2000

Identifying and Classifying Processes (traditional and soft factors) that Support COTS Component Selection

Douglas Kunda
University of York, douglas.kunda@cs.york.ac.uk

Laurence Brooks
University of York, laurence.brooks@cs.york.ac.uk

search

Identifying and Classifying Processes (traditional and soft factors) that Support COTS Component Selection: A Case Study

Douglas Kunda and Laurence Brooks
Department of Computer Science
University of York,
York YO10 5DD, UK
{Douglas.Kunda, Laurence.Brooks}@cs.york.ac.uk

***Abstract*-COTS-Based Systems (CBS) development focuses on building large software systems by integrating previously existing software components. CBS success depends on successful evaluation and selection of Commercial-Off-The-Shelf (COTS) software components to fit customer requirements. Literature shows that successful selection of off-the-shelf systems to fit customer requirements remains problematic.**

This paper presents the outcome of a study aimed at using a social-technical approach to identify and classify processes (including traditional and soft factors) that support COTS software selection. The identified factors and lessons learnt from case study assisted in elaborating and further development of Social-Technical Approach to COTS Evaluation framework (STACE).

I. INTRODUCTION

Modern Information Systems are becoming increasingly expensive to build and maintain; therefore organizations are turning to CBS. CBS can potentially be used to reduce software development and maintenance costs, as well as reducing software development time by bringing the system to markets as early as possible. According to Oberndorf, the term "COTS" refers to things that one can buy, ready-made, from some manufacturer's virtual store shelf (e.g., through a catalogue or from a price list) [1]. It carries with it a sense of getting, at a reasonable cost, something that already does the job. It replaces the scenario of developing unique system components with the promises of fast, efficient acquisition of cheap (or at least cheaper) component implementations. Examples include word processors, operating systems, libraries and functional modules.

COTS packaged application selection can be seen as a process of determining "fitness for use" of previously-developed components that are being applied in a new system context [2]. Component selection is also a process for selecting components when a marketplace of competing products exists. Selection of a component can also extend to include qualification of the development process used to create and maintain it (for example, ensuring algorithms have been validated, and that rigorous code inspection has taken place).

The current approaches and proposed frameworks for COTS software selection do not adequately deal with the human, social and organizational issues. For example the Technology Delta framework does not incorporate an appreciation of political and economic factors [3]. The STACE framework has been developed to apply the social-technical approach to COTS software selection process. The STACE framework has been developed through literature survey and empirical studies and is fully documented elsewhere [4]. STACE has been greatly influenced by the work of Kontio [5] and Maiden [6]. STACE is based on a number of important principles:

1. Support for a systematic approach to COTS evaluation and selection. Most organizations currently select their COTS components in an ad-hoc manner.
2. Support for evaluation of both COTS products and the underlying technology. Most COTS evaluation frameworks currently emphasize either COTS products evaluation or technology evaluation.
3. Use of social-technical techniques to improve the COTS software selection process. Research has shown that most software failures are due to human, social and organizational issues rather than the technical issues [7].
4. Use of AHP a multi-criteria decision-making technique to consolidate evaluation data.

This paper presents findings of a case study aimed at using social-technical approach to identify and classify processes (including traditional and soft factors) that support COTS software selection. The preliminary results of the qualitative interviews are presented in this paper. The results from this case study assisted in further elaboration and validation of STACE framework. Furthermore the STACE framework has also proved an invaluable tool in identifying and classifying important factors in COTS software selection.

II. SOCIAL-TECHNICAL APPROACH

Software systems do not exist in isolation they are used in social and organizational contexts. Experience and many studies show that the major cause of most software failures is the human, social and organizational issues rather than technical issues [7]. Even with the availability of a wide range of advanced software development methodologies, techniques and tools, serious problems with software are still being faced. It is the people and culture of the organization that determines how any system is used. For example poor training may result in people not co-operating with an information system leading to failure and project abandonment [8].

A lot of effort has been made by researchers and software developers to address these organizational issues. For instance, a social-technical development approach. The social-technical approach is a method to develop a system that consists of a human subsystem and a technical subsystem. Social-technical development is oriented to developing both subsystems in an integrated way, such that the integrated system functions in an optimal way. This development strategy has its origin in studies from the 1950s on the relationship between the social structure and technology in organizations [9].

