Challenges in COTS Decision-Making: A Goal-Driven Requirements Engineering Perspective

Carina Alves
Department of Computer Science
University College London
Gower Street, London
WC1E 6BT UK
+44 020 7679 7190
c.alves@cs.ucl.ac.uk

Anthony Finkelstein
Department of Computer Science
University College London
Gower Street, London
WC1E 6BT UK
+44 020 7679 7293
a.finkelstein@cs.ucl.ac.uk

ABSTRACT

This position paper outlines the problems and risks of selecting COTS products. In particular, we highlight the challenges of the decision-making process where requirements specification plays an essential role to evaluate and compare products features. It is necessary to perform a careful balancing between requirements and COTS features. Customers may have to compromise on requirements not satisfied by any available product or request products modifications. We analyse the problems and risks arising in the selection process and review related work. We argue that a goal-oriented approach can support an effective balancing between requirements and COTS feature during the decision-making.

Keywords

Requirements engineering, decision-making COTS-based system.

1. INTRODUCTION

The development of systems based on COTS (Commercial-Off-The-Shelf) products is considered a procurement-centric instead of development-centric approach. This approach is based on the acquisition and integration of off-the-shelf products over in-house development. COTS-based development brings fundamental changes in how organizations do their work [6]. Some familiar activities will be altered. Architecture design must be performed together with package evaluation; and new activities will become part of development process, for example product adaptation and integration.

The use of COTS involves some challenges and risks [3]. For instance, organisations have very limited access to product's internal design and the typical description of commercial packages is an incomplete and confused textual description. In fact, when evaluating these COTS, customers have limited chance to verify in advance whether the desired requirements are met. Attending demonstration sessions is after the way to understand available products. On the other hand, when buying COTS products, customers can take the advantage that a product that has been tested many times by users with consequent improvement in software quality.

In a COTS-based development process, early evaluation of candidate COTS software products is a key aspect of the system development lifecycle [10][17]. 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 a careful consideration of multiple criteria [14][1].

In practice, most selection decisions are based on subjective judgements, such as current partnerships, commercial profits, and successful vendor marketing. Moreover, organizations 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 potential COTS candidates.

It is widely accepted that COTS procurement must be an interleaved process with requirements specification [6], [7], [10], [11], [17]. Current methods for COTS selection fail to effectively support requirements specification for development of such systems. In particular, the evaluation process demands some form of inexact matching between products features and requirements specification, it is also necessary to engage in an extensive process of requirements negotiation in which the requirements of the organization are balanced against the capabilities of the package. Our work aims to develop a better understanding of how this balancing should be carried out in order to support the COTS decision-making.

This paper is organized as follows. Section 2 describes some challenges in requirements specification for COTS-based systems. Section 3 reviews some related works. Section 4 describes a

SEDECS '02, July, 2002, Ischia, Italy.

potential approach for COTS evaluation. Section 5 presents the conclusions of this work.

2. REQUIREMENTS SPECIFICATION FOR COTS

In traditional systems development, the requirements engineering (RE) activity basically consists of eliciting stakeholders needs, refining the acquired goals into non-conflicting requirements statements, and finally validating these requirements with stakeholders. The main goal of the requirements engineer is to ensure that the requirements specification meets stakeholders' desires and it represents a concise and clear description of the system to be developed. Broadly speaking, the specified requirements will be translated into software architecture and ultimately, implemented. Therefore, it is reasonable to assert that requirements play a controlling role in system development [21]. The RE process for COTS-based development is affected by problems that are very different from those of traditional systems. Below we discuss some challenges of developing systems from off-the-shelf products.

2.1 Requirements Flexibility

In COTS-based development, requirements statements need to be much more flexible and less specific [5]. For instance, suppose that performance is a critical requirement for a database system but none of the evaluated products satisfies the desired response time. This is a typical situation to deal with the buy versus build decision. If the final resolution is buying a product, customers must accept product limitations and requirements that cannot be met by any available COTS.

When developing systems with the goal of maximizing the use of COTS, the specified requirements should not be so strict that either exclude the use of COTS or that require large product modification in order to satisfy them [6]. In fact, an interesting approach is to let the available COTS features determine requirements. Consequently, it is necessary to achieve the best balancing of requirements precision and flexibility.

