CRE: A Systematic Method for COTS Components Selection

Carina Alves, Jaelson Castro
Universidade Federal de Pernambuco, Centro de Informática,
Av. Prof. Luiz Freire, s/n – Cidade Universitária
CEP 50732-970 Recife, Pernambuco
e-mail: {cfa, jbc}@cin.ufpe.br

Abstract

The market interest in developing reliable and stable products at shorter development time and reduced cost, has led to an increasing surge of interest in Component-Based Software Engineering. The success of these systems largely depends on the effective selection of components that meet users requirements. In this context, the products evaluation needs to be a simultaneous process with the requirements acquisition. This paper presents the CRE Method (COTS-Based Requirements Engineering), which is centered on requirements to assist the COTS selection process. The selection of medical packages is used as a case study.
Key words: *COTS-Based Development, Selection Process, and Requirements Engineering.*

1. Introduction

As the size and complexity of software systems grows, increases the interest in developing systems based on reusable components, known in literature as COTS (Commercial-Off-The-Shelf). According to Oberndorf [19], "COTS are products that are sold, leased or licensed to the general public; that is usually available without source code; that is supported and evolved by the vendor who returns the intellectual property rights". The potential benefits of this new technology are reduced costs and shorter development time [18]. The nature of COTS suggests that the model of component-based software development should be different from the conventional development model. As a result, a significant shift has been observed from the development-centric toward a procurement-centric approach [24]. In general, the COTS-based development (CBD) lifecycle consists of the following phases: identification, evaluation, selection, integration and update of components [19].

It looks very promising to use COTS components in order to improve productivity and quality of software systems development. Although, the use of COTS software introduces new problems and risks, including difficulty in selecting suitable components and insufficient requirements analysis. For example, in a COTS-intensive system, many products from different vendors have to be integrated and tailored to provide complete system functionality. In many cases, these COTS products will be developed at different times, by different supplies. Moreover, organizations will have very limited access to product's internal design and its pre-defined options for customizing its behavior. Therefore in assembling these COTS products into an integrated system, organizations are placed in a situation over which they have no control.

The selection process initially decomposes the requirements for the selection of COTS products into a hierarchical criteria set. The criteria usually include components' functionality, non-functional requirements (reliability, portability, integrability, etc.), architecture constraints and non-technical factors such as vendor guarantees and legal issues. Then, during the selection activity, the properties of each COTS candidate are identified and assessed according to this set of evaluation criteria. In general, the requirements engineering process for the COTS-based development is affected by problems that are very different from those of traditional systems development processes [2]. In traditional systems development, user requirements are defined in detail and then the system is built from the specification that matches those requirements. Although in a COTS-based development, requirements statements need to be much more flexible and less specific [23]. In particular, there is no guaranty that a COTS product can meet all stated requirements. On the other hand, sometimes COTS products include features that were not originally required. Dealing with these unrequired features might complicate the evaluation process. We argue that the analysis of non-functional requirements can improve the discrimination process between competing COTS products that already meet the core functional requirements.

This work presents the CRE (COTS based on Requirements Engineering) Method, which focuses on non-functional requirements modelling to assist the processes of evaluation and selection of COTS products. This paper is organized as follows. Section 2 describes some challenges in COTS selection activity. Section 3 provides a comparison of current methods for COTS selection. Section 4 presents an overview of the CRE Method and its phases. Section 5 presents a case study in order to illustrate how the CRE Method can be applied in a practical manner. Finally, Section 6 presents the conclusions of this work.

2. COTS Selection Process

In a COTS-based development process, early evaluation and selection of candidate COTS software products is one of the key aspects of the system development life cycle. Its success largely depends on the accurate understanding of the capabilities and limitations of the individual candidate products. The selection of suitable COTS products is often a non-trivial task and requires careful consideration of multiple criteria [17]. We have identified four main dimensions that should be considered during the selection phase [2] see Figure 1.

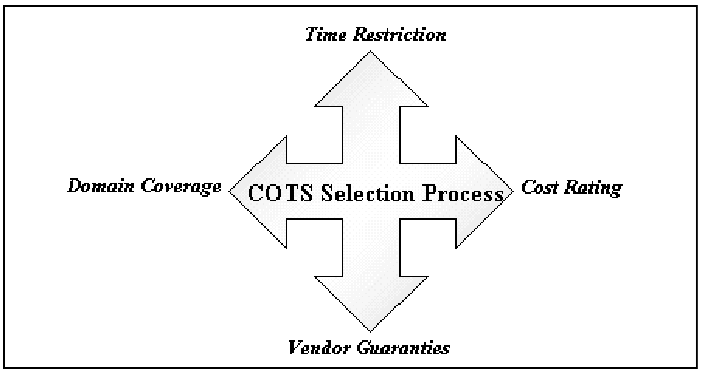

Figure 1 - Main dimensions of COTS selection

Domain coverage – The components have to provide all or part of the required capabilities, which are necessary to meet core essential customer requirements. Among these requirements, non-functional requirements play a critical role during the assessment process. In some cases, extra new components need to be developed to meet the shortfalls.

Time restriction – Software companies usually operate in a very rigid development schedule, on which their competitiveness depends. Selection is a time consuming activity, where a considerable amount of time is necessary to search and screen all the potential COTS candidates.

Costs rating – The available budget is a very important variable. The expenses when selecting COTS products will be influenced by factors such as: license acquisition, cost of support, adaptation expenses, and maintenance prices. Boehm [4] provides an economic model for estimating the cost of COTS-based system development.