Cherns developed nine principles for organizational designers to use that operationalize the concepts of sociotechnical systems theory [10]. He suggested that these be used as a checklist, not a blueprint for designers to consider. An overview of the important principles are:

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 Socio-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 I encapsulates the differences between the traditional and social-technical view of organizations [11].

According to Jirokta there are several strategies that have been adopted to apply the social-technical approaches to software engineering [12]. One such approach is to integrate the social issues with existing requirements engineering methods. In this case, an extra level of analysis could be added that incorporates the social elements. This would preserve the separateness and apparent strengths of each in addressing different issues, which are to be combined subsequently in some way. Multiview is a good example of this approach [8].

Other social-technical approaches include participative design and ethnography during requirement engineering phase. Participative design involves the participants directly in the requirements engineering process. Here analysts use materials drawn from meetings between participants and designers, or from user trials of prototypes.

TABLE I
DIFFERENCE BETWEEN TRADITIONAL AND SOCIAL-TECHNICAL VIEW

Traditional View	Social-technical View
People are extensions of machines	People are complements to machines
People as expendable spare parts	People as a resource to be developed
Maximum task breakdown	Optimum task grouping
Narrow skills	Multiple broad skills
Autocratic management style	Participative management styles
Competition	Collaboration

An example of participative approach is the Effective Technical and Human Implementation of Computer-Based Systems (ETHICS) [8]. In ethnographic designs, the social and technical are seen as thoroughly intertwined and this approach attempts to develop analytic categories from the participants themselves. Here the technical is thoroughly embedded within the social environment.

III. CASE STUDY OBJECTIVE AND METHOD

The overall goal of this study is to identify processes (including traditional and soft factors) that support COTS software component selection for CBS from the industry. The following are the immediate objective of the case study:

Identify problems (and solutions) experienced by companies in evaluating COTS components for CBS;

Elicit techniques and tools for evaluating COTS components and consider in what ways they promote evaluation success;

Review the COTS components evaluation criteria used by companies (whether it includes the social-economic factors);

Investigate in what ways customer participation can contribute to COTS components evaluation success; and

Classify the identified processes for COTS software selection using social-technical theory.

This study would provide an opportunity to learn the best practices from the industry regarding evaluation and selection of COTS. The outcome of this study would lead to the development of a validated framework for selecting COTS components. The framework would also incorporate the often-neglected non-technical issues.

A multiple case study approach has been used to identify the factors that support COTS component selection. Data collection for this case study relied on face-to-face interviews and documentation. The advantage with the interview technique is that it focuses directly on case study topic and provides perceived causal inferences. The major criticism against interviews is that it expensive and bias may result from responses or poorly constructed questions. However documents were collected from respondents to corroborate and augment the evidence collected through interviews invoking a process called triangulation. With triangulation, the potential problems of construct validity also can be addressed, because the multiple sources of evidence essentially provide multiple measures of the same phenomenon [13].

A grounded theory approach was used to analyse data from this case study. Grounded theory is where the analysis is inductively derived from the study of the phenomenon it represents [14]. That is, it is discovered, developed, and provisionally verified through systematic data collection and analysis of data pertaining to that phenomenon. This involved data preparation, developing concepts through coding, explanation and building causal networks. Data analysis was supported by a qualitative data analysis software package (ATLAS/ti), which also acted as case study database.

This comprises two principal modes: firstly the textual level which includes activities like segmentation of data files, coding of text, image and audio passages and the writing of memos; and secondly the conceptual level which focuses on model building activities such as linking codes to form semantic networks [15]. The organizational level offers the necessary "infrastructure" for the other two. In the organizational level the materials are prepared, the database is organized, the encoding type of documents are changed, and allows the researcher to migrate complete projects to other computers.

IV. Background to the Organizations

A total of 12 in-depth interviews have been conducted in 7 organizations within the UK. These organizations were selected on the basis of their experience with CBS and willingness to participate in the case study. However a deliberate effort was made to ensure that a wide variety of organizations from different sectors were included in the case study (see table II).

A. Case 1

This organization 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 synchronised way, enabling them to deliver sustainable value.

