

# On the use of Requirements for Driving Call-for-tender Processes for procuring Coarse-grained OTS Components<sup>§</sup>

Juan Pablo Carvallo<sup>1</sup>, Xavier Franch<sup>2</sup>

<sup>1</sup> Universidad del Azuay  
Av. 24 de Mayo 7-77 y Hernán Malo, Apartado 01.01.981 Cuenca, Ecuador  
jpcarvallo@uazuay.edu.ec

<sup>2</sup> Universitat Politècnica de Catalunya (UPC)  
c/ Jordi Girona 1-3 (Campus Nord, C6) E-08034 Barcelona, Catalunya, Spain  
franch@lsi.upc.edu

## Abstract

*Because of their complexity, software systems are currently built by integrating components of different nature into hybrid architectures, usually including some third party Off-The-Shelf (OTS) components. Several methods have been proposed to support OTS component selection. Most of them work by comparing requirements, incrementally gained through several iterations, and component descriptions. However, the iterative nature of these methods makes them practically unsuitable in some cases, particularly when conducting call-for-tender-based processes for selecting coarse-grained OTS components. In these cases, company goals need to be thoughtfully and systematically engineered to obtain a well structured set of requirements, which is to remain static during the entire procurement process and cannot be validated through hands-on experimentation. In this paper we report a case study in the selection of an ERP system for a telecommunications company using a call-for-tender process. We present the activities that were undertaken to elicit, analyze and structure the requirements to be included in call-for-tender documents, and the evaluation of components based on the identification of mismatches between requirements and component capabilities.*

## 1. Introduction

Most of today's software systems are built by integrating software components of different nature, into *Hybrid Systems Architecture* [1]. Software components used in this approach include third-party Off-The-Shelf (OTS) components [2] (e.g., commercial components –COTS–, free and open source components –FOSS– and Web services), as well as legacy and bespoke systems.

In this kind of approach, component selection plays a prominent role. As a result, several methods have been proposed to conduct this process (see [3] for a survey). Most of them require the comparison of system requirements, incrementally gained through several

iterations, to component characteristics, in order to identify mismatches. Mismatches are then used as basis for the analysis, negotiation and eventually the decision about the most appropriated components to integrate the proposed architecture. However, iterative selection methods do not behave well in several contexts:

- Call-for-tender-based processes. Invitation-based processes, mainly conducted by public organizations that require a predefined list of requirements to be established. In order to grant transparency, legislation in many countries requires the complete set of documents that conduct the process, to be made publically available prior the process even starts. Even if allowed, the modification of requirements is a cumbersome task, which requires time and paper work.
- Coarse-grained OTS components procurement. This kind of components, e.g. ERP systems, requires long periods of time for their parameterization (from 6 months to 2 years) before their deployment in a particular environment. Because of that, it is practically infeasible to perform hands-on experimentation on the products before they are selected and operational. In other words, it is not possible to validate quality requirements mismatches, before products are in operation and it is too late to go back in the decisions.

In this paper we analyze call-for-tender-based selection processes for coarse-grained OTS components from a requirements engineering perspective. Several lessons learned in the selection of an ERP system for a telecommunications company are presented, and related to the project concerns that they address. The paper is structured as follows: Section 2 gives some background to our work; Section 3 introduces the case study; Section 4 presents the lesson learned and Section 5 gives some conclusions and lines of future work.

This work is built upon our previous work in quality models presented in several forums (see [4, 5, 6] for a summary). The most distinctive characteristic of this current work is its application to the call-for-tender case which demands several particularities that we hope to make clear throughout the paper.

<sup>§</sup>This work has been partially supported by the Spanish project ref. TIN2007-64753.

## 2. The Experience

ETAPATELECOM is a new entrant telecom company, based in Cuenca, Ecuador. Established in 2002, it currently provides nationwide internet access, data carrying and public and domiciliary fixed telephone services. Its deployment strategy required the construction of a hybrid system to support its operation. System architecture (see Fig. 1) included strategic components, to be developed in-house, and generic ones to be acquired. In order to infer the architecture, *i\** models and strategic analysis techniques were jointly used. Details on the architecture are not a goal of this paper, see [7] for details.

