International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

COMPONENT-BASED SOFTWARE ENGINEERING: THE BEST HOPE FOR DEVELOPING COUNTRIES

Douglas Kunda, dkunda@mu.ac.zm, dkunda@mu.edu.zm
Centre for ICT Education, Mulungushi University, Zambia

Abstract

Although there have been many advances in the information technology (IT) field, most developing countries (DCs) have not yet fully benefited because of some of the specific problems experienced by these countries. Examples include a lack of systems infrastructure and resources to invest in IT. Component-Based Software Engineering (CBSE) offers a number of benefits that the DCs can tap into such as reducing development and maintenance costs and improving reuse across projects. This paper will discuss the findings of a survey conducted in a developing country and the UK to elicit current CBSE practices and experiences. The results show that although CBSE is in its infancy in DCs, it has a lot of potential because of a number of benefits and institutions familiarity with COTS products. The paper recommends adapting the best CBSE practices from developed countries and applying them to DCs.

Keywords: Commercial Off-The-Shelf (COTS), Component-based software engineering, COTS evaluation, software engineering, information systems

1. Introduction

Many information systems in developing countries can be categorized as failing either totally or partially [1]. A range of factors, including infrastructural, financial, political and cultural aspects have acted against the effective development and exploitation of information technology. Redressing this situation will require significant resources and a willingness to tackle the long-term underlying causes of the problem rather than offering short-term solutions. Institutions in DCs are turning to Commercial Off-The-Shelf (COTS) software components, because modern information systems are becoming increasingly expensive to build and maintain. These components are typically bought from third-party vendors and integrated into a system [2]. Alternatively, an application that satisfies most of the system requirements could be purchased, and then extended, tailored for local requirements. In both cases neither is the source code for these components available to the system developer nor do the system developer control specification, release schedule and evolution of the components.

Building systems from COTS, also known as Component-Based Systems Engineering (CBSE), is focused on improving the technologies and practices used for assembling previously existing components (COTS and other non-developmental items) into large software systems, and migrating existing systems towards CBSE approaches [3][4]. The CBSE approach changes the focus of software engineering from one of system specification and construction to one of component: identification, qualification, adaptation, integration and upgrade (for system evolution). The CBSE approach relies on the existence of an inventory of existing software components, the emergence of component integration technologies such as Common Object Request Broker Architecture (CORBA) and Component Object Model (COM), and the development of organizational capabilities for CBSE trade-off analysis and design. Growing capabilities in each of these areas is encouraging the migration towards CBSE in a broad range of domains.

CBSE 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 [4][5]. CBSE also improves reuse across programs and promotes a competitive component marketplace. CBSE therefore has a higher potential to benefit DCs compared to other systems development methods.

This paper presents the results of a survey aimed at eliciting and synthesizing current CBSE practices from both a developing country and the UK. The survey involved administration of self-completion questionnaires and data analysis using SPSS (Statistical Package for Social Scientists).

ISSN:2319-7900

2. Background

2.1 Problems of developing software systems in DCs

Developed nations have used IT to help them change the way they do business so as to give them a strategic advantage in their operations (e.g. the use of ATMs in banks to improve customer service) [6]. However the investment returns in DCs have fallen short of the potential due to problems unique to them. Mohan [6] attributes these to (a) inappropriate choice of applications, and (b) inadequate consideration of the organisational and environmental factors in the formulation and implementation of the IT strategy. Manufacturers and vendors of IT and IS tend to focus on technical issues for solving the information problems of DC. In reality the most urgent problems are not technical, they deal with the management of this technology.

W'O Okot-uma [7] argues that the problems in introducing IT in DCs can be classified into three generic problems namely contextual, strategic and operational. Contextual problems are due to weak fit between models of developed countries design and applications in the DCs' context, semantic discrepancies in the wording and understanding of phenomena as well as references to different value systems and different concepts of rationality. Strategic problems are those relating to local, national and regional policy initiatives, as reflected in the institutional intervention mechanisms of influence, regulation and implementation. Operational problems are faced by DCs due to technical and economic constraints and lack of skilled personnel.