TABLE II
Background Information about Participant Organizations

Case #	Main business area	Size of organization
1	Software house	Small
2	Research and development	Medium
3	Local authority	Medium
4	Manufacturing/ Engineering	Large
5	Finance/ Banking	Large
6	Consulting services	Small
7	Retail/ Wholesale	Large

B. Case 2

This organization is 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. The applications of COTS components that were investigated in this case study were high integrity systems and multimedia systems.

C. Case 3

This organization provides services to 175,000 people including education, social services, leisure, highways, planning, cleansing and many more. Information Technology & Telecommunications (IT) department provides support for computer systems throughout the authority. For example the IT department advise the other departments on the process to follow when acquiring COTS software and also help in evaluation by allocating IT staff to evaluating teams.

D. Case 4

This organization is a world leading 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 £1billion with over 42,000 employees and customer in 135 countries. The interviews were conducted members of staff in the department responsible for software development and maintenance.

E. Case 5

This organization offers a number of specialist business services including business banking, commercial banking, and corporate and institutional banking. The IT department has over 1500 software developers dedicated to developing and maintaining software systems. The interviews focussed on the business architectural group with the IT department. The respondents indicated that they are using standard object oriented concept modelling techniques using UML notations. This is supported by the Rational Rose CASE tool.

F. Case 6

This organization, although small, is one of the market leaders in management and financial services consultancy. The management services specialise 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, USA, Ireland, Malaysia, Spain, and South Africa.

G. Case 7

This organization is a leading retailer with over 370 stores in the United Kingdom, Europe, Hong Kong and Canada. It employs 68,000 people around the world and achieved a group turnover of £8billion. Although outside contractors do most of the software development the organization still maintains a pool of its own analysts and developers. Interviews were based on experiences and lessons learnt from the stores system project (SSP) and Contracts Management System (CMS).

V. Results and Discussion

A. Requirements and component specification

In order to select the component from the marketplace the business requirements need to be clearly elicited, business architecture developed. Once the business component has been specified then the software component that matches this business component can either be purchased or developed. The respondents brought out a number of factors that are important during this process of business component specification and these include issues concerning business architecture, Object Oriented modelling, Use-cases and scenarios, Unified Modelling language (UML), and Joint Application Development (JAD) sessions.

B. Social-technical criteria definition

1) Compliance (functionality) issues: The respondents in all the cases indicated that a selected COTS component must meet some kind of high level requirements. This is consistent with findings in literature [5][6]. Two cases emphasized the importance of open system in selecting COTS software for CBS. These respondents were not in favour of selecting any COTS software that uses propriety standards or formats. In other words the product itself has to have certain attributes which should be considered to be best practice, for example, its data or information should be accessible with other available commercial tools.

Compliance to international standards or standards set by organizations is an important criteria especially in high integrity systems. Six out of seven cases indicated that evaluators of COTS software should consider including customer and organizational standards in the selection criteria. The respondents from three cases acknowledged the importance of organizational policies in COTS software selection. Organizations sometimes set up software policies and standards to which the selected software must adhere to, for example, an organization may standardize on Microsoft software products. The inclusion of this criterion helps to screen the available software products and reduce the number of candidate products for further detailed evaluation. However some organizational policies can be prohibitive and may exclude "better" COTS software from the marketplace.

2) Product quality characteristics: The product itself has to have certain attributes which are considered to be the best practice for it to be selected, for example, it has to have the ability to plug and play with other components and adaptable [16]. Four out of seven cases brought out the importance of interoperability when selecting COTS products. Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged. Some of the respondents emphasized the importance of selecting products that can compatible with (interoperate) Microsoft products. The importance of the scalability in COTS selection was also brought out by two cases, that is the ease with which a system or component can be modified to fit a different problem area.

Only two cases brought out the importance of including maintainability in COTS selection criteria. Maintainability is the ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment. Most of the cases explained that maintainability should be left to the COTS provider and therefore it is not seen as important factor that should be included in the selection criteria.

The respondents emphasized the importance of selecting products that are reliable and dependable. Reliability is the ability of a system or component to perform its required functions under stated conditions for a specified period of time. The most frequently discussed and quoted product characteristic for the high integrity systems was reliability (case 2 and case 4). The respondents stressed the importance of selecting products with good graphical user interface (GUI) and suggested that the GUI can assist in assessing the usability. The user interface of the COTS product must be friendly and make it easy for the user to learn and operate the product.