Vendor guaranties – An important aspect to be considered in the selection activity is to verify the technical support provided by the vendor. Some issues have to be taken into account, for example: vendor reputation and maturity, number and kind of applications that already use the COTS, clauses characteristics of the maintenance licenses.

A mistaken assumption about COTS product evaluation is the idea that this activity is a one-off event for each selected product [6]. Rather, there are evaluation activities that precede the selection process as well as evaluation activities after the product have been selected, some of these are performed concurrently. This multi-stage evaluation process usually happens when there are several candidate products to be considered and where new versions of these products are emerging in the market sufficiently rapid to justify deferring some aspects of the evaluation until more information is known. Furthermore, successful selection and effective integration of COTS products is problematic for a number of reasons:

Lack of well defined process – Most organizations perform the selection process in an ad-hoc manner, this makes planning difficult, appropriate evaluation methods and tools are not used and lessons from previous experiences are not learnt;

Evaluation criteria – Sometimes evaluators include inappropriate attributes in the criteria and do not provide detailed specification discriminative attributes;

Black-box nature of COTS components – Lack of access to source code makes the understanding of components restrictive and therefore makes evaluation hard;

Continuous product updates – Rapid changes in the product marketplace and user needs makes COTS evaluation difficult [18], for example, a new release of the product may have a feature that is not available in the product that is currently being evaluated.

Given all these potential problems that makes the COTS selection a difficult activity during the COTS-based development, some methods have been proposed in the literature. Next section provides an overview of the main methods, describing their advantages and limitations.

3. Related Works

There are currently three strategies to COTS evaluation: keystone identification, progressive filtering and puzzle assembly [16]. In keystone selection strategy, products are evaluated against a key characteristic such as vendor or type of technology. Progressive filtering is a strategy whereby a COTS product is selected from a larger set of potential candidates, in which products that do not satisfy the evaluation criteria are progressively eliminated from the products list. In the puzzle assembly model, a valid COTS solution will require fitting the various components of the system together.

A range of COTS-based development methods has been proposed. For instance, the *OTSO (Off-The-Shelf Option) Method* [13] provides specific techniques for defining evaluation criteria, comparing the costs and benefits of alternative products, and consolidating the evaluation results for decision-making. Even though OTSO realizes that the key problem in COTS selection is the lack of attention to requirements, the method does not provide or suggest any effective solution. The method assumes that requirements already exist since it uses a requirements specification for interpreting requirements. Another work presented in [16] describes the *STACE (Social-Technical Approach to COTS Evaluation) Framework,* an approach that emphasizes social and organizational issues to COTS selection process. The main limitation of this approach is the lack of a well-defined process of requirements acquisition/modelling. Moreover, the STACE does not provide a systematic analysis of the evaluated products using a decision-making technique. The *PORE (Procurement-Oriented*

Requirements Engineering) Method is a template-based approach to support requirements acquisition. The method uses an iterative process of requirements acquisition and product evaluation. Although the PORE method includes some requirements acquisition techniques, it is not clear how requirements are used in the evaluation process and how products are eliminated. Table 1 provides a summary of these methods' features.

	Product Identification	***Requirements Acquisition***	***Non-functional requirements description***	***Product Evaluation***	***Decision making analysis***
OTSO	√	-	-	√	√
STACE	√	*	-	√	*
PORE	√	√	*	√	√

(√) addresses the issue fully (*) deals with the issue but not fully (-)does not deal with the issue

Table 1 - Summary of the processes covered by described methods

Moreover, current methods for COTS selection have not adequately treated non-functional requirements. It is worth noting that the role of non-functional requirements becomes more important because COTS components have their functionality already built-in [10]. As component's functionality is specified with the view of a generic customer, component's capabilities are likely to exceed the ones needed for a specific end-user. Functional requirements need to be specified, but only to a level which enables efficient assessment and evaluation of the available COTS components in the market.

On the other hand, the lack of a careful consideration of non-functional requirements increases the risks of COTS failure and the costs of the final system because these requirements often correspond to strategic or business objectives of the organization as a whole. Therefore, they are likely to have higher priority if conflicting with some of the functional requirements. In particular, the analysis of non-functional requirements such as performance and security can improve the discrimination between competing COTS products. For instance, if two components implement the same task (i.e. they have similar functionality), non-functional attributes may be used in the selection process as further and decisive criteria. Next section presents the CRE (COTS-Based Requirements Engineering) method.

4. The CRE Method

The CRE (COTS-based Requirements Engineering) Method [1][2] was developed to facilitate a systematic, repeatable and requirements-driven COTS software selection process. A key issue supported by this model is the definition and analysis of non-functional requirements during the phases of COTS evaluation and selection. The selection of COTS products is made by rejection. The products that do not meet customer requirements are rejected and removed from the candidate list. As the list decreases, the number and detail of customer requirements increases. The result is an iterative process whereby the requirements acquisition process enables products selection and this selection process also informs requirements acquisition, see Figure 2.

The CRE Method is goal oriented, i.e. each phase is oriented by predefined goals. Each phase has a template that includes some guidelines and techniques for requirements acquisition/modelling and products evaluation. These templates describe the main goals as well as the final results of each phase, see Figure 3.

