Social-Technical COTS Development: The STACE Contribution

Laurence Brooks[1] and Douglas Kunda[2]

[1] *School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, Middlesex, UB8 3PH. U.K.*; [2] *Integrated Financial Management Information Systems (IFMIS), Ministry of Finance and National Planning, PO Box 50062, Lusaka, Zambia*

ABSTRACT

COTS-Based Systems (CBS) development is the process of integrating existing software components to create larger software systems. The success of these systems depends on an appropriate evaluation and selection of the software components to match requirements. Previous research has shown that successful selection of 'off-the-shelf' systems to fit customer requirements remains problematic, mainly because of a lack of appreciation of the non-technical issues in the evaluation. This paper presents the outcome of a series of studies aimed at using a social-technical approach to identify and classify processes (including traditional and soft factors) that support COTS software selection. The development of the STACE (Social-Technical Approach to COTS Evaluation) framework is explored to show how the iterative processes relate and influence each other. From this approach, minimizing the risks and address problems in the selection of COTS software is possible.

KEYWORDS

component off-the-shelf, COTS-based systems development, requirements engineering, requirements elicitation, evaluation

Reprint requests to: Laurence Brooks, School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, Middlesex UB8 3PH, U.K.; e-mail: Laurence.Brooks@brunel.ac.uk

1. INTRODUCTION

Information Systems (IS) are now commonly accepted to be at the core of most modern organizations (if not modern life). Nevertheless, as the requirements for functionality and flexibility increase, so does production cost and maintenance cost. One approach seen as a way to combat rising costs while maintaining high functionality and reducing development times is a move to Commercial-Off-The-Shelf (COTS)-Based Systems development (CBS) (Chung & Cooper, 2002). The CBS is the process of building systems by integrating multiple 'off-the shelf' software components, which are self-contained and can be executed independently (Brown & Wallnau, 1996, Haines et al., 1997). Successful CBS development, however, depends on successful evaluation and selection of COTS software components. Within this framework, COTS software component selection can be seen as a process of determining the 'fitness for use' of previously-developed components that are being applied in a new systems context (Haines et al., 1997).

A successful selection of COTS software to fit requirements is still problematic for a number of reasons—lack of a well-defined process (Kontio, 1996), the 'black box' nature of COTS components (Vigder et al., 1996), rapid changes in the market place (Carney & Wallnau, 1998), and the misuse of data consolidation methods (Kontio, 1996, Morisio & Tsoukiàs, 1997).

The major problem with COTS software evaluation, however, appears to be with the lack of attention paid to non-technical issues in the evaluation criteria (Lawlis et al., 2001). Evaluators tend to focus on the technical capabilities at the expense of the non-technical or 'soft' factors, such as human and business issues (Powell et al., 1997). That these softer factors are the most important to consider has been increasingly realized (see Kling, 2000, Walsham, 1993 for discussions that support this argument), yet they are more difficult to include, both from a theoretical and a practical approach.

2. THEORETICAL FRAMEWORK

Previous frameworks have focused on evaluating the technological capabilities for component selection (eg. Boloix & Robillard, 1995, Brown &

Wallnau, 1996, Kontio, 1996, Maiden & Ncube, 1998, Chung & Cooper, 2002), but this approach can lead to significant problems later in the development process.

The approach taken in this paper to addressing these problems is the development of a Social-Technical Approach to COTS Evaluation (STACE)-based component-selection framework STACE (Fig. 1). The STACE approach utilizes a number of interlinked processes in the COTS software-selection process (for detailed discussions of the development of this framework see (Kunda & Brooks, 2000, Kunda, 2003)). In addition, a brief discussion of the STACE elements follows.

Following an initial study to identify processes (including traditional and soft factors) that support COTS software component selection for CBS from the United Kingdom (UK) a number of factors were identified and classified into four major themes (processes): requirement definition, social-technical criteria definition, identification of COTS software alternatives and evaluation (assessment). However, the first study did not investigate the relationship between these processes and their impact on the success of COTS software evaluation and selection, which is the aim for this paper.

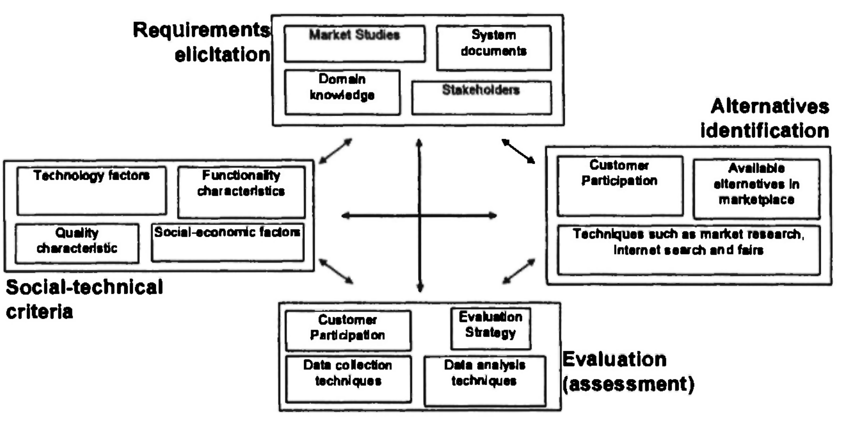

Fig. 1: The STACE framework

2.1 The Social-Technical Approach

The main thrust of the social-technical argument derives from the basis that software systems are used in social and organizational contexts. Further, this view emphasizes that *thorough insight into the work practices in which IT applications will be used should be the starting point for design and implementation* (Berg, 1999). As already specified, the major cause of most software failures is the human, social and organizational issues (see Avgerou et al., 2004, Luna-Reyes et al., 2005). Therefore, the social-technical approach consists of a human subsystem and a technical subsystem working together. As can be seen in Table 1, this approach and philosophy have huge implications for the way that ICTs are perceived and hence for the design/ development process. The social-technical approach originates in studies from the 1950s on relations b etween social structures and technology in organizations.