TABLE III
Identified Product Quality Characteristics (X indicates data matched)

Identified factors	Case #						
	1	2	3	4	5	6	7
Interoperability	X		X	X	X		X
Portability	X			X			
Reusability	X						
Scalability				X			X
Maintainability		X					X
Performance		X		X		X	X
Dependability		X		X			
Efficiency	X			X		X	
Reliability		X	X	X			X
Robustness							X
Usability	X			X	X		X

3) Social-economic (non-technical) factors: Five cases out of seven brought out the importance of cost issues when selecting COTS software products. The general costs include the cost of adapting and integrating the COTS, maintenance (upgrades) costs, training and support costs. The cost of adapting and integrating the COTS is the cost of making changes to the COTS software so that it meets the requirements set for the system. According to cases 2,3 and 4 licensing arrangements and product costs are important factors and should be included in the COTS selection criteria. Licensing and product costs is the cost involved in obtaining an adequate number of licenses for software development and may include delivery (run-time) version costs, that is the possible costs of obtaining the right to deliver the OTS software as a part of the software to users.

Case 1 and 7 indicated that product reputation in the marketplace is a significant factor in the selection of COTS components, for example if a product has a large user base or has been used successfully in a certain domain compared to another it will have a competitive advantage. However in Case 2, one of the respondents was hesitant with this kind of approach and argued that it does not necessary follow that if a product is good enough in civil with safety integrity then it can also be reliable in military. The argument is that you cannot generalize and select the product simply because you can use it in one system.

According to the respondent what the market thinks about the product determines whether it is selected or not. It was interesting to note that the respondents considered market trends to be significant factors in the selection of COTS components. For example if a product has a large user base or has been used successfully in a certain domain compared to another it will have a competitive advantage.

TABLE IV
IDENTIFIED NON-TECHNICAL FACTORS (X INDICATES DATA MATCHED)

Identified factors	Case #						
	1	2	3	4	5	6	7
Business issues							
Contractual issues	X	X	X				
Costs issues		X	X	X		X	X
Escrow or buy rights					X	X	X
Licensing issues		X	X	X			
Marketplace variables							
Marketplace changes					X		
Delivery period						X	
Market leaders							X
Market trends				X	X		X
Product reputation	X						X
Vendor issues							
Vendor capability			X	X	X	X	X
Training & support						X	X
Vendor reputation	X			X		X	X

It is interesting to note that the respondent considered the vendor reputation and capability to be significant factors in the selection of COTS components. It is important that COTS software is acquired from a vendor who has financial viability in order to benefit from future releases. The respondents' indication that they normally deal with people they can trust and with a good company profile shows that vendor capability and stability is an issue when selecting software components. Although the respondents considered vendor capability and stability to be significant factors in the selection of COTS components, they also indicated that they could always buy rights to the software or buy the company if they were dependent on that software. Respondent in cases 6 and 7 brought out the importance of the availability of training and local support when selecting COTS software.

4) Technology factors: The respondents, recognizing the importance of technology factors, indicated that COTS component selection should not be based on obsolete technology. Case 3, 5 and 7 brought out the importance of assessing the COTS underlying technology standards and protocols. Furthermore Case 3 indicated that COTS products should support plug and play.

Case 2, 4 and 7 indicated that they have started to look at technology that support integrability for example interface definition and implementation. Two such technologies that support integrability that they are investigating are CORBA and DCOM. It is important at an early stage to agree on the underlying technology and the architecture before embarking on selecting the COTS components.

The other important technology issues identified in this case study include multi-user systems, distributed technology, security, architectural styles and framework.

C. Search for alternatives

Some of the cases brought out the problem of finding COTS components in the marketplace. This problem could either be due non-availability of COTS components in the marketplace or poor documentation for the few available components. The availability of COTS software in the marketplace has a significant impact on the selection process, for example in some organization the tender procedures indicate that you must have at least three products to choose from.