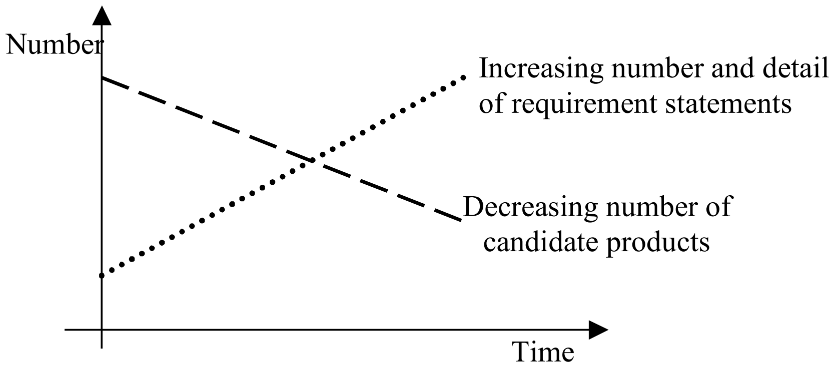

Figure 2 – Overview of the CRE iterative process

The method has four iterative phases: Identification, Description, Evaluation, and Acceptance. In particular, the sequence presented here is not rigid; each template can be used several times during the COTS selection process. CRE suggests that the requirements engineering process should drive the whole selection process, in order to facilitate the discrimination of candidate products. The overall phases and processes of CRE method are presented in Figure 4, which shows the iteration among the phases. Next sections describe each CRE phase individually.

Template1
Goals:
Final result:
Information and requirements to acquire:
Steps sequence:
Important considerations:

Figure 3 – Part of the CRE templates

4.1. Identification

The first task of the identification process is the goals definition. This must be based on careful analysis of the influencing factors. This work uses the classification proposed by Kontio [14], which identified five groups of factors that primarily influence the COTS software selection:

User requirements – include functional and non-functional requirements. First, the requirements specification is used to conduct products searching, and then it is used for evaluating COTS alternatives conformance degree with requirements.

Application architecture – provides a set of constraints deriving from how particular applications are built, this includes: components and design patterns used, communication and interface standards, platform characteristics. All of these introduce a set of constraints that may make integration of some alternatives impractical or costly.

Project objectives and restrictions – may influence the selection through the schedule or the budget of the project such as, early deadlines or low resources.

Products availability – it is important to check that the evaluation criteria is based on realistic expectations, i.e. the criteria set should not assume characteristics that are not provided by any COTS alternative. Otherwise, the system has to be fully or partly developed.

Organization infrastructure – includes the organization maturity with previous COTS selection processes, for example: skills of selection team, the availability of specific tools for supporting evaluation activities.