2.2 Dilution of Control

COTS products are developed based on a set of requirements that vendors believe will meet the widest number of potential customers [11]. Vendors try to meet the needs of a marketplace instead of satisfying the requirements of a particular organization. Therefore, COTS are designed to satisfy very general requirements. This in turn requires the customer to have an accurate understanding of products features to decide which parts must be adapted to conform to their particular needs.

An additional complication is that the vendor has full control over the product releases and upgrades. Therefore, customers are put into unexpected situations over which they have no control. Note that not only the source of control but also the scope of control has changed. For example, consider you had bought a product from a supplier that introduced a new packaging strategy, which included a new product that you do not want into the COTS you had purchased. You had no choice but to update the new product or to change for another supplier and perform a new assessment process.

2.3 Continuous Requirements Process

In traditional system development, requirements evolve as the environment in which these systems operate change. Typical changes to requirements specifications include adding or deleting requirements and fixing errors [18]. Evolution in requirements might lead to a temporary instability but as soon as the changes are managed and requirements agreed, the situation is controlled. However, in COTS-based systems, requirements are extremely volatile mainly because of rapid changes in the COTS marketplace. The vendor requires customers to accept new releases that bring new features that can be either unwanted or conflicting with stated requirements. Thus, this new situation leads to a continuous process of negotiation and trade-offs. We have to keep the decisions made during the assessment process in order to understand the reasons that forced requirements to change or why a particular product was eliminated. Capturing such rationale facilitates adaptation to ongoing changes [4].

Competitive pressures in the marketplace force vendors to innovate and differentiate products features rather than standardize them. This results in complex decision-making in which customers have to deal with incomplete and often mistaken understanding of product features. Standardization is a key issue to support the matching between COTS and requirements. However, we do not believe it will be a reality in the COTS marketplace at least for the next few years.

3. RELATED WORK

A number of COTS-based development methods have been proposed in the literature. Kontio [14] proposes the *OTSO (Off-The-Shelf Option)* method that provides specific techniques for defining evaluation criteria, comparing the costs and benefits of alternative products, and consolidating the evaluation results for decision-making. The definition of hierarchical evaluation criteria is the core task in this method, it identifies four different subprocesses: search criteria, definition of the baseline, detailed evaluation criteria definition, weighting of criteria. 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 specific solution. The method assumes that requirements already exist since it is based on the requirements specification for defining the evaluation criteria. *STACE (Social-Technical Approach to COTS Evaluation) Framework* [15], is an approach that emphasizes social and organizational issues related to COTS selection. The main limitation of this approach is the lack of a process of requirements gathering and specification. Moreover, the STACE does not provide an analysis of the evaluated products using a decision-making technique. *PORE (Procurement-Oriented Requirements Engineering) Method* [17] is a template-based approach to support off-the-shelf selection. The method is based in an iterative process of requirements acquisition and product evaluation. This method integrates some techniques, methods and tools, such as: knowledge engineering techniques, multi-criteria decision making methods, and requirements acquisition techniques. It also provides guidelines for designing product evaluation test cases. Although the PORE method includes some requirements acquisition techniques, the templates only give a preliminary view of the steps necessary to perform a systematic evaluation. It is not clear how requirements are specified in the evaluation process and how products are eliminated (i.e. do not capture the decision rationale). The main shortcomings of the proposed methods for COTS selection are:

In general, these methods rely on the definition of pre-established and structured criteria based on fixed requirements. These approaches are not appropriate to handle with the impositions of a highly volatile and uncertain COTS marketplace. They emphasize the importance of requirements analyses in order to conduct a successful selection that satisfies the customer. Although none of them support the complex process of requirements analysis and balancing with COTS features limitations. A common approach found in all methods described above is the use of multi-criteria decision making (MCDM) techniques to support the evaluation of COTS packages. The two most used approaches are AHP (Analytic Hierarchy Process) [19] and WSM (Weighted Scoring Method). The basic concepts of these approaches are establishing a list of criteria that products should meet, assigning scores to each criterion based on its relative importance in the decision and then ranking products based on their total scores. AHP provides a hierarchical approach for consolidating information about alternatives using pair-wise comparisons.