Briefly, Cherns (1993) developed a set of principles that operationalize the concepts of social-technical systems in such as way that they can be used as a checklist, not a blueprint, for designers to consider. An overview of the important principles is as follows:

Compatibility: The process of design must be compatible with its objectives. If the objective of design is a system capable of self-motivation, of adapting to change and of making the most use of the creative capacities of the individual, then a constructively participative organization is needed.

Minimal Critical Specification: No more should be specified than is absolutely essential. While it may be necessary to be quite precise about what has to be done, it is rarely necessary to be precise about how it is to be done.

The Social-technical Criterion: Variances if they cannot be eliminated, must be controlled as near to their point of origin as possible.

Support Congruence: The systems of social support such as incentives should be designed so as to reinforce the behaviors the organizational structure is designed to elicit.

Design and Human Values: The objective of organizational design should be to provide a high quality of work. Quality is a subjective phenomenon and everyone wants to have responsibility, variety, involvement and growth.

TABLE 1

Conceptions of information and communication technologies (ICT) in organizations/society (Kling, 2000)

Standard (Tool) Models	Social-technical Models
ICT is a tool	ICT is a social-technical network
Business model is sufficient	Ecological view is also needed
One shot ICT implementation	ICT implementations are an ongoing social process
Technological effects are direct and immediate	Technological effects are indirect and involve different time scales
Politics are bad or irrelevant	Politics are central and even enabling
Incentives to change are unproblematic	Incentives may require restructuring (and may be in conflict)
Relationships are easily reformed	Relationships are complex, negotiated, multi-valent (including trust)
Social effects of ICT are big but isolated and benign	Potentially enormous social repercussions from ICT (not just quality of worklife, it's overall quality of life)
Contexts are simple (a few key terms or demographics)	Contexts are complex (matrices of businesses, services, people, technology history, location, etc.)
Knowledge and Expertise are easily made explicit	Knowledge and Expertise are inherently tacit/implicit
ICT Infrastructures are fully supportive	Additional skill & work needed to make ICT work

2.2 Strategies to Apply Social-Technical Approaches

The main issue appears to be the need to integrate social issues with existing requirements engineering (RE) methods. This requirement adds an extra level of analysis that incorporates the social elements and preserves the separateness and apparent strengths of each in addressing different issues, eg. Multiview (Wood-Harper et al., 1985). Within this umbrella exist approaches such as participative design, which involves the participants directly in the RE process and involves analysts using materials drawn from meetings with participants, from user trials of prototypes—for example, methods such as ETHICS, JAD (Mumford, 1990, Beynon-Davies et al., 1999). Therefore, the social and technical aspects are thoroughly intertwined, and this approach attempts to develop analytic categories from the participants themselves, whereby the technical is thoroughly embedded within the social environment.

Social-technical analysis is useful for incorporating the social (non-technical) objectives of the system and so ensuring that the proposed system addresses the correct problem (Mumford, 1990, Flynn, 1998). Therefore, this technique is recommended in the STACE to decompose the high-level requirements into social criteria and technical criteria.

3. STUDY OBJECTIVE AND METHOD

The overall goal of this research is to identify the range of important processes (where a process is a collection of related tasks leading to an outcome) and factors that support COTS software component selection for CBS from organizations in the UK. Such processes and factors are considered essential by experts in the field to minimize the risks and address problems of COTS software selection. Identifying these processes and factors will also assist in defining how an organization is supposed to perform its activities related to COTS software selection and how people work and interact. The outcome of this study is to create further development and refinement of the social-technical framework for COTS components selection (i.e. STACE).

We used a field study approach comprising a set of interviews for the empirical research. This approach allows for cross-organization analysis and comparison, which is important for identifying the patterns to be used in the development of theoretical categories. Focused (semi-structured) interviews were used to provide insight into the phenomena being studied, as well as being amenable to interpretation (Easterby-Smith et al., 1991). The interviews, lasting from 1 hour to 3 hours, were tape recorded and then transcribed. The interviews were supplemented with documentary evidence. Following an interview protocol helps to increase the reliability and sharpen the construct validity of the research. We developed this interview protocol through an extensive review of technical literature, definition of research questions, and definition of construct and field procedures. The definition of research questions assisted in refining the research focus, whereas the definition of construct and field procedures helped to sharpen the construct validity of the research. Following an internal review, the protocol was also pilot tested with an organization experienced in COTS and CBS selection and evaluation. We used theoretical sampling as the basis for selecting organizations for this study, i.e. to focus on organizations that confirmed and extended the theoretical framework.

The semi-structured interview was constructed such that the questions covered relevant areas of the STACE framework. Following several introductory contextual questions, the interview turned to the *Criteria definition* (covering the topics of technology factors, product quality factors, compliance issues, business issues, customer capability, marketplace variables, vendor capability variables); *Search for alternatives* (covering the topics of techniques and tools, COTS availability); and *Assessment* (covering the topics of evaluation strategy, techniques and tools, customer participation); and then concluded with some general concluding questions.