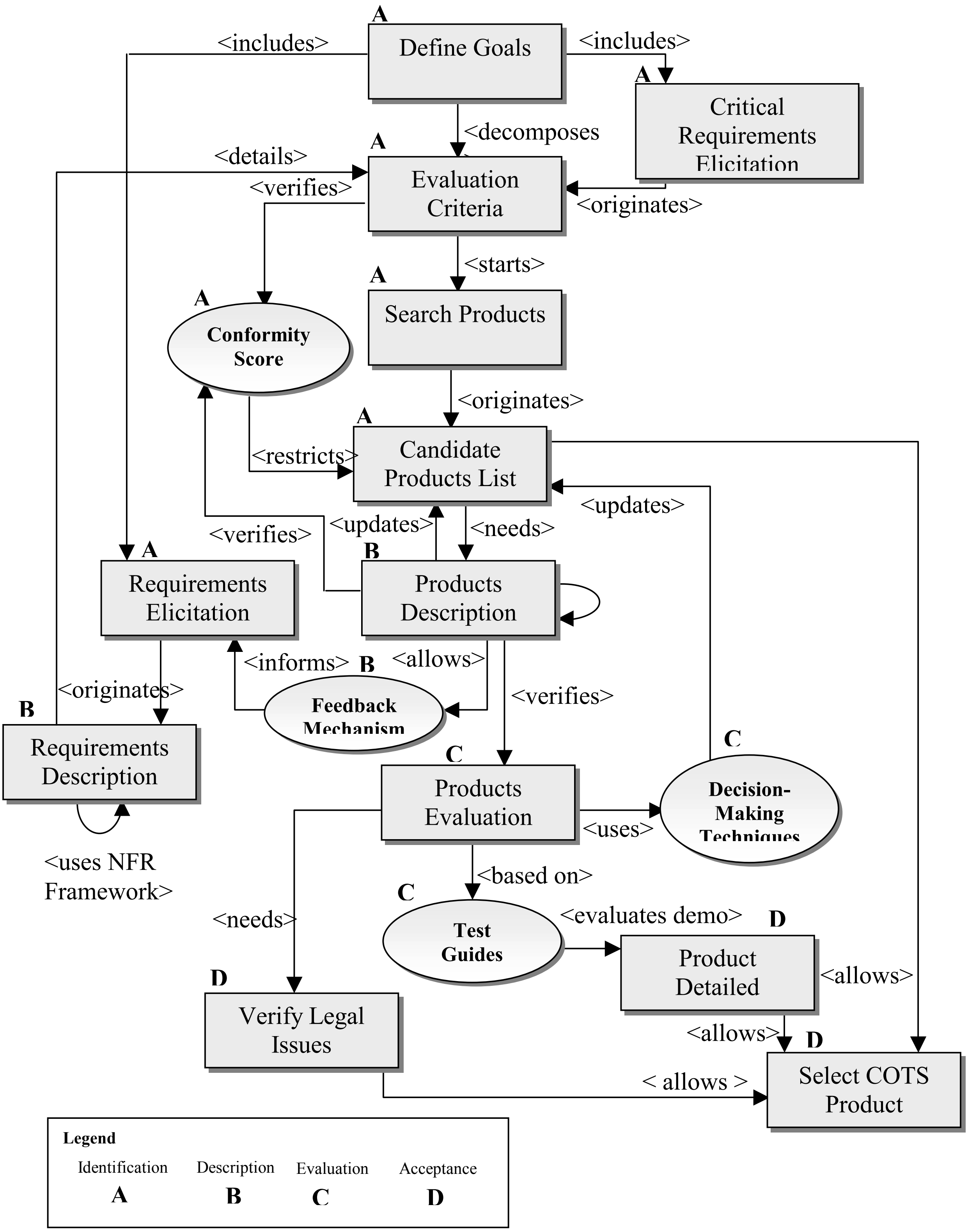

Figure 4 – Overview of the CRE Method

The evaluation criteria should be developed with awareness of all these factors. In most cases, this requires that each factor be explicitly analyzed and used as input for the evaluation criteria definition. The aspects related with resource and time constraints are usually critical and need to be carefully considered. Firstly, the evaluation criteria elaboration includes the elicitation of mandatory requirements, i.e. those related with central functionalities (such as the ability to manage and display patient records). In general, mandatory requirements are inflexible and rarely change. During this stage, the most appropriate elicitation techniques are interviews and questionnaires.

It is important to use several sources of information in the search process, typical sources are described in the following: in-house libraries, Internet, magazines, conferences, vendors, consultants and other organizations. The localization of products available in the market should be undertaken in parallel with requirements acquisition. The result of the search process is a list with all COTS candidates; this list contains general information about the products such as: name, supplier, main functionalities, hardware requirements, etc. Furthermore, we observe that at this moment requirements statements are still poorly described, specially non-functional requirements because they are usually very difficult to be quantified by customers. In this way, they need to be modelled and refined before initiating the technical evaluation. The following phase attempts to describe these requirements in adequate detail.

4.2. Description

During this phase, the evaluation criteria must be elaborated in detail, hence the important role of non-functional requirements in the discrimination between similar products. The CRE method suggests that the selection team should become familiar with requirements and products at the same time, thus making both requirements acquisition and market research more flexible and responsive. The next step consists of refining the requirements description, in special non-functional requirements (NFRs). These requirements are notorious for being difficult to elicit, express, quantify and test [9]. In general, suppliers do not provide a complete description of quality aspects (stability, flexibility, performance, etc). Furthermore, non-functional requirements can often be interacting, in that attempts to achieve one requirement can hurt or help the achievement of other. For all these reasons, it is usually difficult to evaluate if a product satisfies non-functional requirements. Yet, dealing with NFRs can be vital for the success of COTS-based software system.

The CRE method uses the NFR Framework [8] for representing and analysing non-functional requirements. This framework is a process-oriented approach where non-functional requirements are explicitly represented as goals to be achieved. Each goal will be decomposed into satisficing sub-goals represented by a graph structure inspired by the and/or trees used in problem solving. For instance, Figure 5 shows a decomposition of non-functional requirements using the NFR Framework. The goal *security of information* is decomposed into the subgoals *integrity, availability, confidentiality* through AND type of contribution (i.e. only if all subgoals are met the overall goal is achieved). While the goal *system performance* is decomposed into *throughput* and *response time*. Interestingly, it is necessary to address interactions between different kinds of non-functional requirements even though the non-functional requirements were initially stated as separate requirements. Note that *cryptography* contributes negatively (show as "-") for *system performance*. The NFR decomposition using this approach helps the definition of the evaluation criteria because it facilitates the understanding of what a particular non-functional requirement means.

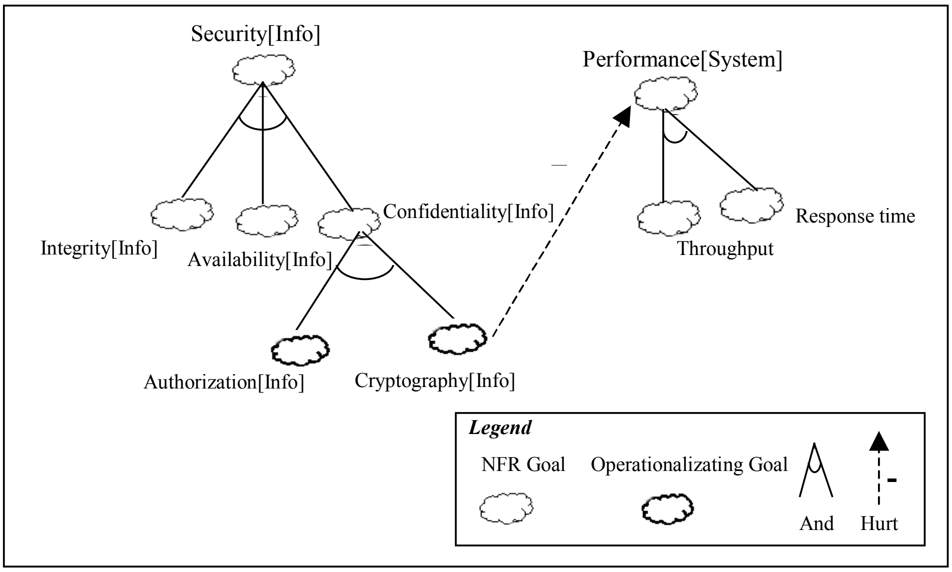

Figure 5 - Decomposition of non-functional requirements using the NFR Framework

An important process that occurs during this phase is the feedback mechanism, see Figure 6. It consists of an information exchange between the requirements process and the products description. It is quite possible that among the COTS alternatives, some extra functionalities (not initially considered) may be available. Some of these new requirements, upon a careful consideration, might indeed be required. This mechanism can be used to enhance the development process and user satisfaction.