WSM technique has some limitations when applied in COTS assessment, for instance [14]: (i) this approach produces real numbers as results, so they can easily be interpreted as the true differences between the alternatives rather than the relative ranking (ii) difficulty in assigning weights when the number of criteria is large. AHP has some advantages over WSM, some previous experiments claim that AHP give decisions makers more confidence in their decisions [14]. The main limitations of these techniques are: (i) they assume independence between requirements (ii) they are weak in supporting multi-valued features and inexact matching of features with requirements [13]. We believe that MCDM techniques are adequate to handle quantitative criteria, in the sense that they measure the degree to which COTS satisfies customer requirements. On the other hand, they do not properly support qualitative reasoning. In this work we are particularly interested on addressing qualitative aspects of COTS decision-making.

4. USING GOASL TO SUPPORT THE DECISION-MAKING PROCESS

The general motivation of our research is getting a deeper insight into the COTS selection process. In this context, we consider some fundamental concepts for supporting a successful COTS-based development. Firstly, we need to keep the rationale of decisions made over the development life cycle not only during the evaluation process. Second, it is necessary to address the matching and consequent balancing between requirements and COTS.

Finally, the selection process must be systematic and well defined. In order to reach all these objectives we believe that a potential approach might be a goal-oriented one. The selection of a bug-tracking tool is used as an example to explain the presented concepts.

We use some ideas from goal-oriented requirements engineering. Thus, it is necessary to give a brief introduction of goals. According to Lamsweerde [16], "*A goal is an objective the system under consideration should achieve. They may be formulated at different levels of abstraction, ranging from high-level, strategic concerns to low-level, technical concerns.*" In fact, goals are particularly important in RE process:

- Goals provide the rationale for requirements i.e. requirements represent one particular way to achieve high-level goals. In general, there are many refinement alternatives to be considered during the requirements specification process.
- Goal refinement process provides a suitable abstraction level to support decision makers evaluating requirements alternatives.
- Requirements are known to constantly evolve during system life cycle, once requirements are refinements of goals, the latter are more stable. Thus, the higher level a goal is, the more stable it will be [2].

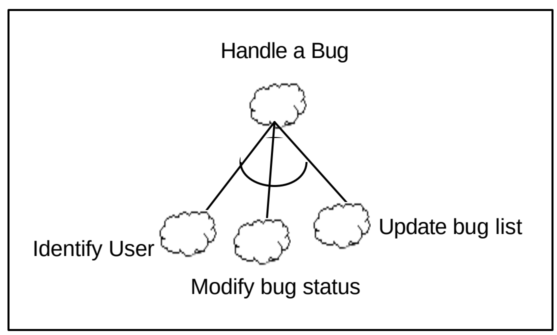

Figure 1 – Modelling Goals Refinement

Goals have been recognized as a leading concept in the RE process [2],[8],[9]. The modelling of goals has many benefits, such as: to represent goals explicit, to identify interdependencies among goals, to support qualitative reasoning. Each goal can be decomposed into satisficing sub-goals represented by a graph structure inspired by the AND/OR trees used in problem solving. AND links refine a goal into a set of subgoals; which means that only if all subgoals are met the overall goal is achieved. OR links refine a goal into an alternative set of refinements, which means that satisfying one of the subgoals is sufficient for satisfying the parent goal. For example, the bug tracking goal "Handle a bug" can be decomposed into the following subgoals: " Update bug list", "Modify bug status" and "Identify user" through AND link (see Figure 1). Using the main concepts of goal RE, we give an