Although identification of problems faced by DCs is a very complex task, the concentration should be on those problems that are unique to the DCs and which may have a significant impact on the assimilation of the IT. This paper highlights the problems of systems infrastructure deficiency, economic constraints, skilled human resources deficiency and applications problems.

Systems infrastructure deficiency

Successful implementation of computer systems is dependent upon there being a systems infrastructure on which to build. For example in the UK, the taxation system works comparatively efficiently because it has been built up over years and because individuals, companies, banks, accountants and so on, accept the procedures [8]. In DCs this is not the case. In most DCs [7] the electrical power utility has been intermittent and the inconveniences caused have not been negligible. DCs also do not have adequate telecommunication infrastructure and use voice telecommunication lines for data communication, which have not proved useful.

Economic constraints

Economic constraints is another set of major obstacles restricting the application of IT in DCs. These includes the non-existence of reliable background statistical information and inadequate capital to finance IT [7]. Several DCs suffer from both a lack of resources and a limited domestic market [9]. These countries import IT because they lack an indigenous IT industry. Scarcity of foreign currency also forces them to depend on donor agencies for much of their IT imports.

Skilled human resources deficiency

The lack of skilled human resources is certainly the principal barrier blocking the diffusion and efficient/effective exploitation of IT systems in DCs [1][8][10]. There is clearly a problem of quantity and quality (mismatch between needs of industry and trained personnel) of IT personnel [9]. A number of factors have been isolated as prime causes of these deficiencies such as evolution of technology, high turnover of skilled staff due to poor conditions of service, lack of counterpart training under technical assistance.

Applications problems

The priority areas of application of computer systems in DCs are different from developed countries [8]. This has implications for the transferability of software. For example DCs have the following problems and tasks:

- Development and exploitation of natural resources,
- Raising educational standards of the population,
- Raising the standards of health, and
- Increasing food production

Bhatnagar [9] argues that the factor contributing to the low impact and penetration of computers in these economies is the type of use that such computers have been put to. Most DCs have used their computers for routine transaction processing tasks rather than strategic information systems. There is a *model mismatch* meaning that wrong choices of technology are often made or essential elements of technology transfer such as training are not implemented.

ISSN:2319-7900

2.2 Potential benefits of CBSE

Building systems from components, that are available commercially, offers the opportunity to lower costs by sharing them with other users, thus amortizing them over a larger population, while taking advantage of the investments that industry is putting into the development of new technologies [11]. Most institutions typically spend far too much effort on defining to the lowest level of detail the desired characteristics of systems and how the contractors are to build those systems to achieve those characteristics. Thus a lot of resources are expended developing systems and components that often already exist- or exist in "good enough" form with nearly the same capabilities- elsewhere.

CBSE can potentially be used to reduce software development costs, assemble systems rapidly, and reduce the spiraling maintenance burden associated with support and upgrade of large systems [4][5]. Other potential benefits of building systems from COTS include cost benefit obsolescence management, improving reuse across programs and promoting competitive marketplace enabling system integrators a wide range of choices.

To date, the commercial components available and reliable enough for operational systems, and whose interfaces are well-enough understood, have primarily been operating systems, databases, email and messaging systems, office automation software (e.g., calendars, word processors, spreadsheets), and Graphical User Interface (GUI) builders [12][13]. The list is made up of applications and components that are mature and pervasive in a large number of systems. This maturity is a likely reason why they have been successfully marketed as COTS software components. The number of available components continues to grow and quality and applicability continue to improve. However, in spite of the possible savings, using COTS components to build safety-critical systems where reliability, availability, predictability, and security are essential is frequently too risky [3][12].

In addition to the increasing availability of components applicable to specific domains, understanding of the issues and technologies required to expand CBSE practice is also growing, although significant work remains. Various new technical developments and products, including CORBA, COM, DCOM, and related capabilities [13] and changes in acquisition and business practices has further stimulated the move to CBSE.