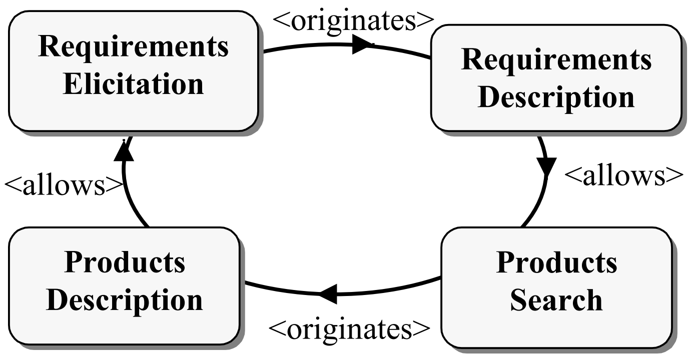

Figure 6 – Overview of the feedback mechanism

During this phase, the requirements document is elaborated and must contain all relevant information about the stakeholders requirements. In addition, the analysis of the requirements document is a fundamental part to refine the evaluation criteria. In the sequel each part of the document is explained:

Req-id – each requirement has a unique identifier;

Type – functional or non-functional requirements;

Description – detailed information about the requirements;

Priority – vary from 1 to 4;

4 – very high priority (mandatory)

3 – high priority (important)

2 – medium priority (would be nice)

1 – low priority (do not care)

Contributions – non-functional requirements can interact in synergy or conflict;

(++) make[1]

(+) help

(-) hurt

(- -) break

Comments – some observations about the requirement.

After identifying requirements, customers are asked to weight and prioritize them. In most cases, it is not possible that a particular product can meet all customer requirements, then an extensive prioritization and negotiation process is necessary. The CRE method provides some situation rules to treat these conflicts. The rules are represented in the form of a logical implication, ***IF <condition> THEN <action>***. Bellowing we describe a situation rule that deals with strong conflict between two requirements.

```
If Strong_Confl[Req1,Req2] and Req1_Prio > Req2_Prio
Then Attend Req1
Else If Strong_Confl[Req1,Req2] and Req2_Prio > Req1_Prio
Then Attend Req2
```

The CRE method provides a checklist that helps the acquisition of product information (see Table 2), this checklist informs what kind of information about candidate products and its suppliers is relevant to acquire. During this stage, the evaluation team relies on supplier data in sales brochures, technical documents, telephone conversations and information on the Internet. An important aspect of the checklist is to evaluate the products' conformance with quality standards. The ISO 12119, for example, establishes a set of quality attributes by which a COTS product might be described and evaluated. In general, evaluating and analyzing all the relevant characteristics of COTS candidates takes a great amount of time, typically more than the organization has [13]. Therefore, it is both necessary and cost-effective to select the most promising candidates for detailed evaluation.

Checklist	
Products Aspects	***Vendor Aspects***
√ Price	√ Maturity
√ Conformance with quality standards	√ Time delivery
√ Capacities	√ Stability
√ Benefits	√ Training
√ Constraints	√ Reputation
√ Version control	√ Support quality

Table 2 – Checklist for products description

4.3. Evaluation

This phase allows the use of appropriate techniques in evaluation data analysis for decision-making. In particular, the decision of selecting a particular COTS product is based on the estimated cost versus benefit analysis of each COTS alternative. The CRE method suggests the use of cost models such as the COCOTS (COnstructive COTS) [4]. This approach identifies four main classes of COTS software integration costs. These are costs due to the effort needed to perform (1) candidate COTS component assessment, (2) COTS component tailoring, (3) the development and testing of any integration or "glue" code needed to plug a COTS component into a larger system, and (4) increased system level programming

[1] This notation is adapted from the NFR Framework [8]

due to volatility in incorporated COTS components. The COCOTS model provides formulations, parameter definitions, and rating criteria for estimating the associated costs for COTS alternative integration.

In order to select or recommend a suitable required COTS product, the evaluated alternatives must be ranked according to their perceived relative importance to meet the customer's requirements. A simple and useful decision-making technique is the WSM (Weighted Scoring Method). The overall score of each alternative is calculated using the following formula:

$$\text{Score}_a = \sum_{j=1}^{n} (\text{ weight } * \text{ score}_{aj})$$ Where *a* represents an alternative and *n* the number of criteria

Formula 1 – The WSM Method formula

Conformance	*Score*	*Priority*	*Weight*
Do not meet the requirement	0	Low	1
Meets with restrictions	1	Medium	2
Meets partially	2	High	3
Meets	3	Very high	4

Table 3 – Values for score and weight using the WSM method

We suggest the values for score (from 0 to 3) and weight (from 1 to 4) as described in Table 3. The WSM method has some shortcomings that are often ignored. In fact, the resulting scores only represent the relative ranking of alternatives and the differences in their value does not give any indication of their relative superiority [13]. For more complex decision-making processes, a more effective technique is the AHP (Analytic Hierarchy Process). Thomas Saaty developed the AHP technique [22] for multiple criteria decision-making situations. This technique is based on the idea of decomposing a multiple criteria decision-making problem into a criteria hierarchy. At each level in the hierarchy, the relative importance of factors is assessed by pair-wise comparisons. We apply this technique in the following fashion: the product's characteristics, such as functional requirements, quality characteristics and non-technical aspects are considered as criteria during the decision making process, where each criterion is assigned a weight or a score. Finally, COTS products are compared in pairs with respect to the criteria. During the evaluation activity, the team have to decide which decision-making techniques is more suitable to each selection process. The result of the evaluation phase is the ranking of COTS candidates.

