11th International Research/Expert Conference
"Trends in the Development of Machinery and Associated Technology"
TMT 2007, Hammamet, Tunisia, 05-09 September, 2007.

A FRAMEWORK SUPPORTING "BUILD OR BUY" DECISIONS IN THE ENTERPRISE SOFTWARE ACQUISITION PROCESS

Ceyda Güngör Şen
Hayri Baraçlı

Yildiz Technical University, Faculty of Mechanical Engineering,
Department of Industrial Engineering, Besiktas, Istanbul, Turkey

ABSTRACT
The software acquisition problem is depicted as two dimensional: what acquisition approach to use (custom develop or base it on a package) and who should complete the task (internal resources or external providers). Although there are a lot of studies about aspects of the sourcing decision, the custom/package decision has previously received little attention in the literature. However, decision on whether to develop software applications or buy commercially available products is a common dilemma that frequently emerges. This "build or buy" decision has become even more complex in recent years with new developments in the enterprise software environment. To make the best choice, firms must weigh the top considerations of their business processes against the advantages of each type of system. In this paper, we introduce a framework to support "build or buy" decision in the enterprise software acquisition process. Following the stepwise procedure of our proposed framework, system requirements are analyzed based on a set of desired business functions and a list of software alternatives is produced. A heuristic algorithm is presented to obtain functional suitability scores of the different software products. This algorithm allows decision makers to compare the actual performance value of the software alternatives with the expected performance value of the resulting software system in terms of the system requirements. The "build or buy" decision is made by using the outputs of the algorithm. If no product meets an acceptable level of functional coverage, as determined by the decision makers, the required capabilities must rely on custom development rather than purchasing a packaged product.
Keywords: Enterprise software, software acquisition, build or buy decision, functional suitability, heuristic algorithm.

1. INTRODUCTION
In the early days of computing, if an organization wanted to automate a process, that organization had to arrange for technical staff to design and develop the required program. However, since the first software packages were developed, that scenario has changed. Today, in virtually every case, an organization contemplating the automation of a process for the first time or changing technology must face the "build or buy" question. In some cases, the answer is obvious. It would make no sense for an organization to build its own word processing, spreadsheet, or e-mail application. Commercial firms have invested millions of dollars to develop systems that are sold into many markets and generally meet the vast majority of every organization's requirements. In other cases, particularly when an organization is considering an enterprise-wide system, the answer is less obvious and many factors need to be considered before reaching a conclusion [1].
The enterprise software acquisition problem is depicted as two dimensional: what acquisition approach to use (custom develop or base it on a package) and who should complete the task (internal resources or external providers) [2]. Using packaged software involves identifying the company's requirements, evaluating alternative packages, selecting one, possibly modifying it, and installing and

testing the selected system. If an external provider carries out these functions, the acquisition is an outsourced package. If in-house staff is used, the acquisition is an insourced package. Customized software is developed when it is not cost-beneficial to meet the user's requirements through modifications to packaged software. Custom software acquisition generally involves a series of possibly repeated steps: needs analysis, system design, coding, testing, training, and installation. Outsourced custom projects use vendors to complete these activities; insourced custom development uses only internal resources. Each acquisition option has some positive and negative consequences that must be fully examined, if the decision is to bring optimum results to the organization [2].

Until about ten years ago, the build option was the only viable option, because packaged enterprise software products were not yet available in the market. Today, organizations are increasingly purchasing enterprise software packages [e.g. enterprise resource planning (ERP) systems] instead of custom developing their own software applications; due to the growth in specialized software companies, coupled with diverse skill requirements, and rapidly changing technology [3]. Key to this choice is often the perception of lesser risk in choosing such a product that others have successfully implemented. Although most organizations now rely on complex packaged software solutions, the build alternative is not irrelevant. Many organizations have continued to develop software in-house through necessity, driven by unique business practices or the need to address areas that remain unfulfilled by the alternatives available in the market.

While there is no single answer to the build versus buy dilemma, by clearly identifying needs, expectations and resources early in the process, it is possible to make a decision that minimizes risk and leads to the best possible result for the organization. In this paper, we introduce a framework to support "build or buy" decision in the enterprise software acquisition process. In the following section, we provide the details of the proposed framework.

2. A FRAMEWORK FOR MAKING THE "BUILD OR BUY" DECISIONS

The aim of the proposed framework is to evaluate the software alternatives by the system requirements and to obtain functional suitability score for each product. The proposed framework allows decision makers to analyze the system requirements based on a set of desired business functions, evaluate the commercially available enterprise software alternatives by these requirements and obtain functional suitability score for each software system. The "build or buy" decision is made by using the outputs of the proposed heuristic algorithm. To clearly present the framework, a stepwise procedure is described. The details of each step are presented below.