2.3 Risks associated with CBSE

It is widely assumed that the component-based software development approach, particularly in the sense of using COTS components, will be significantly less costly (i.e. shorter development cycles and lower development costs) than the traditional method of building systems "from scratch." In the case of using such components as databases and operating systems, this is almost certainly true [12]. However, there is little data available concerning the relative costs of using the component-based approach and there are a number of new issues that must be considered.

Furthermore, integrating COTS components requires additional system development and maintenance cost such as negotiating, managing, and tracking licenses to ensure uninterrupted operation of the system. For example, a license expiring in the middle of a mission might have disastrous consequences. The following is a summary of some risks associated with COTS systems [4][12][13][14]:

- Difficult to discover the actual technical capabilities of the COTS software components;
- Instability due to periodic releases of COTS software;
- Legal implications in case of system failure;
- Difficult to identify and resolve product incompatibilities (mismatches);
- Additional functionality causing side effects;
- Additional qualification certification tasks;
- Failure to meet requirements;
- Belated discovery of inadequate product reliability and undesired feature interaction; and
- Lack of support if COTS provider goes out of business.

3. Empirical study

3.1 Goal, Objectives and Method

The overall goal of the study was to elicit and synthesize current practices and potential benefits of CBSE in developing country. The outcome of this survey was to document and validate whether CBSE is the best hope for developing countries. The following are immediate objective of the survey:

- obtain a thorough understanding of the information systems development process using COTS;
- identify current practices for building systems using COTS software;
- review current benefits, costs and risks associated with COTS software reuse;

ISSN:2319-7900

- learn what kind of problems (and solutions) people have experienced from the past in relation to building systems from COTS software; and
- review current tools and techniques used for building systems from COTS software;

Data was collected through the administration of self-completion questionnaires to a sample of institutions. Questionnaires were used because they provide access to geographically dispersed samples at low cost i.e. a large population can be surveyed relatively cheaply [15]. In addition questionnaires provide a high degree of anonymity and respondents have time to think about their answers and consult other sources. The questionnaire was organized under three main themes, 1) the overview of development process, 2) requirements engineering phase, and 3) approaches to CBSE.

The chosen developing country for this empirical study was Zambia. This country was chosen for the empirical study because most of the problems experienced by Zambia match the problems faced by other developing countries in general [1][17]. In Zambia, for example, the communication infrastructure is poor making telephone calls expensive and beyond the reach of the common person. In addition, Zambia was chosen because of accessibility and cost of conducting the empirical study. Therefore the findings and conclusion from this study can prove useful and applicable to developing countries.

The sources of data are personnel within institutions responsible for specifying, procuring and developing software systems. Zambian institutions were obtained by purposively sampling from the Zambia National Directory of Companies for those that have experience in building software systems. Most companies in Zambia develop their software system in-house because of the small number and unreliability of software development houses. The same questionnaire was administered to UK software houses to elicit good practices for comparison purposes. The UK sample was generated by searching the Kompass database for software houses and systematically sampled to reduce the number to the required sample.

The data collected were coded and analyzed in SPSS (Statistical Package for Social Science). Frequency distributions were used to categorize the demographic data variables. The mean were calculated to measure the central tendency of the variables, unlike the mode or median it takes into account all the values in the distribution, making it sensitive to extreme values. Variability was measured through standard deviation (*S.D.*) because it is more stable from sample to sample and can be used for two or more combined groups. Finally the coefficient of variation (*CV*) were calculated and used as standard to compare the relative importance of the variables.

$$CV = \frac{(SD)}{\bar{X}}$$

Therefore the adopted data analysis approach for this study is descriptive statistics method. Descriptive statistics method is type of exploratory survey research aimed at describing the distribution of a phenomenon in a population, thereby ascertaining facts [16]. For example descriptive survey might document the types of manufacturing processes being used by small and large manufacturing firms. Hypothesis related to common perceptions or changes over time could be formulated and tested.

3.2 Results

The full report detailing the results of this survey is documented separately [17] while this paper highlights the main findings and conclusions.

Overview of development process

It is interesting to note that 90% of respondents from Zambian institutions indicated that they were using CBSE methods compared to 80% from UK software houses. This indicates the relative popularity of CBSE in DCs. The most common software development process was evolutionary development for Zambian institutions (36.4%), while software houses indicated incremental development (41%).