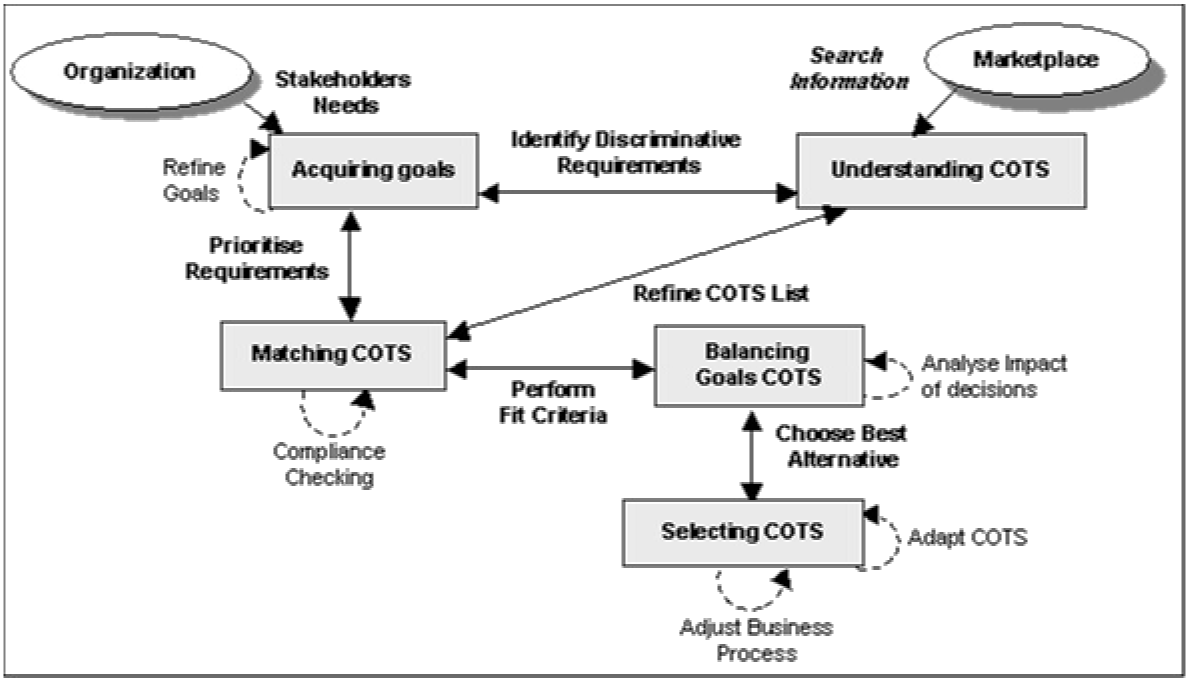

Figure 2 – Main Activities of COTS Selection Process.

overview of the main activities within COTS selection and outline the related challenges of each activity. Figure 2 shows the activities and their interactions. Each of these is described below.

4.1 Acquiring Goals

The processes of goals acquisition and specification must be incremental and iterative [8]. In the beginning, high-level goals are identified using typical elicitation techniques, such as use-cases. From these goals, possible COTS candidates are identified in the marketplace, in which new goals may be recognized in features that make this process highly iterative.

In RE literature, the notion of requirement is related to services that the system should provide [18]. As we discussed in Section 2, requirements for COTS systems should be more flexible and specified as desirable rather than mandatory features. Therefore, we believe it is more reasonable keeping the notion of goal until the package has been selected. Along the modelling process, goals are defined; interdependencies among goals found; goals are refined based on package features; different refinement alternatives are tried; priorities and trade- offs are addressed. Early decisions need to be considered and reconsidered when making later decisions.

During the refinement process it is necessary to identify goals that helps to distinguish between products (called *core* goals) from those that are provided by most available products (called *peripheral* goals). For example, the feature "web-based interface" is supported by all evaluated packages (example of peripheral goal) while "support of multiple projects" is supported by only a few packages (core goal). Thus, the latter feature can be a decisive criterion and should be investigated in order to support the decision of selecting one product instead of others. Besides the classification of goals as *core* and *peripheral*, we propose two attributes for goals description:

Desirability – the importance of a goal described in the customer specification to be satisfied by a particular feature provided by the package.

Modifiability – the capacity to restructure a goal definition when a conflict arises between a specified goal and any feature provided by the package.

In other words, *desirability* specifies the priority of goals; here one possibility is assigning numbers that represents the relative importance of goals as applied in some MCDM methods. In particular, the number of hard goals should be minimised because even if any product fits well with a complex set of goals, the volatility of marketplace practically imposes that the fit will be short-lived [21]. Instead of hard goals, we should have more fluid goals, some mandatory, some strongly desirable, some only nice to have [5]. The *modifiability* attribute refers to the flexibility in which a goal can change in order to accept features constraints. For example, consider that a stakeholder specified the goal "define a filter for searching bugs" she also wanted to search which bugs were resolved by each member of the project. All available products provide filters for searching bugs, but none has this specific filter. In this case, we need to verify how easily modifiable this goal is and if we want to relax this specific need in order to meet a constraint.