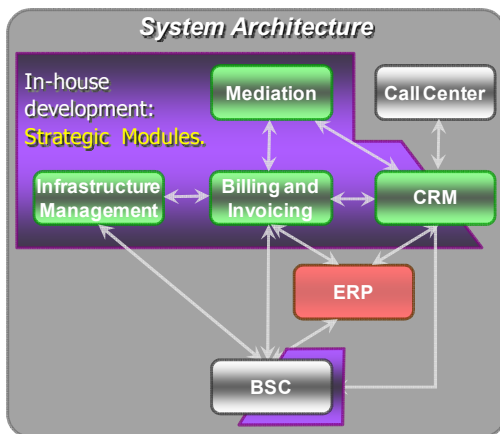


Figure 1. ETAPATELECOM hybrid system architecture.

The process for the selection of the core component of the architecture, an ERP system, was first authorized by the company board in December of 2005, and finally launched in January 2007. Although private in nature, the company was originally capitalized with public funds. Because of that, the selection process had to be conducted in accordance to local legislation for public companies.

A call-for-tender process was then established. Requirements documents were prepared in a six-month period by a multidisciplinary committee structured with one representative of each of the financial, administrative, legal and informatics departments of the company, under guidance of one of the authors of this work. The final call-for-tender documents included over 2000 requirements of different nature, structured into three hierarchies: functional, non-functional and non-technical.

Eighth candidates acquired the process documents, but only five presented offers. The process was conducted in a two-month period and was appointed to one of the two candidates that made it to the final phase of the process.

## 3. Main Project Concerns

During the process we had to deal with several concerns emerging from company directors, CEOs and technical staff involved in the project. Some were already

present before the project started whilst other become apparent during project execution. We mention:

- 1. Large set of multidisciplinary requirements.** Because of its coarse-grained nature and strategic impact, a great deal of technical (functional and non-functional) and non-technical requirements, emerging from stakeholders with very dissimilar interests and concerns (e.g., technical, financial, administrative, legal, etc.), needed to be elicited, consolidated and represented in a uniform way.
- 2. Documentation of component characteristics.** Components characteristic needed to be identified and validated in a reasonable time. Unfeasibility to perform hands-on experimentation, both for the coarse-grained structure of the component and the call-for-tender characteristic of the process, forced to rely on other sources, especially documentation. The extent, trustworthiness and quality of technical manuals were seen as additional obstacles for this task.
- 3. Identification of mismatches.** Mismatches among system requirements and component characteristics needed to be identified. The representational gap among the assets typically used in this kind of process, e.g., use cases and technical manuals, made this a really challenging task.
- 4. Local legislation.** Rules guiding the call-for-tender process (technical, economical, legal, etc.) needed to be established before the process started. Local legislation is not prescriptive, and the company had no previous experience in this kind of process, so the rules had to be defined from the scratch. All the activities had to be legally documented and there was no room for direct negotiation with suppliers. People conducting the process were administratively and legally responsible for the decisions made in the process.
- 5. Time and resource constraints.** Time span and resources assigned for the process were highly constrained. The component had to be selected and made operational in parallel with deployment of nationwide communications platform (2-year time). Preexisting system was collapsing and its suppliers ran out of business before completing technology transfer.
- 6. Costs and investment risks.** The company board was concerned about adoption and hidden costs, and recurring fees. Budget for the project was about 10% the cost of the nationwide telecommunications platform. Investment risks needed to be minimized and contractual clauses had to be engineered to grant project continuity to deployment.
- 7. Management of supplier contract.** The relationship and responsibilities of the supplier during and after project implementation had to be contractually agreed. Structured assets needed to be designed to help non-legal managers keep on track with the agreements.

## 4. Lessons Learned

This section addresses the lessons learned when preparing the call-for-tender documents and during component selection. Table 1 presents the relationships among the lessons and the concerns that they address. Lessons are grouped according the typical requirements engineering activities.

### L1: Use structured representations for requirements

**Discussion:** Requirements are the basis to drive component selection. In the selection of coarse-grained OTS components, several hundreds or even thousands of requirements, e.g., represented as use cases, have to be thoroughly compared against several volumes of technical and user manuals. There is a huge representation gap among these two types of assets, making this activity tedious and cumbersome, consuming far more time than the usually available and leads to uncertain results.