The respondents emphasized that user community is an important source of information on COTS software components. The respondents emphasized that the web sites (and repositories) of the big vendors are important sources of information on available COTS software components. Internet search is one of the techniques brought out by all the cases as important technique for identifying COTS components from the marketplace.

search

TABLE V
IDENTIFIED SEARCH TECHNIQUES (X INDICATES DATA MATCHED)

Identified factors	Case #						
	1	2	3	4	5	6	7
Past experience						X	
Networking						X	X
Mailing list						X	X
COTS repository							X
Internet search	X	X	X	X		X	X
Market surveys							X
Request for proposals					X	X	
User community			X	X	X	X	X
Vendor promotions			X	X		X	X
Vendor publications			X	X		X	X

It is interesting to note that the respondents brought out in this case study the use of public tender procedures as a possible technique for selecting software components. The customer can request for proposals from the vendors regarding some software component and make selection based on the submissions.

D. Assessment (evaluation)

1) Evaluation strategy: The lack of an explicit evaluation strategy in cases 3, 4 and 5 indicates that the evaluation is most likely conducted in an ad-hoc manner. For example case 3 indicated that organization policy for COTS software selection should be based on the PRINCE method, however in practice it is not. The internal guidelines document for Case 4 recommends progressive filtering strategy for COTS software selection but the respondents again indicated that in practice the evaluation was non-systematic.

However to draw maximum benefit from the COTS selection, companies should conduct the COTS selection process in a systematic manner. In literature there are three evaluation strategies proposed and these are progressive filtering, keystone selection strategy and puzzle assembly model [17]. Case 6 indicated that they followed progressive filtering whereby they would start with a candidate set of components, progressively more discriminating evaluation mechanisms are applied in order to eliminate less "fit" components. Case 7 were not explicit on the evaluation strategy because the projects in which the respondents were currently working has initially focussed on developing the component from in-house rather than selecting them from the marketplace.

2) Data collection techniques: Four out of seven cases indicated that experimentation and user community proved to be valuable sources of attribute data about COTS software components. In order to elicit such data it may be necessary to subscribe to some mailing lists or keep a directory of relevant contacts phone numbers. It was argued that if the product has been working well with other users it should work well for the new situation.

The case study also brought out studying vendor literature as an important technique (Case 1, 4 and 6). Other techniques identified in this case study for collecting attribute data about the product include auditing the development process, qualification of COTS, software tests and the use templates and questionnaires. The problem of how to evaluate COTS software since they are "black boxes" was brought out by all the cases and it was suggested that experimentation and pilot studies could assist in this regard. During experimentation, for example, the evaluator could artificially introduce a fault in the COTS product and see how the system will respond to this.

3) Data analysis techniques: Two cases brought out the weighted sum method (WSM) technique for data analysis and consolidation of attribute data. The problem of WSM has been discussed in literature and the Analytic Hierarchy Process (AHP) has been recommended as a better method for consolidating evaluation data [5][18]. AHP is a multi-criteria decision technique based on pair-wise comparison between the alternatives. The result of this pair-wise comparison is converted to a normalised ranking by calculating the eigenvector from the comparison matrix's largest eigenvalue. The advantages of the AHP technique are that it provides a systematic approach for consolidating information; an objective weighing technique for setting the weighing scale for qualitative and quantitative data, and allows for consistency checking [5][18].

However all the cases investigated did not bring out the importance of the AHP or any multi-criteria decision techniques for consolidating attribute data.

E. Organizational issues

In literature organizational considerations are said to be important in software development, and the case study brought out this observation. For example user participation in the software process can give a sense of user ownership [19]. Four out of seven cases brought out the importance of customer motivation and systems ownership. They argued that it is difficult for example to convince developers who are used to structured programming to migrate to CBS, because there is a threat to professionalism. Educating stakeholders on the CBS benefits and training of developers in CBS can help.

The respondents indicated that customer participation is an important factor during COTS selection and this leads to customer ownership and motivation. Case 3 supported the idea brought out by Case 1 on the importance of maintaining a relationship with the customers throughout the development process and eliciting support from key sponsors and stakeholders. In other words if users are not involved this leads to user resistance because the user does not identify with the software system. However one of the respondent in case 2 argued that customer participation is not a major factor during COTS selection.