According to this survey both the UK and Zambian respondents indicated that the main constraints, or obstacles to developing software systems, is lack of adequately trained human resources. This supports the findings already identified in literature [1][8][10]. However, in addition respondents from UK also scored lack of time as a major constraint. However while the literature [18] [19] indicates the importance and impact of political and external issues on the success of the information system, it scored lowly in this survey.

Reducing software development and maintenance costs has been identified as potential benefits of building systems from COTS software [4][5]. Respondents in this survey supported these findings. In addition Zambian respondents highly rated improving reuse across projects as an important benefit of CBSE (see Table 1).

ISSN:2319-7900

Table 1: Important benefits of CBSE (respondents CV)

	UK software houses	*Zambian institutions*
Reduces development cost	**4.11**	**3.92**
Reduces maintenance cost	**3.18**	**3.29**
Improves reuse	**3.52**	**5.84**
Obsolescence management	**3.50**	**2.95**
Competitive market	**3.31**	**3.74**

Zambian institutions highly scored product mismatches and periodic releases of COTS as the major risk of building systems from COTS components. Software houses from UK indicated that the major barrier or risk associated with CBSE is the difficult to discover the technical capabilities of COTS components (see Table 2).

Table 2: Risks associated with CBSE (respondents CV)

	UK software houses	*Zambian institutions*
Lack of guidelines	**2.95**	**1.75**
Technical capability	**4.58**	**2.67**
Periodic releases of COTS	**3.63**	**3.34**
Loss of schedule control	**3.13**	**2.28**
Legal implications	**2.80**	**1.90**
Product mismatches	**3.69**	**3.54**
Side effects	**3.46**	**2.45**
Additional tasks	**3.51**	**1.78**
Failure to meet requirements	**3.96**	**3.27**
Lack of provider support	**4.24**	**2.50**
Difficult to select	**3.53**	**2.85**

The respondents from both Zambia and the UK indicated that the main application of CBSE is office automation, database systems and business applications. This indicates the maturity of these COTS products and confirms what has been identified in literature [12][13]. Zambian respondents did not indicate applying CBSE to real-time and embedded systems.

Requirements Engineering Phase

The survey indicated that observation, prototyping and demonstrations are the leading techniques used in requirements acquisition and specifications. Although the literature [20] encourages generation of scenarios and use-cases and matching them to COTS products, the importance of prototyping and demonstration has also been recognized. It is interesting to note that the techniques advocated by social-technical approaches such as rich pictures and SSM conceptual models scored very low in this survey.

Respondents from Zambian institutions indicated user community experience, study of COTS documentation and customer prior knowledge, as the techniques used to evaluate and select COTS components. This suggests that COTS selection could be conducted in an ad-hoc manner by these institutions. On the other hand, the UK software houses indicated attending COTS demonstration, studying documentation and extensive experimentation with COTS products. This agrees with the literature [3][22][23].

Customer participation in software development has been recognized and advocated by social-technical approaches such as ETHICS [24]. The respondents indicated their preference for customer participation in the earlier requirements acquisition compared to later part of the system development such as design stage. Some respondents suggested discussion with customers as the best way of dealing with social issues. However, there was little agreement on which social factors are important and should be considered during CBSE.

Table 3: COTS evaluation techniques (respondents CV)

	UK software houses	*Zambian Institutions*
Study documentation	**4.53**	**2.98**
Attend demonstration	**4.72**	**2.59**
Extensive experimentation	**4.10**	**2.28**
Customer experience	**3.17**	**2.88**
User community experience	**4.00**	**3.05**
Cards sorting and laddering	**2.13**	**2.12**
Feature analysis	**2.08**	**1.58**
Multi-criteria decision making	**1.65**	**1.59**
Outranking	**1.79**	**2.14**
Analytic Hierarchy Process	**1.84**	**1.63**

Approaches to building systems from COTS

Three primary approaches to building systems from COTS component have been identified by literature [12][21] and these are: 1) the buy and use; 2) buy and adapt; and 3) the component integration. This survey confirmed these findings. In the buy and use approach a single complete working COTS software system that satisfies most of the user requirements is purchased and used without adapting or extending it. For example purchasing and using a standard word processing package. This approach can be considered

ISSN:2319-7900

to be the beginner's approach.