**Our approach:** We have used Quality Models (QM) as the conceptual vehicle to structure requirements. QMs provide a hierarchical decomposition of Quality Features (QFs) (characteristic, subcharacteristics and attributes), and metrics to compute their values. Once available, requirements over the domain can be stated as constraints of the quality features using the predefined metrics (see table 5 for some examples). QMs were constructed following the guidelines provided in [5, 6]. Hierarchies of technical (functional and non-functional) and non-technical QFs were included in the resulting model (see Table 2 for a detail of its general structure).

At first sight the amount of QFs included in the models may seem high, but is important to remark that coarse-grained OTS components are to support companywide organizational strategies and therefore cover large functional areas. On the other hand, the nature of call-for-tender processes requires certain degree of precision when writing requirements.

Table 1. Lessons - concerns relation.

Lessons		Concerns						
No.	Description	1	2	3	4	5	6	7
<b>Requirements Elicitation</b>								
2	Be aware of requirements evaluation objectives	X			X	X	X	
3	Focus on requirements for generic modules	X				X	X	
4	Don't let too much requirements detail run your project out of time	X		X		X	X	
<b>Requirements Analysis and Negotiation</b>								
5	Help stakeholders prioritize their requirements	X			X	X	X	
<b>Requirements Documentation</b>								
1	Use structured representations for requirements	X	X	X		X		
8	Make requirements part of your contract						X	X
<b>Requirements Validation</b>								
6	Make explicit the requirements evaluation rules				X	X	X	
7	Use requirements to drive the evaluation process		X		X	X		

Table 2. General structure of QM used in the case study

Type of Quality Feature	Subchar.				Derived Attributes	Basic Attributes				Total Quality Features
	Level 1	Level 2	Level 3	Total		Required	Obligatory	Desirable	Total	
Functional	12	49	216	277	82	990	-	89	1079	1438
Non-Functional	6	19	0	25	87	183	6	55	244	356
Non-Technical	3	19	0	22	27	161	-	7	168	217

### L2: Be aware of requirements evaluation objectives

**Discussion:** Requirements can be used to evaluate quality objectives directly related to their nature, e.g., functional requirements evaluate component's functional coverage, whilst supplier's assets and balances, help to determine its economical health. However, requirements can also be used to evaluate transversal objectives, e.g., costs and investment risks.

In practice, every call-for-tender process has its own evaluation objectives, in relation to the organizations environment, type, size, and business model, among others. As the number of transversal objectives to be evaluated grows, so does the associated set of requirements and viceversa.

**Our approach:** Two types of evaluation objectives were established. First, direct objectives in relation to technical and non-technical requirements (e.g., Evaluate Functional Scope of the ERP, and Evaluate Economical Health of Supplier). Second, two transversal objectives to address the main project concerns, namely, Risk Analysis and Cost Analysis. Evaluation objectives were used to guide requirements discovery, reducing the number of requirements and elicitation time. QFs in QM were individually checked to see if they were of interest for the process, labeled with the appropriated evaluation objective (see Table 3 for an excerpt) and further decomposed into more atomic QFs when required. Branches of QM with irrelevant QFs were discharged from the start.

### L3: Focus on requirements for generic modules

**Discussion:** Coarse-grained OTS components are built upon the knowledge gained in many implementation processes. Packaged experience and included best practices make them feature-rich and adaptable to several environments. As they evolve, numerous general (horizontal) and specialized (vertical) modules are incorporated into the products. Organizations may adopt them, and quickly gain access to the included knowledge. Parameterization and customization are the most costly activities when it comes to implementation. As the number of selected modules increases, consultancy time and cost are also increased. Customization of highly specialized or strategic modules can considerably increase adoption costs.

**Table 3.** QM excerpt of non-technical subchar. at level 2.

Subcharacteristics		Evaluation Objectives
<b>1 Supplier</b>		
1	Company Structure	Risk Analysis
2	Positioning and strength	Risk Analysis
3	Prestige	Risk Analysis
4	Services offered	Direct
5	Support	Direct
6	Warranties offered	Direct
...		
<b>2 Economic</b>		
1	Licensing Schema	Cost Analysis
2	Cost of deployment platform	Cost Analysis
3	Implementation costs	Cost Analysis
4	Network operation costs	Cost Analysis
5	Recurring fees	Cost Analysis
...		
<b>3 Product</b>		
1	Stability	Risk Analysis
2	Property	Risk Analysis
3	Deliverables	Direct
4	Parameterization /Customization	Risk Analysis

One of the objectives of call-for-tender processes is to help minimizing costs and maximizing return on investment. Thus, it is important to watchfully select the modules that are most appropriated for the organizational context and environment; focus the process on the selection on generics modules; and recommend strategic and specific modules to be developed in-house when possible.

