software is that all ownership rights therein are exclusively held by the owner, as well as any possibility of improvement or adaptation. The user merely pays for the right to use the product, rather than buyiong it outright.

The problems associated with software, regardless of whether free or proprietary, lie in its own nature. The key problem addressed by free software is precisely the possibility of reusing it, in the logical sense that you can use parts already coded by others and create derivatives. For any transformation of a person´s work authorization of the copyright holder is required. Instead of using the simple copyright of the proprietary software licenses which means “all rights reserved”, these other free software licenses only reserve some rights, and report whether or not to allow the user to make copies, create derivative works such as adaptations or translations, or give commercial uses to the copies or derivatives.

In contrast, the essential feature of free software is that it is freely used [4]. Specifically, it allows the user to exercise four basic freedoms. These freedoms are:

- The freedom to run the program for any purpose,
- The freedom to study how the program works, and change it to make it do what you wish,
- The freedom to redistribute copies,
- The freedom to improve the program, and release your improvements to the user.

Open source code[1] is required to meet these freedoms. With open source code we mean that the source code is always available with the program. In addition, the exercise of these freedoms facilitates software evolution, exposing it as much as possible to its use and change – because greater exposure means that the software receives more testing – and by removing artificial constraints to the evolution – being more subject to the environment.

2.2 Background of the Case: Alternatives and Objective

Considering the future commercialization of computer models with free software preinstalled, an entrepreneur, who plans to start a small business, analyzes the possibility of buying for his business computers with free software installed. Given this possibility, he needs to make a decision between both alternatives, that is, proprietary software or free software according to risk criteria, with the objective consisting in minimizing the total cost, taking into account an estimated difference of 100• favoring a computer with free software operating system.

[1] Text written using the format and syntax of the programming language with instructions to be followed in order to implementthe program.

> “If risk management is not performed along the whole project, the Project Manager probably will not be able to take advantage of its full benefits”

	x_1 (Free software)	x_2 (Proprietary software)	Action	Operator	B Threshold
Resistance to change	0.85	0.15	MIN	≥	0.15
Dependency	0.16	0.64	MIN	≥	0.16
Lack of security	0.125	0.375	MIN	≥	0.125

Table 1: Characteristics of the Problem.

> “There is a widely held belief that free software operating systems are inherently more secure than a proprietary one”

3 Problem Characteristics

The characteristics of the problem are summarized in Table 1.

The options and constraints of the problem, reflected in this table, are explained in the following points.

3.1 Criteria

This case raises two alternatives, effectively two projects which will be analyzed on the basis of criteria that take into account the various risks covered by both projects. Specifically, we consider three selection criteria that correspond to three of the potential threats related to the software, and which are mirrored in the main differences between free software and proprietary software. Of course, in a real case there may be many other criteria related to the economy, availability and experience of personnel, environment, etc., but all are considered simultaneously, together with the risk criteria. Therefore, the alternatives or options will have to comply simultaneously with all factors.

The risk in a project involves a deviation from its objectives in terms of the three major project success criteria; schedule, cost and functionality. In this sense, risk indicates the probability that something can happen which endangers the project outcome.

Risk can be measured as the combination of the probability that an incident occurs and the severity of impact [5]. Mathematically, risk can be expressed as follows:

$$Risk = Likelihood\ x\ Impact \quad (1)$$

In the certainty of the materialization of the threat, the risk would be equal to the impact; if the probability of the threat materialization is zero, then there is no risk at all. However, risk is a combination of both probability and impact, and in statistics, risk is often modeled as the expected value of some impact. This combines the probabilities of various possible threats and some assessment of the corresponding outcomes into a single value. Consequently, each pair threat contributes partially to this expected value or risk.

The threats considered, which appear as rows in Table 1, are as follows:

- **Resistance to change**

It is clear that there is still a lot of inertia and reluctance to move from the proprietary model, and despite the advantages of free software, this is the main barrier. Inertia is the resistance of the user to give up something he knows (proprietary software), i.e. there is a resistance to change (to free software), supported by the other side in the laws of physics (e.g., resistance to initiate a movement).