2.1. Form a project team and identify system requirements

The first step is to form a project team that consists of decision-makers, functional experts and senior representatives of user departments. An enterprise software project is not only installing a new information technology system but also reshaping the business processes to overcome the challenges of dynamic market [4]. Business process reengineering (BPR) is necessary to be undertaken to rationalize and standardize the workflows of all business processes in advance. The project team can develop the functional requirements of enterprise software during the BPR and then incorporate these characteristics appropriately into the decision model. Our framework relies on user-defined requirements instead of a short list of evaluator-generated criteria to determine product suitability. The cross-functional team can easily gather the basic functional requirements. These requirements may be represented in a list of process and business related functions, scenarios or use-cases, and should include essential user requirements and standards [5]. Such basic requirements are often both vague and non-verifiable. Hence, we propose to expand these primary requirements into secondary and tertiary, more detailed requirements.

2.2. Collect information concerning commercially available enterprise software alternatives

A wide range of information concerning enterprise software vendors and systems should be obtained from professional magazines, exhibitions, yearbooks, the Internet, and the other sources [6]. Instead of using a short-list of candidate products, we propose to identify every product that possibly addresses the requirements. To narrow the list into the serious candidates, evaluators compose a questionnaire involving all system requirements and submit it to vendors of potential products. Furthermore, evaluators in cooperation with users can create appropriate scenarios and investigate

product performance in these scenarios. After examining the inputs from vendors, the project team can eliminate the clearly unqualified vendors and thereby reduce the number of alternatives.

2.3. Obtain the functional suitability scores of the software alternatives using heuristic algorithm

The aim of functional suitability analysis is to evaluate the software alternatives by the system requirements and to obtain functional suitability score for each product. This evaluation is a qualitative process that produces data on how well each alternative meets the identified requirements.
In this step, team members assign weights ranging from 0 to 1 to each primary (w_i) and secondary system requirement (w_{ij}) through brainstorming sessions. Team members also assign numerical values (0, 1, 2 or 3) to each tertiary system requirement indicating the expected performance value of the resulting software system (e_{ijk}) by reaching consensus. Requirements receive a 3 for "full coverage", 2 for "partial coverage", 1 for "inadequate coverage" and 0 for "no coverage". During the vendor demonstrations, the software alternatives are evaluated according to the actual performance. Numerical values to each tertiary requirement indicating the breadth of coverage for the *s.* candidate product (p_{sijk}) are assigned by using the same scale. Thereafter, the heuristic algorithm is performed as depicted in Figure 1. This algorithm allows comparing p_{sijk} s to e_{ijk} s and determining the functional suitability score of *s.* alternative (f_{sijk}) for each tertiary system requirement. By applying this algorithm, functional suitability score for each secondary (f_{sij}) and primary (f_{si}) system requirement, and the overall functional suitability score of *s.* alternative are also obtained. This algorithm produces the scores varying in the range of [0,1].

2.4. Make the "build or buy" decision

The "build or buy" decision is made by using the outputs of the heuristic algorithm. Beginning of this step, an acceptable level of functional coverage, *f*, which indicates the degree of optimism of the decision makers, has to be determined. A larger *f* represents a lesser degree of optimism. If there exist different alternatives with the functional suitability score that is higher than or equal to *f*, then the case in which situation the organization can decide to continue the software selection process. In other words, meeting at least *f* percent of system requirements is enough to purchase an enterprise software for this organization. These alternatives are selected for the detailed evaluation, final selection decision is made and the best alternative is purchased. On the other hand, if no product meets an acceptable level of functional coverage, *f*, as determined by the decision makers, building a custom-designed application is the only way to meet specific system requirements. In this case, the required capabilities must rely on custom development rather than purchasing a packaged product.

3. CONCLUSION

The build-or-buy decision is an important issue that is occasionally faced by organizations in every business. While there are many other factors that can be considered in the build versus buy question, functional suitability of the resulting application or system is one that cannot be ignored. Framework presented in this study allows organizations to evaluate commercially available software alternatives and to obtain functional suitability scores of each product. This framework not only supports the organization's "build or buy" decision, but also provides a basis for the software selection process. In case of choosing "buy" option, the organization has already evaluated the alternatives from functionality points of view. Thereafter, decision makers can measure performances of selected software alternatives to obtain non-functional suitability according to non-functional criteria. A multi objective model that offer a compromise between conflicted goals such as maximizing functional and non-functional suitability, minimizing total cost of ownership and implementation time can be used to make final purchasing decision.