The general mode of analysis used in this study was a five-stage model of explanation building (see Fig. 2):

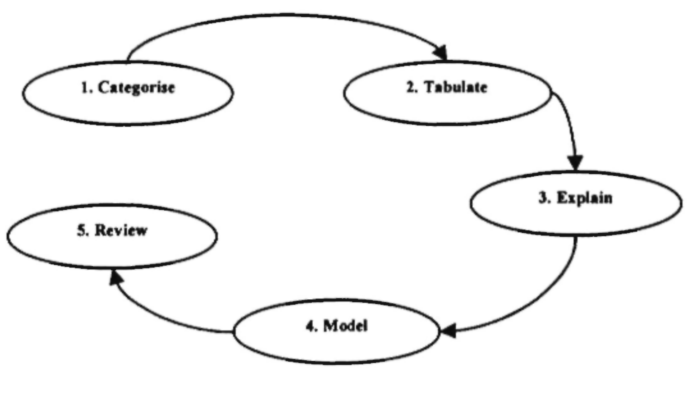

Fig. 2: Five stage model of explanation building

1. *categorize*–identify concepts and develop associated coding categories;
2. *tabulate*–create tables for data display allowing valid conclusions to be drawn;
3. *explanation*–provide explanations that validate the relationships between higher level coding categories;
4. *modelling*–build networks based on the relationships between higher level coding categories;
5. *review findings*–key informants review the draft field study reports to increase construct validity and facilitate selection of other organizations. ATLAS/ti™, a qualitative software analysis tool, which also acted as field study database, supported the data analysis process.

The use of a software tool and field study database enhanced the reliability and validity of the findings, whereas the rigorous nature of the data collection and analysis procedures supports validation of the findings (further discussion about this can be found in Kunda, 2001).

Table 2 shows an example of the effects matrix that was generated by searching for the codes of the effect of the requirements definition process on the evaluation (assessment) process. The quotations that match these codes were then categorized according to confirmatory or contradictory evidence. An explanation then was formulated and conclusions were confirmed (i.e. tested or verified) by focusing on negative evidence or checking out rival explanations. The relations in the effect matrices were entered as network links and explanations were stored as memos in the ATLAS/ti™ software. The process of generating an effects matrix is similar to the selective coding in grounded theory. Selective coding is the process of selecting the core category, systematically relating it to other categories, validating those relationships, and filling in those categories that need further refinement and development (Strauss and Corbin, 1990).

4. BACKGROUND TO THE CASE ORGANIZATIONS

Sixteen in-depth interviews were conducted in eight organizations within the UK, selected on the basis of their experience in CBS and a variety of

TABLE 2

Example of explanatory effects matrix

Variable 1: Requirements definition
Variable 2: Evaluation (assessment) process
Effect: The evaluation (assessment) process depends on the requirements definition.
Confirmatory evidence
In terms of best practices, it is recommended that you let users draw up specifications of what they want before embarking on purchasing COTS packages. P 1: tJames.txt - 1:22 (79:82)
The process of evaluation, you begin with high-level criteria (although we would have written requirements); you look at the documentation and this process sometimes is a nightmare because manuals are not available; and you arrange training course or vendor demonstration in order to understand the package. P 2: tDavid.txt - 1:39 (53:58)
Contradictory evidence
In component selection, you first have to find out who has the components that meet your need, you may find the component out there that meet half your needs and the other half does not or too much your needs. You have to make a decision whether to get a component that gives more than you actually need and pay for that or to get a component that delivers slightly less than you need but allow to extend it. P 3: tJohn.txt - 2:8 (99:103).
In COTS software selection, you should search for alternative COTS packages before procuring and let the supplier do some presentations. It is also important to understand the licensing arrangements before procurement. P 1: tlames.txt - 1:23 (83:85)
Explanations
E1: The importance and effect of the requirements definition on the evaluation (assessment) process is supported by the quotations above. The respondents were arguing that it is important to define the requirements prior to evaluation (i.e., COTS software selection must be driven by requirements).
E2: The contradictory evidence suggests that COTS software selection process must be driven by what is available in the marketplace. However, this is not contradictory because to realize the benefits from COTS software, the requirements must be defined according to what is available from the marketplace.

systems development techniques. In addition, each case was selected in the order that it was either predicted to produce similar results (literal replication) or to produce contrasting results but for predictable reasons (theoretical replication) (Yin, 2003). Furthermore, a deliberate effort was made to ensure that a wide variety of organizations from different sectors were included in the field study (see Table 3).

TABLE 3

Background information about participant organizations

Organization		Main business area and overview	Interview	No
No.	Size			
1	Large	*Software house*: provides consulting services that enables people from across an organization to work together, defining, communicating and improving the way their business works. This organization develops software tools to help its customers develop their process, people and technology assets in a synchronized way, enabling them to deliver sustainable value. The organization had an annual turnover of about £675,000 in 1997.	The interview was conducted with a senior consultant responsible for project management, workflow systems and business modelling, and software development (integrating different modeling tools).	1
2	Large	*Research and development*: one of the world's largest independent contract research, development and testing organizations. The organization employs an expanding team of 500 research scientists, engineers and support personnel. In terms of software engineering the organization has experience in software reliability; safety critical software; software testing; software development; Internet and Intranet technologies; systems integration.	Three separate interviews were conducted with senior members of staff at this organization. The applications of COTS software components that were investigated in this organization were high integrity systems and multimedia systems.	3
3	Small	*Local authority*: provides services to 175,000 people including education, social services, leisure, highways, planning, cleaning and many more. The information Technology and telecommunications (IT) department provides support for computer systems throughout the authority. The organization is funded from local taxes and government support.	The two interviews were conducted with IT managers.	2
4	Large	*Manufacturing/Engineering*: a world leader in the power systems business, providing cost-effectively engineered products and services to commercial and military customers in propulsion, electrical power and materials handling markets around the world. It has a turnover of over £1 billion with over 42,000 employees and customers in 135 countries.	The interviews were conducted with three senior members of staff in the department responsible for software development and maintenance.	