Although the data are dependent on many factors including company size, the software’s purpose, its scope, field of application, etc., 85% of small businesses would opt for proprietary software products by inertia, lack of knowledge about free software alternatives or simply the fear of moving to a new field, compared with 15% who would venture into something new. Therefore the likelihood of resistance to change for free software is higher (85%) than the one for proprietary software (15%). On the other hand, we consider both the impact is total, i.e. 100%, since what is at stake is the choice of an alternative or another.

- **Dependency**

A non-technical advantage of free software is its independence from the supplier, ensuring business continuity even though the original manufacturer disappears.

Initially, free software arose from abusive practices used by leading developers of proprietary software, which requires users to permanently buy all updates and upgrades; in this sense the user has their hands tied since they have very limited rights on the product purchased. But when companies turn to free software, they liberate themselves from the constraints imposed by the software vendor. Indeed, free software appears to ensure the user certain freedoms.

In addition the user is dependent not only on the manufacturer, but also on the manufacturer’s related products. The product often works best with other products from the same manufacturer. With free software, however, users have the power to make their own decisions.

To simplify the problem equal values of probability and impact have been considered, resulting in a dependency risk of 16% for free software and 64% for proprietary software.

- **Lack of security**

There is a widely held belief that free software operating systems are inherently more secure than a proprietary one because of their Unix heritage, which was built specifically to provide a high degree of security. This statement can be justified as follows:

On the one hand, a coding error can potentially cause security risk (such as problems due to lack of validation). Free software is higher quality software, since more people can see and test a set of code, improving the chance of detecting a failure and to correct it quickly. This means that quality is assured by public review of the software and by the open collaboration of a large number of people. This is why free software is less vulnerable to viruses and malicious attacks.We could estimate that the vulnerability of

> “The main advantage of Linear Programming is that it is possible to represent real world scenarios with some degree of accuracy”

free software against security issues is 25%, while for proprietary software such vulnerability amounts to 50%.

On the other hand, the impact of a security problem is generally lower in the case of free software, because these bugs are usually addressed with speedy fixes wherever possible because of an entire global community of developers and users providing input. In contrast, in the world of proprietary software, security patches take considerably longer to resolve. We might consider impacts 50% for free software and 75% for proprietary software.

In short, considering risk as a combination of vulnerability and impact, the risk due to lack of security results in 12.5% for free software versus 37.5% for proprietary software.

Furthermore, since in fact transparency hinders the introduction of malicious code, free software is usually more secure.

3.2 Constraints

Since in the three cases we are talking of negative events, or threats, and we have not considered any opportunity, the constraints that we impose on these criteria respond to minimization, effectively finding a solution greater than or equal the value of minimal risk, since we cannot find a solution with lower risk than this.

For example, the opposite of resistance to change could have been considered. The term inertia may refer to the difficulty in accepting a change. While not applying any force, we follow our own inertia, which is an opportunity for the favored option. In this approach, the appropriate action had been to maximize, or find a solution less than or equal to the maximum benefit because we cannot find a solution with greater benefit.

4 Linear Programming Resolution

The matrix expression of the LP problem is as follows:

$$A \,.\, X = B \qquad (2)$$

Where:

$$A = \begin{pmatrix} 0.85 & 0.15 \\ 0.16 & 0.64 \\ 0.125 & 0.375 \end{pmatrix}$$

is the decision matrix, shown boxed in Table 1.

The components A_{ij} of this matrix are the values of risk that each threat brings for each alternative.

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

is the vector of unknowns, i.e. the option to choose in this case.

is the vector of thresholds, i.e. the limits of each constraint according to the discussion in Section 3.2.

To meet the objective of minimizing the objective function Z, this objective function is expressed as the sum of the products between the cost of each alternative for the value of each of them (i.e. the unknown X represents what we wish to determine).