**Our approach:** Requirements elicitation focused in 12 modules considered highly generic (e.g., Accounting, Inventory, and Fixed Assets among others). This decision reduced the number of requirements and thus the time and resources required for their discovery. Decision to develop strategic modules (e.g., Billing and Mediation) in-house was taken. They are required to evolve continuously, both to integrate new platform components and to incorporate new marketing and sales strategies. Total control over them provided a major advantage over competitors. Other legislation-dependant modules, e.g. Human Resources and Payroll were developed in-house too.

**L4: Don't let over-specification run your project out of time**

**Discussion:** In practice it is almost impossible to discover all the requirements for a coarse-grained component. No matter the amount of time invested, the list of requirements will always keep on growing. This is particularly true for technical requirements (both functional and non-functional ones). Furthermore, this growing may not add significant value to the selection process itself, unnecessarily increasing costs and time. There are several reasons for this:

- Coarse-grained components are usually very flexible. Workflow or activity sequencing, screen designs, additional data fields or field validations, among others, are easy to add or modify.
- They are usually provided in several versions to support a wide range of platforms (Hw., OS, DB, etc.).

- Too much detail may lead to an over-customized solution which increases implementations costs, forbids the adoption of best practices (preventing the acquisition of know-how typical of this kind of components) and compromises system evolution.

**Our approach:** Instead of going into too much detail (adding additional levels of decomposition into quality requirements in QMs), we focused on non-technical issues required to grant the continuity of the project and protect the investment. Supplier experience and economical health, warranties, services provided, and hidden costs are some of these issues. For more details regarding them, we refer to [8, 9].

**L5: Help stakeholders prioritize their requirements**

**Discussion:** It was mentioned in previous lessons that coarse-grained components encompass important amounts of features, addressing multiple kinds of stakeholders' needs. These needs are difficult to satisfy simultaneously. Technological and economical constraints, components limitations, and unrealistic expectations are some of the reasons behind this problem. In most cases, some requirements must be relaxed or eventually dropped. Otherwise, over specification may lead to dismissal of candidate components, compromising the success of call-for-tender process.

As the number of stakeholders and multidisciplinary requirements grows, it gets more difficult to objectively decide about the priorities and the most relevant requirements. Several systematic, matrix-based methods (see [10] for details), have been proposed to reduce subjectivity and help in the decision of which requirements have a greater weight or priority over other.

**Our approach:** From the legal point of view, clear definition of requirement priorities is a basic commitment in call-for-tender process. To fulfill this commitment Leopold matrixes were used to support requirements prioritization and the assignment of relative weights. Stakeholders were already familiar with them because they had been used in previous studies conducted by the organization. Technical staff led the process, trained the involved personnel, and prepared artifacts to help non-technical personnel input and compute their assessments. Although somehow limited, spreadsheets worked great for this purpose due to their tabular nature, and sorting and computation facilities. Users compute their opinions freely, without interference of technical staff. Validation techniques were adopted and results were communicated and negotiated with stakeholders.

**L6: Make explicit the requirements evaluation rules**

**Discussion:** One of the most relevant aspects in call-for-tender processes is to grant transparency. Call-for-tender documents shall make explicit the evaluation rules that

will guide the process. Suppliers need to be aware of them, even before the process starts. Their decision of whether or not to participate in the process may largely depend on the judgment of their capacity and ability to address the process as stated. Evaluation rules also constrain the capacity of selection teams: to evaluate and disqualify suppliers and their products; to require additional information; to address legal and economical concerns and to prevent from potential investment risks, among others.

In addition, coarse-grained OTS component selection processes are cumbersome. Evaluation of the important amount of requirements and dependencies among them, make these processes time and resource consuming. The rules guiding the process can help to significantly reduce time and costs of the process.

For instance, by adopting staged decision making schemas, in which components have to succeed in the evaluation of groups of requirements associated to evaluation objectives before going on with the remaining stages, costs can be reduced by reducing the number of components considered on each stage. However, this may lead to inappropriate results, if rules are not well analyzed. E.g., if it is decided to analyze only acquisition costs in a first stage, components technically more appropriated for the architecture may be discarded from the start. But if integration costs are also considered in that stage it may lead to a different result: they can be significantly lower for discarded components, balancing the original difference in costs.