The buy-and-adapt model is characterized by acquiring a single complete working system that satisfies most of the requirements of the acquisition agency and adapting or extending it for local needs. The component integration model involves purchasing a number of off-the-shelf components each satisfying some part of the requirements of the system and integrating these components into the required system. This model usually depends on the use of some "gluing technology", which may be unrelated to the components, to provide an interface between components [12]. This approach can be considered to be the ultimate stage.

Table 4: CBSE approaches (respondents %)

Approaches	*Software Houses (UK)*	*Zambian Institutions*
Buy and use	**37.5**	**69.2**
Buy and adapt	**31.2**	**76.9**
Integrate COTS	**54.4**	**61.5**

It was interesting to note that the main programming language or development tool by the Zambian respondents was CA-Clipper and Visual Basic. The UK Software houses scored Visual Basic and C/C++ as the main programming platforms used to adapt COTS. Visual Basic language can be easily used to tailor most of the COTS components sold by Microsoft, such as Office. CA-Clipper is a general purpose, high-level programming language well suited to corporate and commercial applications development. Zambian institutions would greatly benefit from CBSE if they would strengthen capacities in C/C++ or other tools used by developed countries like the UK.

According to the literature, adaptation of COTS components is achieved by extending it through add-ons, interfacing with other applications, or modifying the off-the-shelf application through source code changes [12]. Zambian institutions indicated the use of the component application-programming interface (APIs) and source code to adapt COTS components. However, once a modification to the source code is done the acquisition agency no longer has an off-the-shelf component and may not benefit from vendor support during the normal course of upgrading the product.

Although the literature suggests that CORBA is an important standard for COTS component integration, respondents from this survey indicated OLE and DDE as the most significant technologies for integration (see Table 5). OLE and DDE are both from Microsoft Corporation who have a monopoly on the PC operating system.

4. Lessons learnt from survey

Although the survey sample was not large enough to draw statistically valid generalisations, from the analysis of the coefficient of variation, nevertheless, the exercise proved to be useful and taught us a few lessons. These lessons as well as other measures not reported here guided us in the development of the STACE framework [25].

Table 5: Technologies for COTS integration (respondents %)

	Software houses (UK)	*Zambian institutions*
COM	**22.8**	**15.4**
DCOM	**10.5**	**0**
OLE	**36.8**	**30.8**
DDE	**29.8**	**15.4**
ActiveX	**17.5**	**7.7**
CORBA	**8.8**	**0**
OpenDOC	**1.8**	**0**
OSA	**3.5**	**7.7**
RMI	**3.5**	**0**

Lesson 1: *Developing countries can benefit from building systems from COTS components by reducing the software development costs.*

Building systems from COTS components is cheaper because the essential requirements need not be specified in detail (as with bespoke systems) and the cost of COTS component is shared among a number of users. However building systems from COTS has it own problems. The most significant of which, identified in this survey are the lack of access to technical information, COTS product mismatches, lack of support from COTS provider and failure to meet all customer requirements.

Lesson 2: *CBSE can contribute to enhancing human resource capacities in developing countries.*

The lack of adequately trained human resources was identified as the main obstacle to developing information systems in Zambia. The familiarity of Zambian institutions with COTS components and other CBSE benefits possibly will enable CBSE contribute to improving their technical capacities.

ISSN:2319-7900

Lesson 3: *Developing countries can learn a lot from developed countries regarding CBSE.*

For example the survey shows that there are fewer Zambian institutions using component integration approach, compared with those purchasing COTS components and adapting them to their local needs (see Table 4). On the other hand the UK companies indicated more institutions using component integration approach.

Lesson 4: *Customer participation during requirements engineering and COTS component selection can contribute to systems acceptance and ownership.*

Although the survey shows that customer participation is more important during requirements acquisition than during COTS selection and systems design, it appears that in practice these processes are iterative and parallel.