4.4. Acceptance

The acceptance phase is concerned with the negotiation of the legal contract with COTS suppliers. During this phase, the evaluation team have to resolve legal issues pertaining to the purchasing of the product and licensing. A license between the supplier and the customer should minimally specify [7]:

- The rights the supplier authorizes to exercise in the software (license grant);
- Payments to the supplier;
- Who owns the licensed product and future modifications;
- The risks and liability each party assumes under the license;
- Support, maintenance and warranties for the licensed COTS;
- The confidentiality of the licensed product.

The acceptance test must be derived from requirements to formulate questions to ask about the COTS product or scenarios for it to handle. In particular, the acceptance checking can take a long time if modifications to the delivered system are needed.

5. Case Study

In order to validate the method a real case study was developed. The objective of this study was to evaluate clinical packages available in the market for an oncology clinic. In the sequel we describe how each phase of the method was performed. We used the CRE templates [2] to guide this selection process.

5.1. Product Identification

The first activity was the identification of the selection goals. It included an analysis of the organization infrastructure and the application domain. The organization is an oncology clinic that realizes advanced research for cancer treatment. The clinic was looking for a new computational system to support new user requirements. Therefore, we conducted informal interviews with stakeholders to gather information and problems they encountered in the current system. During this initial process, we identified core requirements to be considered during the search of COTS products. The sources of market survey were: Internet searching, background reading (relevant medical journals) and contacting consultants. A total of over 20 products currently available on the market were identified. At this moment, we obtained general information about the products such as its name, supplier, hardware requirements and main functionalities.

5.2. Product Description

During this phase, we conducted brainstorming and interviews sessions with clinic staff. The goal of these sessions was to acquire and refine requirements statements. Table 4 provides an overview of the evaluation criteria (i.e. requirements statements) that candidate products should meet. Based on the initial evaluation criteria, we assessed the candidate products conformance with stated requirements. At the end of this analysis, only four products continued in the products list. According to CRE templates guidance, we used the evaluation criteria to develop a questionnaire to acquire products information that was sent to all candidate suppliers to request information about their products.

Evaluation Criteria		
Req_ID	***Type***	***Description***
Req_1	Functional	The system shall allow medical research from clinical cases
Req_2	Functional	The system shall allow to attach exams images into patient's records
Req_3	Functional	The system shall allow to customize patient's records
Req_4	Functional	The system shall provide billing management
Req_5	Functional	The system shall enable the doctor to create prescription templates
Req_6	Functional	The system shall display data from records in graphs
Req_7	Functional	The system shall create prescription from medical protocols
Req_8	Functional	The system shall display patients records on Web
Req_9	Non-Functional	The system shall be flexible
Req_10	Non-Functional	The system shall be user-friendly
Req_11	Non-Functional	The system shall be adaptable
Req_12	Non-Functional	The system shall have detailed documentation
Req_13	Non-Functional	The supplier shall provide efficient support
Req_14	Non-Functional	The supplier shall have good reputation

Table 4 – Evaluation criteria

The questionnaire asked suppliers to indicate the degree of compliance of their products with each requirement statement, it was divided into four sections: product's platform information, main functionality, supplier's technical support arrangements, and contract conditions. After the responses analysis, each supplier was contacted by phone to inform details about their products. In order to facilitate products comparison, we summarized the information obtained in the products list, part of this list is shown in Table 5.

General Information	*Arnaut Clinical Suit*	*PersonalMed*	*MultiMed*	*MV2000*
Supplier	Arnaut Informática	Gens Informática	Tecso Informática	MV Sistemas
Hardware Requirements	Server: Pentium II 96 MB RAM 2 GB Hard Disk CD-ROM drive Client: Pentium 100 24 MB RAM 280 MB Hard Disk	Server: Pentium II 96 MB RAM 1 GB Hard Disk CD-ROM drive Client: Pentium 100 24 MB RAM 200 MB Hard Disk	Server: Pentium II 128 MB RAM 4 GB Hard Disk CD-ROM drive Client: Pentium 166 32 MB RAM 400 MB Hard Disk	Server: Pentium II 256 MB RAM 4 GB Hard Disk CD-ROM drive Client: Pentium 166 64 MB RAM 400 MB Hard Disk
Development language	C++	Delphi	Delphi	PL/SQL
Operational System	Windows NT, Novell	Windows NT, Novell	Windows NT, Unix	Windows NT, Solaris, Unix, Novell
Data Base	FoxPro	FoxPro	Oracle	Oracle
...	...	…	…	…

Table 5 – List of candidates products

In parallel, we performed a refinement of non-functional requirements because it was difficult to analyze the conformance of each product with these requirements. Moreover, as indicated by CRE, suppliers did not provide a detailed description about product's quality aspects. In this context, we applied the NFR Framework [8] to refine and identify possible conflicts between non-functional requirements. Figure 7 shows the decomposition of requirements 9, 10 and 11. The non-functional requirement *User-friendly* was decomposed into *learnability for beginner and advanced user*, *accessibility of capabilities* and *simplicity of interface*, *documentation and installation*. The NFR *Flexibility* was refined into *integrability of data base*, *wrappability of product* and *extensibility of new modules*.