Organization		Main business area and overview	Interview	No
No.	Size			
5	Large	*Finance/Banking*: offers a number of specialist business services including business banking, commercial banking, corporate and institutional banking in the UK and overseas. At the end of 1997 total Group assets for this organization were £176 billion and there were over 76,000 employees. The IT department has over 1500 software developers dedicated to developing and maintaining software systems.	The two interviews were conducted members of the Architecture group, who are responsible for developing both business and software architectures for the whole organization.	2
6		*Consulting services*: one of the market leaders in management and financial services consultancy. The management services specialize in software solutions for personnel, point of sales systems for sales forces, training, design and publishing of interactive media used in the sales process. The organization has offices in Australia, Canada, Hong Kong, USA, Ireland, Malaysia, Spain, and South Africa.	The interview was conducted with the managing consultant who was the key person in the development of their personnel tracking system for financial services sytem.	1
7		*Retail/Wholesale*: a leading retailer with over 370 stores in the United Kingdom, Europe, Hong Kong and Canada. It employs 68,208 people around the world and achieved a group turnover of over £8 billion. Although outside contractors do most of the software development, the organization still maintains a pool of its own analysts and developers.	Interviews were conducted with three senior members of staff from Stores System Project (SSP) and Contracts Management System (CMS).	3
8		*Telecomms*: one of the leading providers of telecommunication services in the UK. They have also a world-wide presence through a series of subsidiaries, equity ventures and distributorships in Europe, Asia Pacific and the Americas delivering a comprehensive multi-local strategy covering the datawave, the IP world, mobile, multimedia and fixed to mobile convergence. The organization has over 1500 employees and a turnover of over £1 billion.	The interview was conducted with one individual involved in the component-based development (CBD) project within the organization. Briefly, their CBD project aimed to investigate the impact of component-based development on the organization, in terms of whether they would have to then change their processes.	1

5. RESULTS AND DISCUSSION

5.1 Effects of Requirements Elicitation/Definition Process on Other STACE Processes

Four cases indicated that the requirements elicitation/defmition process has significant effect on the social-technical criteria definition (see Fig. 3 and Table 4). For example, respondents from case 3 indicated that initially they write an outline of the basic functional specification of what the system will do. Following on from this, the functional specification is then converted into a technical specification or criteria. The technical specification or criteria is then used as the basis for selection among different vendors.

Similarly, case 4 indicated that the process of evaluation begins with high-level criteria derived from user requirements (i.e. can the tool support a multi-user environment as well as being able to communicate with others tools). The respondents from case 4 argued that the high-level user requirements (the reason for wanting a system) must be defined prior to the definition of criteria. They explained that the high-level user requirements can be a new capability or new technology or a new process to improve the process and support strategic change, such as 'example changing from UNIX to PC based sy stems'.

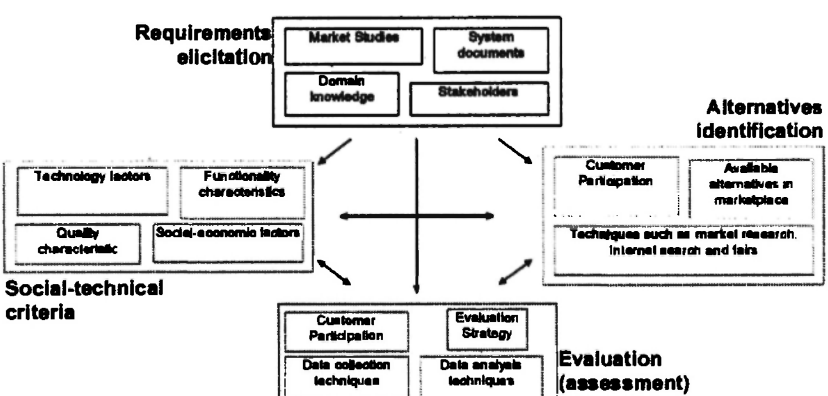

Fig. 3: STACE indicating influences of requirements elicitation/definition process on other processes

TABLE 4

Effects of social-technical criteria definition process on other STACE processes

Process Affected by Criteria	Case #							
	1	2	3	4	5	6	7	8
Requirements						×		
Identification	×			×			×	
Evaluation	×	×	×	×		×	×	

(× indicates data matched)

Respondents from case 6 pointed out that when evaluating COTS software packages, it is important to prepare an invitation to tender (ITT) document, which transforms the requirements definition into a technology-type definition. Respondents from case 8 argued that business owners and design teams are best placed to consider and decide on software acquisition for their business area by specifying components in terms of business and systems capabilities. The findings suggest that eliciting the high-level requirements prior to the COTS software evaluation process is important and further that the high-level requirements must be changed into evaluation criteria. This view is consistent with literature showing how the evaluation criteria definition process essentially decomposes the requirements for the COTS into a hierarchical criteria set (Kontio, 1996).

Three cases indicated that the identification of candidate COTS software from the marketplace must be driven by some kind of high-level requirements (see Table 4). Respondents from case 3 argued that good practices for evaluating COTS software require that users first draw up specifications of what they want before embarking on purchasing COTS software packages. Then the evaluation team must search for candidate COTS packages and request suppliers to make presentations highlighting the important features of the products. The respondents argued that searching for candidate COTS software from the marketplace is important for meeting tender procedures, which stipulate that valid evaluation must have at least three suppliers.

Case 6 supported the findings of case 3 and indicated further that an ITT document, which is used as a basis for identifying COTS software from the marketplace, must be prepared. The advantage with the ITT is that the vendors respond to the ITT, informing the evaluators of the availability of their products, rather than the evaluators searching for the products from the marketplace. Similarly, respondents from case 8 pointed out that the business owners and design teams specify components in terms of business and systems capabilities and then identify candidate COTS software components from the marketplace. This point suggests that the identification of candidate COTS software from the marketplace depends on the requirements definition process.