4.2 Understanding COTS

Our experience shows that there is a language mismatch between COTS features description and customers needs, where this mismatch increases the chances of selection failing. In fact, questions of vocabulary are a critical issue to be treated. Understanding a COTS demands a translation of several types of vocabulary: the vocabularies of various products and the vocabulary of the customer's goal specification [13]. Moreover, the package descriptions provided by the vendor are usually too general and more suitable for a commercial leaflet than for a precise evaluation of package features. During the evaluation it is necessary to gather information about functional features, quality aspects and other characteristics of the package; for example, description of the system environment, previous product versions, customer's support strategies.

4.3 Matching Goals and COTS

The evaluation of COTS demands some inexact matching. For example, there may be goals not satisfied by any available package, goals satisfied by some joint packages, goals partially satisfied, features of package not initially requested but that can be helpful, features irrelevant or even unwanted. Moreover, there are some cases where core goals cannot be entirely satisfied without considerable product adaptation and other cases where these goals must be compromised to match product features. An additional complication is that both goals and package specification might have incompleteness and inconsistencies. In short, it is necessary to perform a complex compliance checking to treat all these issues. Figure 3 shows a schema of a matching between goals specified on the fit criteria and two potential candidates. The fit criteria enable the evaluation of COTS candidates based on stakeholders goals (i.e. how each package meet these goals). In general, they include some aspects the package should provide: functionalities, non-functional aspects (performance, usability, interoperability), vendor issues (reputation, support, stability). It is worth noting that in both matching scenarios the fitting is not complete, there are some gaps between goals and features that must be balanced in order to reach the best solution.

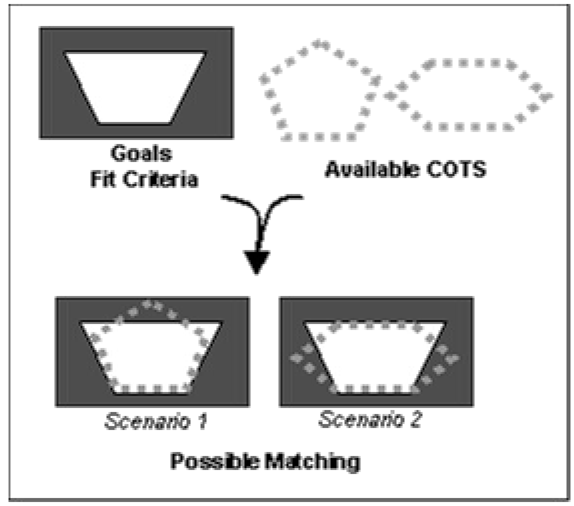

Figure 3 – The Matching Between COTS and Goals

4.4 Balancing Goals and COTS

The fit criteria enable the verification whether or not a COTS solution satisfies stakeholders' goals. Different degrees of satisfiability may be distinguished rather than a binary answer (for example the bug tracking tool support or not file attachments for bugs).

In fact, non-functional requirements are known as not been satisfied in a clear-cut sense [8], for example, how can we evaluate the usability of a bug tracking tool? What means a user-friendly system for us? Which aspects are important (interface, documentation, etc.)? The evaluation of such goals demands a sort of qualitative reasoning strategy and the refinement process helps understanding these subjective goals and how they can be fulfilled. This strategy provides more specific interpretations of what high-level goals mean.

The balancing between goals and COTS features is an important step of the decision making process in which goals must be negotiable until the selection process has finished. The negotiation must be bilateral, analysing the impact of decisions over goals that had to be traded-off and features that had to change. Customers have to analyse the expected benefits of performing large products adaptation in order to meet a specific set of goals. All decisions taken must be properly documented and supported by well-justified arguments.

4.5 Selecting COTS

The selection process includes the ranking of packages based on their compatibility with customers' goals. Besides the assessment of functionalities that a package meets, it is also necessary to analyse the risks associated with each alternative, for instance, the complexity to perform package adaptations, the support that a vendor provides for these adaptations, the number and importance of goals that had to be compromised. Some of these issues are unpredictable what increases the risks that the selected package is not the best solution.

In reality after a product has been selected, there is still a considerable amount of work to do. Customers may change their business practices in order to fit the product, where it is necessary to perform a careful impact analysis of the selected product over the organization. Products probably will need modifications, what can range from simple customisations to large adaptations including the development of wrappers or even in-house development of extra modules to cover critical features not supported by the product.