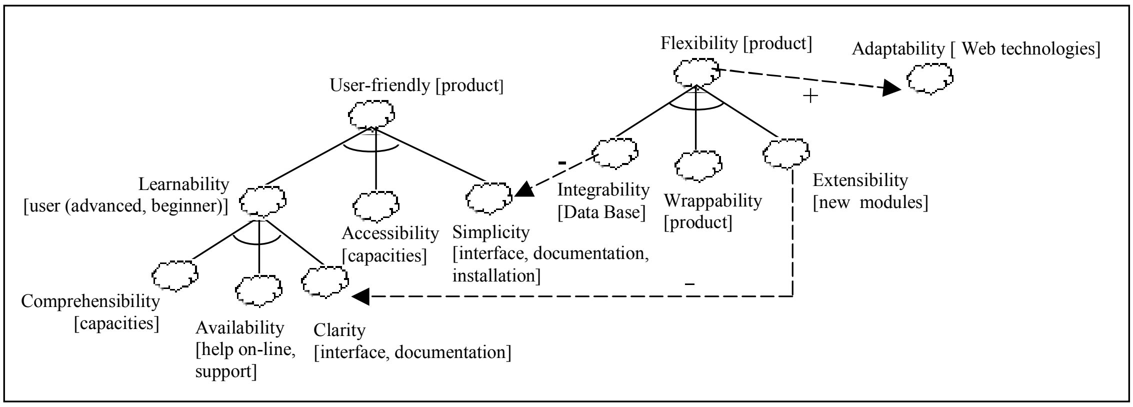

Figure 7 – Non-functional requirements refinement using the NFR Framework

After these refinements, we identified some implicit interdependencies among NFRs. For instance, the extensibility of new modules has a negative influence for the clarity of the system. In this case, the evaluation team weighted the tradeoff and decided that extensibility of new modules is still worth considering despite its negative contribution to system simplicity. Another conflict was found between integrability of data base and simplicity of instalation. In fact, the instalation and configuration of an Oracle data base is complex and need a DBA (Data Base Administrator). During the decision-making process, the evaluation team considered that data base integrability is more important than simplicity of instalation. The NFR Framework improved the understanding and representation of non-functional requirements stated in the evaluation criteria. In particular, it helped the evaluation team during the assessement of products' compliance with non-functional requirements performed in the next phase.

5.3. Product Evaluation

During this phase we requested trial copies for suppliers and conducted demonstration sessions with each candidate product. The objective of these sessions was to guide the evaluation team to acquire detailed product information sufficient to select or reject products. In addition, we were able to explore product suitability in a more realistic environment. We considered, among other things, product's compliance with non-functional requirements such as compatibility, integrability, usability, and performance. At this stage, we gathered information about how much tailoring or wrapping would be required to integrate each product as well as the amount of bespoke modules needed to be developed. We realized that all candidate products needed some additional development in order to satisfy specific user needs. In particular, requirement Req_1 (*The system shall allow medical research from clinical cases*) is very complex and specific because doctors develop advanced research. They perform statistical analysis from clinical cases in order to identify efficient cancer treatments. Therefore, Req_9 (*The system shall be flexible*) is a critical requirement that candidates products must meet.

Requirement	***Weight***	***Arnaut Clinical Suit***	***PersonalMed***	***MultiMed***	***MV2000***
Req_1	4	1	0	1	1
Req_2	3	3	1	2	3
Req_3	4	3	3	3	3
Req_4	4	3	3	3	3
Req_5	3	3	0	3	1
Req_6	2	2	0	2	3
Req_7	4	3	0	2	2
Req_8	2	0	3	0	3
Req_9	4	3	1	3	3
Req_10	3	0	3	2	3
Req_11	3	0	2	3	2
Req_12	1	3	2	3	3
Req_13	4	3	2	3	3
Req_14	2	3	2	1	2
Score		95	70	97	106

Table 6 – Products evaluation using the WSM Method

To enable an effective product evaluation, requirements were ranked using the WSM method. Stakeholders were asked to weight requirements according to their perceived importance (Table 3 shows values for score and weight). Then, we allocated compliance scores during each product demonstration session. Finally, we provided a final weighted score for each product. Table 6 presents these results, where the product MV2000 obtained the highest score and PersonalMed the lowest one and it was eliminated from the products list.

After analyzing the benefits of each candidate product, we performed a cost estimation. The CRE method emphasizes that the cost of developing a COTS-based system extends the acquisition price. In fact, it is also needed to consider other factors such as: training, adaptation, customization, infrastructure upgrade, installation, further development and maintenance costs. In this particular case, we found the main class of cost was additional modules development. We used some guidelines presented in COCOTS model [4] to estimate the effort relative to tailoring and developing additional modules in order to complement each COTS alternative. The cost estimation was made based on person-month effort. After analysing all these issues, MV2000 was considered the best-fit alternative because this product meets most stakeholders requirements as well as has the lowest customization and development costs.