Four cases indicated that COTS software evaluation and selection must be driven by requirements definition process (see Table 4). For example, respondents from case 2 indicated that the evaluator must check that the functionality of the COTS software product meets high-level-user requirements and that it performs well on the basic test routines (quality attributes). Similarly, respondents from case 3 recommended that users must draw up specifications of what they want, identify candidate COTS software packages, and then evaluate the candidate COTS software packages. The respondents from case 3 argued that the supplier must be asked to make presentations and to indicate the licensing arrangements clearly before procurement.

This finding was supported by case 7—that it is important to experiment with the COTS software to test quality attributes and assess how it fits within the organization's own component model. The respondent further argued that the technology it supports is a significant factor. Therefore, the COTS software component should be acquired only if it is on the appropriate technology platform because the technological environment must be managed carefully. Respondents from case 8 indicated that they started with user requirements and then proceeded to check if the 'off the shelf' components would be able to support that process.

5.2 Effects of Social-technical Criteria Definition on Other STACE Processes

The respondents from case 6 pointed out that at times they revise the evaluation criteria for the requirements (see Table 5). The respondents

TABLE 5

Effects of identification of COTS software process on other STACE processes

Process Affected by Identification	Case #							
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Requirements					×			×
Criteria				×				×
Evaluation	×			×	×	×		×

(× indicates data matched)

indicated that this happens when the evaluated candidate COTS software packages do not meet all the high-level requirements but management still want a COTS software solution. The evaluators are asked to revise the requirements, based on available COTS software characteristics, transforming these attributes (criteria) into the requirements. This approach is similar to the experiences of Sledge and Carney (1998) in evaluating COTS products for United States Department of Defense information systems in the domain of human resources and personnel management.

Similarly, case 4 indicated that the evaluation (assessment) affects the requirements definition and criteria definition. Respondents from case 4 pointed out that the COTS software evaluation begins with an initial evaluation of the vendors and the products, then the criteria and requirements are refined based on the screened products. This approach suggests that the requirements will be influenced by the product features and criteria definition, which is consistent with literature that COTS software evaluation is an iterative process between evaluation and requirements definition (Maiden & Ncube, 1998). Nevertheless, SEL (1996) cautions against this strategy of revising the requirements based on available products and points out that it is important that the evaluation criteria and requirements are not revised in such a way that only one product can be selected.

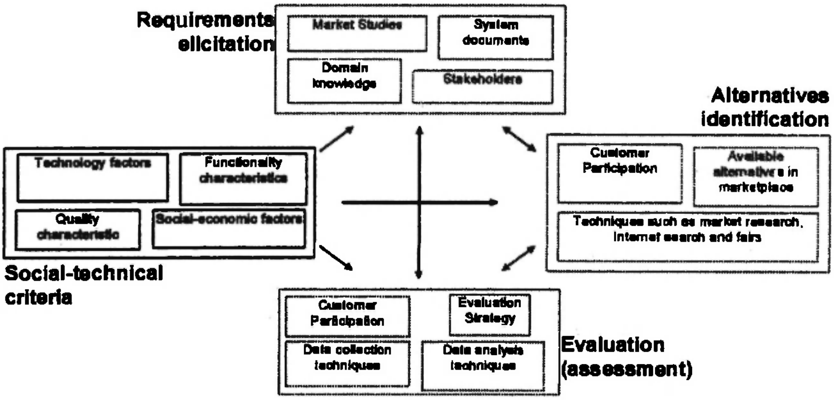

Fig. 4: STACE indicating influences of social-technical criteria definition process on other processes

Kontio (1996) points out that the initial search and screening for candidate COTS software depends on the evaluation criteria. This view was echoed by respondents from case 1, who argued that they selected the design tool based on what was considered industry standard and market viability (curve). Similarly, respondents from case 4 argued that the evaluation criteria must be defined first—for example, whether the product supports multi-user processing. The identification of products in the market must be based on the evaluation criteria and this is done through market survey, interviews, other research, Internet access, publication, safety engineers, etc.

Respondents from case 7 also supported these findings and indicated that the identification of products from the external marketplace or from the organization component repository must be based on the defined criteria. They argued for example, that evaluators must search for information on functionality, interface type, and technology instantiation, suggesting that the social-technical criteria greatly influence the identification of candidate COTS software products from the marketplace. Therefore, if the criteria are not well defined, then inappropriate products will be selected. Kontio (1996) found that the evaluation definition is seldom well defined—for example the phrase "ease of use" leaves the exact meaning of a criterion open to each evaluator's own interpretation. Therefore, operational definitions for criteria are recommended so that all COTS alternatives are compared against a common yardstick.

In the STACE framework, the COTS software components are assessed against the social-technical criteria. That is, the criteria have significant impact on the evaluation process—for example, well-defined criteria will lead to a better evaluation process. Respondents from case 3 indicated that evaluation is based not only on technical criteria but also on other organizational and social factors. For example, vendors are evaluated in terms of services that they provide, their financial stability, and the vendors' understanding of modernizing the local council agenda. The respondents argued that the product must be evaluated as long as the major functionality is nearly there and it is perceived cheaper to do the job.

Furthermore, the respondent from case 1 pointed out that the stock market view of a specific COTS software, whether it is on the upward curve (competitive curve), is an important criterion that people use to evaluate and select products. In addition, they indicated that documentation is studied to assess whether the tool supports other modeling formats apart from the proprietary format. Similarly, the respondents from case 2 provided an example in which the organization used reliability data that were being kept by the COTS software provider to evaluate safety of the software.