TABLE VI
IDENTIFIED ORGANISATIONAL FACTORS (X INDICATES DATA MATCHED)

Identified factors	Case #						
	1	2	3	4	5	6	7
Risks		X		X			X
Cost justification				X			
Customer expectations					X		
Customer experience	X				X		
Ownership & motivation				X	X	X	X
Customer participation		X	X	X	X	X	X
Customer resistance				X	X		
Incentives					X		
Management support				X	X		
Management structure					X		
Management resistance				X	X		
Organizational culture				X	X		
Organizational politics				X	X	X	X
People's attitude					X		
Skills profile					X		X
Team communication					X		X

The way incentives are given to developers can discourage reuse and CBS because some developers are paid according to the number of lines of code they write [20]. In addition, managers are paid based on annual outputs while CBS may take some time before the benefits are realised. The respondents highlighted this in case 2. Changing the policy on incentives and bonus in organizations migrating to CBS can help in dealing with this problem. Case 1 and case 5 indicated that customer and developer's experience and education skills are important and can affect the software system. Case 5 argued the skills-set in their organization is currently based on structured methodologies rather than CBS and developers do not have sufficient CBS techniques. Therefore identification and provision of appropriate CBS training is an important activity.

Political issues are organizational processes or principles affecting power, authority, and status. Some people within organizations are more powerful than others, by virtue of their positions while others by their connections with powerful people within an organization. Four cases (case 4, 5, 6 and 7) indicated that they have experienced political interference within the organization as they have been trying to procure COTS software. Recognition of power differences and their causes can aid in the design and development of information systems that support the organization, its functions and individuals within it.

The respondents indicated that a strong business case or cost justification must be presented in order to secure management support to allocate resources to the project. Therefore management support is critical to the success of any software system. Educating management using incremental approach and successful case studies can assist in eliciting management support. Organizational setting and management style, that is the arrangement of organizational subsystems and the accompanying division of labour and hierarchy of authority relations, can inhibit CBS success [20]. This was brought out by case 5.

Customer, developer and management resistance to change in most cases is due to inherent organizational culture. This was brought by case 4 and supported by case 5. For example the respondents in case 4 indicated that there are certain issues in which they would not be willing to change the way they work. Long term education and training can contribute to changing people's attitude and organizational culture.

VI. CONCLUSION AND LESSONS LEARNT

The major outcome of this case study was that the current industrial practices for COTS evaluation and selection were elicited. A number of processes (both traditional and soft factors) that support COTS software components selection were identified and classified (see tables III-VI). The social-technical approach advocated by STACE method proved to be a helpful framework for identifying and classifying the important factors in the COTS software processes.

Using ATLAS/ti software, noteworthy segments relevant to one idea were retrieved and compared to one another until patterns began emerging. The outcome of this theory-building or creative conceptualization led to the learning of a number lessons. The most important lessons learnt through this case study include:

1. It is important to *develop and agree on the requirements definition and component architecture before embarking on the COTS component selection* process. Selecting the underlying technology for interoperability and testing the COTS product against the adopted technology will assist the organization in selecting the best technical product.
2. *"Word of mouth" from user community is an invaluable tool for identifying COTS components in the marketplace.* Therefore maintaining good contacts with the user community through professional bodies and mailing lists can prove useful and helpful. However this could be complimented by other techniques such Internet search, market surveys, vendor promotions and publications.
3. The *best way of evaluating COTS products is through experimentation within the operating environment* in which the product will be used. This is because lack of access to the COTS internals makes it difficult to understand COTS components and therefore makes evaluation hard.
4. *Procuring and using COTS products is about managing risk* and therefore organizations should be willing to invest in CBS in order to realize benefits. This is because the use of COTS presents security vulnerabilities [21]. For example COTS qualification for safety-critical systems is very difficult and expensive to implement.
5. Remember to *include the non-technical factors in the selection criteria.* Selection of COTS components is not only based on technical factors such as functionality but also on other non-technical attributes.

6. *Consider organizational issues when selecting COTS products.* Organizational issues are important factors in the COTS selection process; for example presenting a strong business case will attract management support. However there are other organizational issues that should be considered such as customer resistance and organizational politics. Customer participation in the selection process will encourage system ownership and motivation.