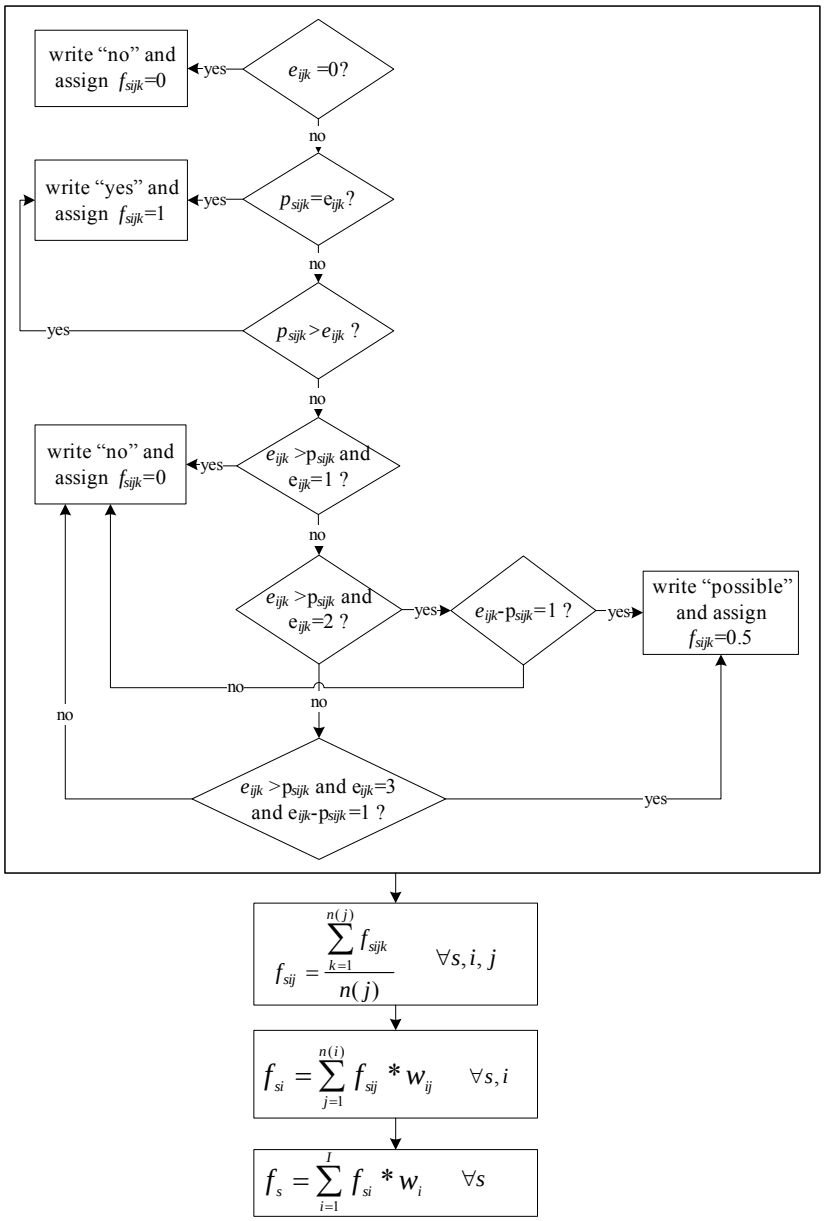

Figure 1. Heuristic algorithm for obtaining the functional suitability scores

4. REFERENCES

[1] The Gartner Group: "Build vs. Buy: New Decision Criteria for Applications" January 31, 1997.,

[2] Nelson, P., Richmond, W., Seidmann, A.: Two dimensions of software acquisition, Association for Computing Machinery, Communications of the ACM, 39(7), July 1996.,

[3] Sherer S.A.: Purchasing Software Systems: Managing the risk, Information & Management 24, 1993.,

[4] Wei C.C., Wang M.J.: A comprehensive framework for selecting an ERP system, International Journal of Project Management 22, 2004.,

[5] Kunda, L., Brooks D.: Applying social-technical approach for COTS selection, In 4th UKAIS Conference, 4th UKAIS Conference Proceedings: Information Systems - The Next Generation, University of New York, USA, 1999.,

[6] Güngör Şen, C., Baraçlı, H., Eser, A.: A conceptual framework for selecting enterprise software with multiple objectives and both qualitative and quantitative criteria, 10th International Research/Expert Conference, "Trends in the Development of Machinery and Associated Technology" TMT 2006, Barcelona-Lloret de Mar, Spain, 11-15 September, 2006.