Cases 6 and 7 supported the importance of defining the evaluation criteriabefore embarking on the evaluation (assessment) and suggested that the evaluation criteria should include functionality, quality attributes, cost, the technology it supports, and the organization's component model. The findings are consistent with literature, indicating that definition of the evaluation criteria for the selection of COTS software products is a very important task in CBS (Kontio, 1996, Tran et al., 1997). For example, Tran et al. argue that selecting an appropriate product typically requires trade-off analyses among the available products in which only a "better than others" solution is available and, therefore, establishing the evaluation criteria is important.

5.3 Effects of Alternatives Identification of COTS Software on Other STACE Processes

Cases 5 and 8 highlighted the influence of COTS software identification on requirements and argued that available COTS software products must drive it (see Fig. 5 and Table 6).

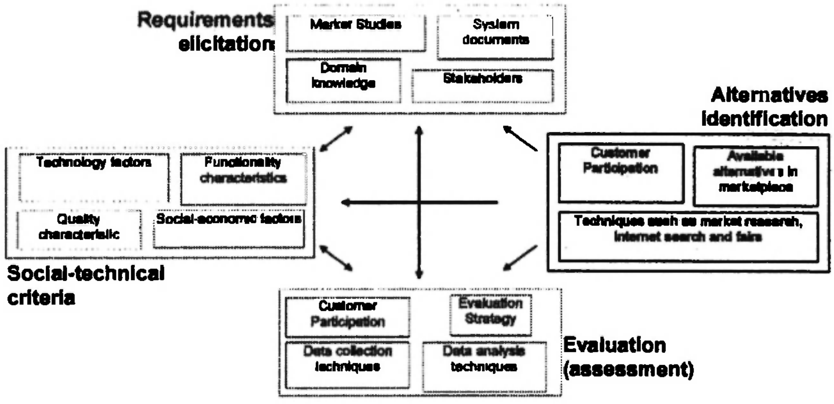

Fig. 5: STACE indicating influences of alternatives identification process on other processes

TABLE 6

Effects of identification of COTS software process on other STACE processes

Process Affected by Identification	Case number							
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Requirements					×			×
Criteria				×				×
Evaluation	×			×	×	×		×

(× indicates data matched)

Respondents from case 5 pointed out that first the evaluator must find out what components are in the marketplace, even if the components only meet half the requirements. The component can then be extended and the requirements can be revised as well. The argument here is that the COTS software available in the market place drives the requirements definition.

Similarly, respondents from case 8 indicated that one of the objectives of the CBD research project was to use COTS software components available in the marketplace to understand the requirements in the organization. For example, using COTS software for customer ordering, the project investigated how it can be applied in the different departments within the organization. The research outcome of the CBD project was to provide business managers with the list of COTS software components and potential applications with the organization.

The findings from cases 5 and 8 suggest that the available COTS software products greatly influence the requirements. This is consistent with literature finding that to realize the benefits of COTS software, needs a current procurement process that defines requirements according to what is available in the marketplace and that is flexible enough to accept COTS solutions when they are proposed (Vigder et al., 1996).

Respondents in case 4 indicated that the COTS software product must first be identified from the marketplace through Internet search, word of mouth from colleagues, attending conferences and published materials. Then the identified products must be initially evaluated to screen for suitable candidate products, which is achieved by interviewing the product suppliers. Finally, demonstration copies of candidate products are obtained for in-depth evaluation against the set criteria. This suggests that the evaluation criteria will be revised based on available COTS software products.

Respondents from case 8 pointed out that the CBD project provided business managers with a list of potential COTS software components from the marketplace and their functionality in the form of a checklist. Then the managers refine the evaluation criteria and decide whether the products offer them the functionality they want or not. This finding suggests that features of the products in the marketplace influence the definition of the evaluation criteria. Finkelstein et al. (1996) suggest that the initial requirements (and criteria) are revised on the basis of advertisements, package descriptions provided by suppliers, demonstrations, use of packages and comparative studies provided by third parties (trade papers, etc.).

In the literature, the importance of identifying the appropriate candidate products for evaluation has been highlighted (Kontio, 1996, Tran et al., 1997). Tran et al. argue that the selection of an inappropriate candidate product for integration can result in an enormous amount of extra time and effort to re-evaluate and to re-implement the system with another product.

This problem was identified in six cases (see Table 5). Respondents in case 4 indicated that the evaluation process began by defining the high level criteria and then searching for products in the market that meet the criteria using various techniques, such as market survey, interviews, word of mouth from colleagues, Internet search, and publications. Then the selected candidate products are evaluated.

The respondent from case 1 also indicated that a certain product was included in the evaluation because it was considered industry standard in the marketplace. Cases 5, 6, and 8 supported this finding and highlighted the importance of allocating human resources and the time to identify appropriate COTS products from the marketplace. This view suggests that the evaluation (assessment) success depends on the availability and successful identification of these products.