The identified factors and lessons learnt from the case study assisted in elaborating and initial validation of the STACE framework. The framework needs further validation; therefore future research will involve evaluation of the STACE framework by the industry in terms of appropriateness of use and usefulness. The framework would incorporate the often-neglected non-technical issues and provide guidance for organization regarding the process of evaluating and selecting COTS software components for CBS. Furthermore the framework will provide a classification of techniques, tools and support for COTS software component evaluation phase of CBS. This will contribute to reducing risks associated with CBS. Further work can also investigate quantitative relationships between the factors and the cases, or draw conclusions about COTS component selection in different industries.

References

[1] Oberndorf P., "Facilitating Component-Based software Engineering: COTS and Open Systems," in *Proceedings of the Fifth International Symposium on Assessment of Software Tools*, IEEE Computer Society, Los Alamitos, California, June 1997.

[2] Haines Capt Gary, Carney David and Foreman John, Component-Based Software Development/ COTS Integration, Software Technology Review, Available WWW (online) <URL: http://www.sei.cmu.edu/ str/descriptions/cbsd_body.html>, 1997.

[3] Brown Alan W. and Wallnau Kurt C., "A Framework for Systematic Evaluation of Software Technologies," *IEEE Software*, September 1996.

[4] Kunda Douglas and Brooks Laurence, "Applying social-technical approach for COTS selection," *Proceedings of 4th UKAIS Conference*, University of York, McGraw Hill. April 1999.

[5] Kontio Jyrki, "A Case Study in Applying a Systematic Method for COTS Selection," *Proceedings of the 18th ICSE*, IEEE Computer Society, 1996.

[6] Maiden Neil A. and Ncube Cornelius, "Acquiring COTS Software Selection Requirements," *IEEE Software*, pp. 46-56, March/April 1998.

[7] Friedman Batya and Kahn, Jr. Peter H., "Educating Computer Scientists: Linking the social and technical," *Communication of the ACM*, vol. 37, No.1, pp. 65-70, January 1994

[8] Avison D. E and Fitzgerald G., *Information Systems Development: Methods, techniques and tools*, McGraw-Hill Book Company, London, 1995.

[9] Trist E.L., "On Socio-Technical Systems" in *Sociotechnical systems: A sourcebook*, Pasmore W. A and Sherwood J. J., Eds. La Jolla: University Associates Inc., 1978.

[10] Cherns Albert, "Principles of socio-technical design" in The Social Engagement of Social Science Volume 2, Trist Eric and Murray Hugh, Eds. University of Pennsylvania Press, 1993.

[11] Banner David K. and Gagne T. Elaine, *Designing Effective Organizations: Traditional and Transformational views*, Sage Publications, London, 1995.

[12] Jirotka Marina and Goguen Joseph A., Eds. *Requirements Engineering: social and technical issues*, Academic Press Limited, London, 1994.

[13] Yin Robert K., *Case study Research: Design and Methods*, Sage Publications, London, 1994.

[14] Strauss Anselm and Corbin Juliet, *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*, Sage Publications, London, 1990.

[15] Muhr Thomas, *ATLAS/ti - Visual Qualitative Data Analysis Management Model Building: User's Manual and Reference*, Scientific Software Development, Berlin, 1997.

[16] Powell Antony, Vickers Andrew, Lam Wing and Williams Eddie, "Evaluating Tools to Support Component Based Software Engineering," *Proceeding of the 5th International Symposium on assessment of software tools*, IEEE Computer Society, 1997

[17] Oberndorf Patricia A. and Brownsword Lisa and Morris Eds, *Workshop on COTS-Based Systems*, Software Engineering Institute, Carnegie Mellon University, Special Report CMU/SEI-97-SR-019, November 1997.

[18] Zviran Moshe, "A comprehensive Methodology for Computer Family Selection," *Systems Software Journal*, vol. 22, pp. 17-26, 1993

[19] Emam Khaled El and Quintin Soizic and Madhavji Nazim H., "User participation in the Requirements Engineering Process: An Empirical Study" in *Requirements Engineering*, Springer-Verlag London Limited, London, pp. 4-26, 1996

[20] Lynex A. and Layzell P. J., "Understanding Resistance to Software Reuse," in *Proceedings of 1997 Software Technology and Engineering practice (STEP97)*, IEEE computer society, 1997.

[21] Lindqvist Ulf and Jonsson Erland, "A Map of Security risks Associated with Using COTS," *IEEE Computer*, Vol. 31, No. 6, pp. 60-66, June 1998.