5. CONCLUSIONS AND FUTURE WORK

This position paper has illustrated the main challenges of selecting suitable COTS products. We outlined the main differences between traditional RE and COTS-based RE. A fundamental aspect of COTS selection is that it must take into account multiple interdependent criteria, in which we need new approaches to treat such issues. We described the potential benefits of a goal-oriented approach to support the COTS decision-making process and sketched the key activities that should be performed during the selection. In terms of future work, we aim to focus on the balancing process between goals and package features. In this perspective, there are a number of important issues to investigate, such as: dealing with the inconsistent and incomplete nature of COTS features, prioritising and negotiating goals, addressing possible trade-offs.

6. REFERENCES

[1] Alves, C. Castro, J. CRE: A Systematic Method for COTS Components Selection. XV Brazilian Symposium on Software Engineering (SBES) Rio de Janeiro, Brazil, October 2001.

[2] Anton, A. Cracken, W. Potts, C. Goal Decomposition and Scenario Analysis in Business Process Reengineering. Proceedings of CAISE'94, 1994.

[3] Barros, M. Werner, C. Travassos, G. Scenario Oriented Project Management Knowledge Reuse within a Risk Analysis Process. SEKE'01. Buenos Aires, Argentina. June 2001.

[4] Boehm, B. Requirements that Handle IKIWISI, COTS, and Rapid Change. IEEE Computer 33(7), July 2000.

[5] Carney, D. COTS Evaluation in the Real World. SEI Interactive, Carnegie Mellon University, December 1998.

[6] Carney, D. Requirements and COTS-Based Systems: A Thorny Question Indeed. SEI Interactive, Carnegie Mellon University, June 1999.

[7] Chung, L. Cooper, K. Towards a Model Based COTS-Aware Requirements Engineering Process. 1st International Workshop on Model-Based Requirements Engineering, San Diego, USA, November 2001.

[8] Chung, L. Nixon, B. Yu, E. and Mylopoulos, J. Non-Functional Requirements in Software Engineering. Kluwer Academic Publisher, 2000.

[9] Dardenne, A. Lamsweerde, A. Fickas, S. Goal-directed Requirements Acquisition. Science of Computer Programming. Vol 20 North Holland, pp 3-50, 1993.

[10] Dean, J. Vidger, M. COTS Software Evaluation Techniques. Proceedings of The NATO Information Systems Technology. Symposium on Commercial Off-the-shelf Products in Defence Applications, Brussels, Belgium. April 2000.

[11] Deifel, B. Requirements Engineering for Complex COTS. 4th International Workshop on Requirements Engineering: Foundation for Software Quality. Pisa, Italy 1998.

[12] Dukic, L. Non-Functional Requirements for COTS Software Components. Workshop Ensuring Successful COTS Development. May 2000.

[13] Finkelstein, A. Spanoudakis, G. Ryan, M. Software Package Requirements and Procurement. Proceedings of the 8th International Workshop on Software Specification and Design, 1996.

[14] Kontio, J. A COTS Selection Method and Experiences of Its Use. Proceedings of the 20th Annual Software Engineering Workshop, Maryland, November 1995.

[15] Kunda, D. e Brooks, L. Applying Social-Technical Approach for COTS Selection. Proceedings of the 4th UKAIS Conference, University of York, April 1999.

[16] Lamsweerde, A. Goal-Oriented Requirements Engineering: A Guided Tour. Invited mini-tutorial paper 5th IEEE International Symposium on Requirements Engineering. Toronto, Canada. August 2001.

[17] Ncube, C. 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] Nuseibeh, B. Easterbrook, S. Requirements Engineering: A Roadmap. The Future of Software Engineering. ACM Press, 2000.

[19] Saaty, T. The Analytic Hierarchy Process. New York: McGraw-Hill, 1990.

[20] Sivzattian, S. Nuseibeh, B. Linking the Selection of requirements to Market Value: A Portfolio-Based Approach. 7th International Workshop on Requirements Engineering: Foundation for Software Quality. Interlaken, Switzerland, June 2001.

[21] Wallnau, K. Hissam, S. Seacord, R. Building Systems from Commercial Components. SEI Series in Software Engineering. Addison Wesley, 2002.