5.4. Product Acceptance

During this phase, we negotiated contractual and legal issues with the MV2000 product supplier including licensing arrangements, maintenance and support. In addition, we established how the customization and further development should be performed, where these modifications will follow a pre-established planning and chronogram.

6. Conclusions

This paper has illustrated the importance of the selection process during the development of COTS-based systems. Main contributions of this research include the CRE method, which is an effective approach to guide the evaluation and selection of COTS products. In particular, it supports an iterative process of requirements engineering and product selection. Unlikely the current approaches that analyze only functional requirements to evaluate COTS products, our method also covers the non-functional characteristics of the candidates products.

It is worth noting that during the case study selection process both functional requirements and non-functional requirements needed to be considered as part of the evaluation criteria. Although, we found that non-functional requirements were more critical than functional ones. Therefore, we observe that our approach properly deals with a relevant aspect of the evaluation criteria, the modelling of non-functional requirements. As limitations of our approach, we consider that in cases with a large number of COTS alternatives and evaluation criteria the decision-making process can be very complex which a large number of possible situations can arise.

In terms of future work, it is necessary to detail the treatment of requirements prioritization and negotiation. Moreover, an important issue that can be better explored is the selection of multiple COTS products. Another area of concern is the interplay between software architecture and the selection of multiple COTS, some preliminary results are presented at [20].

References

[1] Alves, C. Castro, J. Um Método Baseado em Requisitos para Seleção de COTS. Fourth Workshop Iberoamerican on Software Engineering and Software Environment (IDEAS'01) San Jose, Costa Rica, April 2001.

[2] Alves, C. Seleção de produtos de Software Utilizando uma Abordagem Baseada em Engenharia de Requisitos. MSc. Thesis, Universidade Federal de Pernambuco, Centro de Informática. March 2001.

[3] Aoyama, M. New Age of Software Development: How Component-Based Software Engineering Changes the Way of Software Development. International Workshop on Component-Based Software Engineering, April, 1998.

[4] Boehm, B. Abts, C. Bailey, E. COCOTS Software Integration Cost Model: an Overview. Proceedings of the California Software Symposium. October 1998.

[5] Brown, A. W. e Wallnau, K. C. Engineering of Component-Based Systems, Component-Based Software Engineering. Software Engineering Institute, IEEE Computer Society Press, 1996.

[6] Carney, D. Assembling Large Systems from COTS Components: Opportunities, Cautions, and Complexities. SEI Monographs on the Use of Commercial Software in Government Systems, June 1997.

[7] Chaves, A. Tornabene, C. e Wiederhold, G. Software Component Licensing: A Primer. IEEE Software, October 1998.
[8] Chung, L. Nixon, B. Yu, E. e Mylopoulos, J. Non-Functional Requirements in Software Engineering. Kluwer Academic Publisher, 2000.
[9] Cysneiros, L. M. e Leite, J. C. Integrating Non-Functional Requirements into Data Modeling. 4th International Symposium on Requirements Engineering. Limerick, Irlanda. June 1999.
[10] Dukic, L. Non-Functional Requirements for COTS Software Components. Workshop Ensuring Successful COTS Development. May 2000.
[11] Foreman, J. On The Front Lines of COTS – Lessons learned, speculation for the future. Technical Report, Software Engineering Institute, 1999.
[12] Fox, G. Marcom, S. e Lantner, K. A Software Development Process for COTS-based Information System Infrastructure. Proceedings of the Fifth International Symposium on Assessment of Software Tools - SAST'97. June 1997.
[13] Kontio, J. A COTS Selection Method and Experiences of Its Use. Proceedings of the 20th Annual Software Engineering Workshop, Maryland, November 1995.
[14] Kontio, J. Caldiera, G. e Basili, V. Defining Factors, Goals and Criteria for Reusable Component Evaluation. CASCON'96. November 1996.
[15] Kotonya, G. e Sommerville, I. Requirements Engineering – Processes and Techniques. John Willy & Sons, 1997.
[16] Kunda, D. e Brooks, L. Applying Social-Technical Approach for COTS Selection. Proceedings of the 4th UKAIS Conference, University of York, April 1999.
[17] Ncube, C e Maiden, N. A. M PORE: Procurement-Oriented Requirements Engineering Method for the Component-Based Systems Engineering Development Paradigm. International Workshop on Component-Based Software Engineering, May 1999.
[18] Oberndorf, P. e Brownsworld, L. Are You Ready for COTS? Software Institute Engineering. August 1997.
[19] Oberndorf, P. Facilitating Component-Based Software Engineering: COTS and Open Systems. Proceedings of the Fifth International Symposium on Assessment of Software Tools - SAST'97. June 1997.
[20] Rosa, N. S. Alves, C. Cunha, P. Castro, J. Justo, G. Using Non-Functional Requirements to Select Components: A Formal Approach. Fourth Workshop Iberoamerican on Software Engineering and Software Environment (IDEAS'01), San Jose, Costa Rica, April 2001.
[21] Ryan, K. e Karlsson, J. Prioritizing Software Requirements in an Industrial Setting. International Conference on Software Engineering, 1997.
[22] Saaty, T. The Analytic Hierarchy Process. New York: McGraw-Hill, 1990.
[23] Thomas, B. Meeting the Challenges of Requirements Engineering. Spotlight, SEI Interactive, volume 2, issue 1. March 1999.
[24] Tran, V e Liu, D. A Procurement-centric Model for Engineering Component-based Software Systems. Proceedings of the Fifth International Symposium on Assessment of Software Tool. June 1997.