**Our approach:** We reviewed local legislation and defined the following evaluation rules:

- *Total punctuation to be assigned.* We assigned a total punctuation to each requirements branch included in QMs. The punctuation was high enough to make easy its fractioning along different levels. It was proportionally distributed to QFs considering their relative weights (see Lesson 5).

- *Punctuation assignment for requirements partially fulfilled.* The percentages used in the case study are shown in Table 4. In this case, “ready-to-use” products were granted higher values.

- *Punctuation assignment for multi-valuated metrics* (metrics whose values depend of several factors). Two cases were detected in the case study: (1) requirements that had to be further decomposed into simpler ones; (2) requirements for which evaluation functions had to be defined. See Table 5 for some examples.

- *Decision-making schema.* We decided to conduct the process in a staged way. Requirements evaluation objectives were prioritized and stages were defined to follow the resulting order (see Table 3 for partial details): (1) requirements associated to the objective “Risk Analysis”; (2) non-technical requirements with objective “Direct”; (3) non-functional requirements associated to

the objective “Specific Environment”; (4) functional requirements; (5) requirements associated with the objective “Costs Analysis”.

- *Approval criteria for evaluation stages.* That is the criteria to decide when a given component was to be promoted to the next evaluation stage. Examples are:

- Requirements with the evaluation objective *Risk Evaluation* where evaluated with a high, medium, and low scale depending on their criticality. The company had the right to disqualify components/suppliers with requirements marked with high risk index. Mitigation contractual clauses had to be defined for requirements marked with medium risk.
- Requirements with the evaluation objective *Direct* where required to fulfill a minimum punctuation. 75% of the total weight for functional ones (inferior punctuations would have increased in excess the time and implementation costs, and were considered no suitable for the environment) and 70% for non-functional and non-technical ones.
- Requirements with the evaluation objective *Costs* were evaluated with *Net Annual Value* (NAV) methodology.

**Table 4.** Punctuation for partially fulfilled requirements.

Metric Value	Description	Weight value
SOP	Ready to use	100%
MOD	Modification required (reports, personalized GUI, etc.)	80%
CUS	Customization required (Source code modification)	40%
THR	Third party component required	20%
NS	Not supported	0%

**Table 5.** Punctuation assignment for multi-valuated metrics.

Case	Attribute	Metric	Requirement	Priority	Weight	Points
1	Supported export formats				50,00%	15,0
	XLS	Supported: Nominal; Supported=(Yes, Not)	Yes	Desirable	20,00%	3,0
	PDF	Supported: Nominal; Supported=(Yes, Not)	Yes	Desirable	20,00%	3,0
	XML	Supported: Nominal; Supported=(Yes, Not)	Yes	Required	20,00%	3,0
	CSV	Supported: Nominal; Supported=(Yes, Not)	Yes	Desirable	20,00%	3,0
	RTF	Supported: Nominal; Supported=(Yes, Not)	Yes	Optional	12,00%	1,8
	User defined formats	Supported: Nominal; Supported=(Yes, Not)	Yes	Optional	8,00%	1,2
2	Integration with Microsoft tools	List:(<Component: Nominal, Supported: Nominal, Scope: Nominal>); Component=Label ( Word, Excel, Project, etc. ), Supported=Label(Yes, Not), Scope=Text(Description of scope)	<Excel, Yes, All reports output>, <Word, Yes, All reports output>, <Outlook, Yes, workflow processes notifications, alarms, etc.>, <Project, Yes, Projects budget tracking>	Required	25,00%	1,5

### L7: Use requirements to drive the evaluation process

**Discussion:** Traditionally, the identification of component characteristics and their documentation requires the review of several sources of information (e.g., web pages, user and technical manuals, specialized magazines, product demos, etc.) and also some hands-on experimentation. Performing this very meticulous and painstaking task demands important amounts of time and

resources to be invested, as well as some technical skills. As stated in Section 3 the nature of call-for-tender processes which constraints evaluation time and available resources, and the nature of coarse-grained components, which make unfeasible to perform hands-on experimentation, make this task even more difficult.