5.4 Effects of Evaluation (Assessment) on Other STACE Processes

Cases 4 and 6 supported the proposition that the evaluation (assessment) influences the requirements definition (see Table 7). Respondents from case 4 argued that it is important to refine the business case—for example, the benefits of selecting a particular COTS software product. They provided an example of an evaluation that they had begun with a number of requirements for a housing rental system. After evaluating some products, however, they observed that one of the benefits of the system was a rent-collection feature

TABLE 7

Effects of evaluation (assessment) process on other STACE processes

Process Affected by Evaluation	*Case #*						
	1	*2*	*3*	*4*	*5*	*6*	*7*
Requirements				×		×	
Criteria				×		×	
Identification			×				

(× indicates data matched)

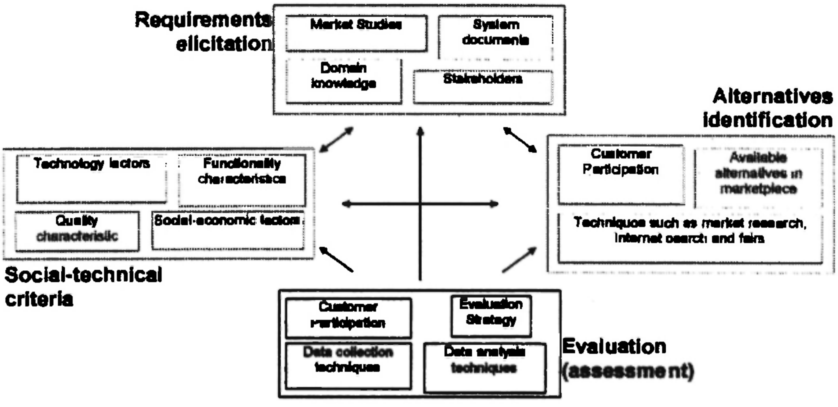

Fig. 6: STACE indicating influences of evaluation (assessment) process on other processes

that would help them to recover the cost of purchasing the COTS products within 6 months. This feature became an important requirement for the organization, suggesting that evaluation can influence customer requirements.

The respondents from case 6 argued that in certain circumstances it is important to revise requirements according to the available COTS software products. This revision happens when the identified candidate COTS software products are evaluated and found not to meet the mandatory requirements, but management still wants a COTS software solution. Furthermore, previous evaluation results in a similar application domain can help to define the requirements and the criteria. This is consistent with literature that storage and management of past evaluation results can help in new evaluation problems particularly in the same application domain (Stamelos et al., 2000).

Respondents from case 4 indicated that the evaluation process begins with identifying the candidate COTS software from the marketplace and then an initial evaluation follows involving vendor analysis and attending vendor demonstration. The evaluation criteria are then refined and demonstration copies are obtained and evaluated through 'hand on' experimentation. Similarly, case 6 indicated that the evaluation criteria are usually revised when the available COTS software components do not fully satisfy the initial high-level requirements.

The findings indicate that the evaluation (assessment) can influence social-technical criteria definition, suggesting an iterative and repeatable process. Kontio (1996) argues that organizations that evaluate COTS frequently benefit from a well-defined, repeatable selection process that facilitates planning, allows for the accumulation of experience, enables a consistent selection process, supports the use of validated methods, and increases the efficiency of evaluation.

The respondents in case 3 pointed out that the evaluation process normally begins with a department showing interest in some particular COTS software product and experimenting with it. Because tender procedures require that at least three products must be evaluated, however, the department is advised to identify alternative products. The problem with this procedure is that most likely the department or evaluators will be biased toward the first product identified. Vigder et al. (1996) argue that the criteria must not be defined such that only one product can be selected.

Respondents from case 8 indicated that results from previous assessments of suppliers should be used to inform the process of identifying COTS software products. This is consistent with the finding of Stamelos et al. (2000) that it is important to learn from previous evaluation and reuse the results. The problem with this approach is that most likely the information of previous assessment will be outdated.

6. CONCLUSION AND LESSONS LEARNED

The paper has presented the results of the assessment of the relation between the processes that support COTS software component selection using the STACE framework. The relation between the STACE framework processes (requirements definition, social-technical criteria definition, alternative identification, evaluation) suggests that COTS software selection is an iterative process. A number of lessons have been learned from this study:

- It is important to elicit the high-level requirements before the COTS software evaluation process. The high-level requirements can be used to define the evaluation criteria and for identifying COTS software from the marketplace.

- A procurement process must be in place that defines requirements according to what is available in the marketplace in order to realize the benefits of COTS software.
- The use of the social-technical evaluation criteria is important for the selection of COTS software products from the marketplace; Dubois and Franch (2004) state that COTS evaluation is central to successful COTS development. This view is consistent with literature indicating that selecting an appropriate product typically requires trade-off analysis among the available products in which only a "better than others" solution is available and therefore it is important to establish the evaluation criteria (Kontio, 1996, Tran et al., 1997).
- The results from previous assessments (evaluations) of COTS software and suppliers can be used to inform future evaluation. For example, the evaluation criteria can be reused for evaluation in the same application domain. This approach is consistent with literature showing how it is important to learn from previous evaluation exercises and to reuse the results (Kontio, 1996, Stamelos et al., 2000).

The lessons learnt from the case study assisted in elaborating and validating the STACE framework. This is a generic social-technical framework for COTS software evaluation and selection, which provides a classification of important processes, factors, techniques, and tools for COTS software selection. The framework can also be used by software engineers and information systems professionals to plan and conduct COTS software selection for COTS-based systems. Although not a complete answer to the multi-faceted problems faced by information systems development, the STACE framework does go some way to help guide the direction for both future theoretical work, as well as being a practical tool, which is usable in real contexts.

REFERENCES

Avgerou, C., Ciborra, C. and Land, F. 2004. *The social study of information and communication technology: innovation, actors and contexts*, Oxford, New York, USA, Oxford University Press.

Berg, M. 1999. Patient care information systems and health care work: a sociotechnical approach, *International Journal of Medical Informatics*, **55**, 87–101.

Beynon-Davies, P. Carne, C. Mackay, H. and Tudhope, D. 1999. Rapid application development RAD: an empirical review, *European Journal of Information Systems*, **8**, 211–223.