Thus, assuming that the cost of a computer with free software operating system preinstalled is 600 • and the one for proprietary software is 700 •, the objective function is:

$$Z = 600x_1 + 700x_2 \ (Minimize) \qquad (3)$$

Applying the LP simplex method [6] which is essentially a repeated matrix inversion (2) according to certain rules, one gets, if it exists, the optimal solution of the problem. That is, the best combination or selection of alternatives to optimize the objective function (3).

5 Discussion of Results

5.1 Optimal Solution

The optimal solution to the LP problem is as follows:

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.125 \\ 0.292 \end{pmatrix} \qquad (4)$$

We choose the higher value because, although both contribute to obtaining the goal, if only one option is actually possible it is clear that the one with the higher value contributes more efficiently than the other and therefore is chosen.; In the case above, proprietary software (x_2) should be chosen.

In our case both values are very close but since the LP indicates that the alternative with proprietary software contributes more efficiently to the objective, taking into account risk constraints, it is chosen.

5.2 Dual Problem

Every direct LP problem, such as this one, can be converted into *‘his image’*, which is called the *‘dual problem’*. In the dual problem the columns represent threats while the rows represent the alternatives. While the direct problem variables indicate which option contributes best to the goal, the dual problem variables provide us with the values of the *‘marginal contributions of each constraint’* or *‘shadow prices’*, which is an economic term. In essence, this means knowing how much the objective function changes per unit variation in a constraint, which ultimately gives an idea of the importance of each constraint.

In this case we obtain the results shown in Table 2.

> “Another major advantage of LP is that, if there is a solution, this is optimal. i.e. the solution cannot be improved”

	Equal value	Marginal value
Lack of security	0.125	1683.333
Dependency	0.207	0,000
Resistance to change	0.150	458.333

Table 2: Equal Value and Marginal Value.

It turns out that the problem of lack of security is the most decisive in the choice of the alternatives, while resistance to change comes in a second place, which intuitively might be seen as the most decisive.The problem of dependency does not affect the solution since its marginal value is zero.

This powerful tool will allow, for example, a discussion of the cost difference that makes the solution change, and thus the selection, making the selection of computers with free software operating system preinstalled more interesting. Moreover, in this case we would observe that the component of inertia fails to be key or even to influence the selection process, and the real criteria for selecting the alternative is in this case the security issue first, and the problem of dependency, second.

6 Conclusions

The use of LP is a new application in the treatment of risks in projects. Its main advantage is that it is possible to represent real world scenarios with some degree of accuracy, as the number of constraints – and alternatives – can be measured in the hundreds. On the other hand, when analyzing the objective function for various scenarios it is possible to infer which is the best option [7].

Another major advantage is that, if there is a solution, this is optimal. i.e. the solution cannot be improved, thus confirming the Pareto optimal.

References

[1] M. Fernández-Diego, N. Munier. Bases para la Gestión de Riesgos en Proyectos 1st ed., Valencia, Spain: Universitat Politècnica de València, 2010.

[2] Project Management Institute. Practice Standard for Project Risk Management, Project Management Institute, 2009.

[3] M. Fernández-Diego, J. Marcelo-Cocho. Driving IS Projects. In D. Avison et al., eds. Advances in Information Systems Research, Education and Practice. Boston: Springer, pp. 113-124, 2008.

[4] R. M. Stallman. Free software, free society: Selected Essays of Richard M. Stallman. GNU Press, 2002.

[5] International Organization for Standardization. ISO 31000:2009 Risk management — Principles and guidelines, International Organization for Standardization, 2009.

[6] G.B. Dantzig. Maximization of a linear function of variables subject to linear inequalities, 1947. Published pp. 339-347 in T.C. Koopmans (ed.): Activity Analysis of Production and Allocation, New York-London 1951 (Wiley & Chapman-Hall).

[7] N. Munier. A strategy for using multicriteria analysis in decision-making. Springer – Dordrecht, Heidelberg, London, New York, 2011.