**Our approach:** Rules of the process required suppliers to provide complete information regarding their products. QM was used as artifact to support this task. Columns were added in the spreadsheets in order to allow suppliers describing their products. Descriptions were stated in the QM in the same way than requirements, that is, as constraints over QFs using the predefined metrics. In that way descriptions of OTS components were stated in a uniform way with respect to requirements describing the “ideal” component. Additional columns were included for suppliers to provide references to technical and user manuals, for validation purposes

This way of working, as a kind of substitutive of hands-on evaluation, yielded crucial advantages: it reduced significantly the time and resources required for evaluation, helped to identify committed suppliers and to obtained first-hand information from them. In addition, provided answers were used as pre-contractual agreement.

### L8: Make requirements part of your contract

**Discussion:** Contracts for coarse-grained OTS components are highly specialized assets. They include clauses in relation to warranties, services to be provided (prior, during and after implementation), property rights and licensing schemas, costs, payment methods and services included in recurring fees, among others. In this kind of medium to long run projects, the nature and origin of the contract can compromise project management. On the one hand, suppliers prefer adhesion contracts provided by manufactures, which include standardized predefined clauses protecting them against future claims and responsibilities for product defects or damages caused in their operation. On the other hand, customized contracts are not concise and can get to be so complex that become unmanageable by technical project managers.

**Our approach:** Rules for call-for-tender projects are not prescriptive and some issues can be established to support contractual problems. In this sense, we required call-for-tender documents to be included as contractual documents. The QM and the requirements included on them, together with supplier’s answers, form a very structured asset that helps in the management of legal issues along the entire project span. In addition to the detailed list of technical requirements included, non-technical ones make explicit a wide variety of compromises among the parties involved, including the issues listed in the discussion.

## 5. Conclusions

Conducting call-for-tender processes for selecting coarse-grained OTS components is a crucial task for many organizations and has many particularities that may compromise its success. In this paper we have identified concerns and lessons learned in a case study in which we acted as leaders of the process. The case study was finally successful, and although this does not necessarily mean a cause-effect relationship, there was an agreement at the different company levels involved in the process that the actions described here were fundamental both to produce a high-quality call-for-tender document, and to conduct an efficient call-for-tender process:

- The document was focused (goal-driven), concise (not over-specified; understandable and easy-to-read), comprehensive (including all relevant aspects), structured (using the structure of the QM) and first-order citizen (acting as part of the contractual agreement).
- The process supported the evaluation of components characteristics (by identifying systematically which quality features had to be assessed), the comparison to the requirements (both system requirements and component characteristics were expressed as restrictions on elements of the QM), relying on some well-known technical methods and artifacts (in this paper we have remarked the use of goals and thus goal orientation, quality models and multi criteria decision making techniques) and issued both internally and externally the message that things were done not arbitrarily but with a well-defined rationale, which is specially crucial in call-for-tender processes.

## 6. References

- [1] *Proceedings of the 7<sup>th</sup> International Conference on Composition-Based Software Systems (ICCBSS’08)*, IEEE, 2008.
- [2] J. Li *et al.* “A State-of-the-Practice Survey of Risk Management in Development with Off-the-Shelf Software Components”. *IEEE Transactions on Software Engineering*, 34(2), 2008.
- [3] A. Mohamed, G. Ruhe, A. Eberlein. “COTS Selection: Past, Present, and Future”. CBSE 2007.
- [4] X. Franch, J.P. Carvallo, “A Quality-Model-Based Approach for Describing and Evaluating Software Packages”. IEEE RE 2002.
- [5] X. Franch, J.P. Carvallo, “Using Quality Models in Software Package Selection”. *IEEE Software*, 20(1), 2003.
- [6] J.P. Carvallo, X. Franch, C. Quer. “Determining Criteria for Selecting Software Components: Lessons Learned”. *IEEE Software* 24(3), 2007.
- [7] J.P. Carvallo. “Supporting Organizational Induction and Goals Alignment for COTS Components Selection by Means of *i\**”. ICCBSS 2006.
- [8] J. P. Carvallo, X. Franch, C. Quer. “Managing Non-Technical Reqt. in COTS Components Selection”. IEEE RE 2006.
- [9] J. P. Carvallo, X. Franch, C. Quer. “Towards a Unified Catalogue of Non-technical Quality Attributes to Support COTS-Based Systems Lifecycle Activities”. IEEE ICCBSS 2007.
- [10] E. Kornysheva, C. Salinesi. “MCDM Techniques Selection Approaches: State of the Art”. *Intl. Journal of Information Technology and Intelligent Computing (IT&IC)*, May 2008.