Boloix, G. and Robillard, P. 1995. A software system evaluation framework, *IEEE Computer*, **28**, 17–26.

Brereton, P. 2004. The software customer/supplier relationship, *Communications of the ACM*, 47, 77–81.

Brown, A.W. and Wallnau, K.C. 1996. Engineering of component-based systems, *Proceedings of the 2nd IEEE International Conference on Engineering of Complex Computer Systems*, IEEE Computer Society Press.

Carney, D. J. and Wallnau, K. C. 1998. A basis for evaluation of commercial software, *Information and Software Technology*, **40**, 851–860.

Cherns, A. 1993. Principles of socio-technical design, in: *The social engagement of social science*, Vol. 2, edited by Trist, E. and Murray, H., Philadelphia, Pennsylvania, USA, Penn Press.

Chung, L. and Cooper, K. 2002. A knowledge-based COTS-aware requirements engineering approach, *Fourteenth International Conference on Software Engineering and Knowledge Engineering SEKE '02*. ACM Press, 175–182.

Dubois, E. and Franch, X. 2004. International workshop on models and processes for the evaluation of COTS components MPEC'04, *ACM SIGSOFT Software Engineering Notes*, **29**, 1–3.

Easterby-Smith, M. Thorpe, R. and Lowe, A. 1991. *Management research: an introduction* (Sage Series in Management Research), London, UK, Sage.

Finkelstein, A. Spanoudakis, G. and Ryan, M. 1996. Software package requirements and procurements, *Proceedings of the 8th International Workshop on Software Specification and Design*, IEEE Computer Society Press, 141–145.

Flynn, D.J. 1998. *Information systems requirements: determination and analysis*, London, UK, McGraw-Hill.

Haines, G. Carney, D. and Foreman, J. 1997. Component-based software development/COTS integration, in: *Software technology review*, Pittsburgh, Pennsylvania, USA, Carnegie-Mellon Software Engineering Institute.

Jirotka, M. and Goguen, J. 1994. *Requirements engineering: social and technical issues*, London, UK, Academic Press.

Kling, R. 2000. Learning about information technologies and social change: the contribution of social informatics, *The Information Society*, **16**, 3.

Kontio, J. 1996. A case study in applying a systematic method for COTS selection, *Proceedings of the 18th International Conference on Software Engineering ICSE*. IEEE Computer Society, 201–209.

Kunda, D. 2001. *A social-technical approach to selecting software supporting*

COTS-based Systems, PhD thesis, Department of Computer Science, University of York, York, UK, 1–286.

Kunda, D. 2003. STACE: social technical approach to COTS software evaluation, in: Lecture Notes in Computer Science, Volume 2693, *Component-based software quality: methods and techniques*, edited by Cechich, A. Piattini, M. and Vallecillo, A., Heidelberg, Germany, Springer-Verlag GmbH, 64–84.

Kunda, D. and Brooks, L. 2000. Identifying and classifying processes traditional and soft factors that support COTS component selection: A case study, *European Journal of Information Systems*, **9**, 226–234.

Lawlis, P.K. Mark, K.E. Thomas, D.A. and Courtheyn, T. 2001. A formal process for evaluating COTS software products, *IEEE Computer*, **34**, 58–63.

Luna-Reyes, L.F. Zhang, J. Gil-Garcia, J.R. and Cresswell, A.M. 2005. Information systems development as emergent socio-technical change: a practice approach, *European Journal of Information Systems*, **14**, 1–13.

Maiden, N.A. and Ncube, C. 1998. Acquiring COTS software selection requirements, *IEEE Software*, **15**, 46–56.

Morisio, M. and Tsoukiàs, A. 1997. IUSWARE: a methodology for the evaluation and selection of software products, *IEEE Proceedings of Software Engineering*, **144**, 162–174.

Mumford, E. 1990. *Designing Human Systems*, Manchester, UK, Manchester Business School.

Powell, A. Vickers, A. Lam, W. and Williams, E. 1997. Evaluating tools to support component based software engineering, *Proceedings of the Fifth International Symposium on Assessment of Software Tools*, Los Alamitos, California, USA, IEEE Computer Society, 80–89.

Saaty, T.L. 1990. *The analytic hierarchy process*, New York, NY, USA, McGraw Hill.

SEL 1996. *SEL package-based system development process*, Software Engineering Laboratory.

Sledge, C. and Carney, D. 1998. Case study: evaluating cots products for DoD information systems, in: *Sei monographs on the use of commercial software in government systems*, Pittsburgh, Pennsylvania, USA, Carnegie-Mellon Software Engineering Institute.

Stamelos, I. Refanidis, I. and Tsoukiàs, A. 2000. Knowledge based evaluation of software systems: a case study, *Information and Software Technology*, **42**, 333–345.

Strauss, A. and Corbin, J. 1990. *Basics of qualitative research: grounded theory procedures and techniques*, London, UK, Sage Publications.

Tran, V. Liu, D. and Hummel, B. 1997. Component-based systems development: challenges and lessons learned, *Proceedings of the Eighth IEEE International Workshop on Software Technology and Engineering Practice*

Incorporating Computer Aided Software Engineering, IEEE Computer Society, Los Alamitos, California, 452–462.

Vigder, M.R. Gentleman, W.M. and Dean, J. 1996. *COTS software integration: state of the art*, National Research Council of Canada.

Walsham, G. 1993. *Interpreting information systems in organizations*, Chichester, UK, Wiley.

Wood-Harper, A. T. Antill, L. and Avison, D.E. 1985. *Information systems definition : the multiview approach*, Oxford, UK, Blackwell Scientific.

Yin, R.K. 2003. *Case study research: design and methods*, Thousand Oaks, California, USA, Sage Publications.