Lesson 5: *Customers in developing countries want to visualise and experiment with products before acquiring the products.*

Therefore, there is much potential for using rich pictures and use-cases. These facilitate communication between the developer and the customer. For example, in this survey, the most notable technique for requirement acquisition and specification is observation, prototyping and demonstrations.

Lesson 6: *COTS component selection in developing countries is still problematic.*

Institutions depend on prior knowledge and experience to select COTS components. In some cases donors impose products from their countries on developing countries. However a systematic and repeatable process incorporating the social issues in evaluation and selection of COTS component can prove advantageous.

Lesson 7: *Few institutions in developing countries were familiar with CORBA and Enterprise JavaBeans technology.*

Contrary to the majority of articles about the importance of CORBA technology in COTS integration the survey shows the most significant technology used by practitioners is Microsoft's OLE and DDE.

Lesson 8: *Databases systems are some of the most mature and well-developed COTS products in developing countries.*

The main application for the CBSE approaches was database systems and this indicates the maturity of these database systems as COTS products.

5. Conclusion

Although there are some risks associated with CBSE, it has great potential in the DCs. Not only because of potential cost saving, but also because DCs are very familiar with COTS components, at least in terms of use and adapting them to local needs. DCs should use CBSE to its full potential by integrating COTS components in an engineering manner rather than simply adapting them.

In order to promote and develop CBSE in DCs there is a need to strengthen and improve support for the requirements acquisition, COTS software identification, qualification, adaptation, integration or assembly and upgrade (for system evolution). This can be achieved by eliciting CBSE best practices and experiences from developed countries. There would also be a need to develop tools to support the use of these "best practices". For example the systematic approach regarding COTS component evaluation and selection would prove invaluable to developing countries.

The methods and techniques from developed country are most often inappropriate to DCs because they generally do not take account of social-cultural contexts of DCs. For example the techniques advocated in literature [22][23] are too complex and laborious for DCs to adopt them. It would be more helpful to develop effective but simple, quick and inexpensive techniques for evaluating COTS components applicable to developing countries. In addition techniques supporting CBSE in DCs should achieve the following measures of success [26]:

- *High levels of system use*, as measured by polling users, employing questionnaires or monitoring parameters such as the volume of on-line transactions.
- *User satisfaction*, as measured by questionnaires or interviews. This may include users' opinions on the accuracy, timeliness and relevance of information, on the quality of service and perhaps on the schedule of operations.
- *Favourable attitudes* of users, analyst and developers about the software systems.
- *Achieved objectives*, the extent to which the system meets its specified goals, as reflected by the quality of decision making resulting from use of the system.
- *Financial payoff* to the organisation, either by reducing costs or by increasing sales or profits.

The importance of the social dimension in IT application has been recognised in both developed [18] and DCs [6]. For

ISSN:2319-7900

example studies in more developed countries show that the major causes of software failures are social rather than technical issues [19]. This suggests that CBSE with social-technical input has greater potential to succeed in developing countries.

6. References

[1] Heeks, Richard, "Information Systems and Developing Countries: Failure, Success, and Local Improvisations." Information Society: An International Journal, Vol. 18, 101–112, 2002

[2] Comella-Dorda, S., Dean, J.C., Morris, E., Oberndorf, P., "A Process for COTS Software Product Evaluation, in: Dean, J., Gravel, A. (Eds.), COTS-Based Software Systems," Lecture Notes in Computer Science. Springer Berlin Heidelberg, 86–96, 2002.

[3] Brownsword Lisa, Tricia Oberndorf, Carol A. Sledge, "Developing New Processes for COTS-Based Systems," IEEE Software, vol. 17, no. 4, 48-55, July-Aug. 2000.

[4] Clements Paul C., "From Subroutines to Subsystems: Component-Based Software Development, American Programmer," Vol 8 No. 11, Cutter Information Corp., November 1995.

[5] Baker Thomas G., "Lessons Learned Integrating COTS into Systems, in: Dean, J., Gravel, A. (Eds.), COTS-Based Software Systems," Lecture Notes in Computer Science. Springer Berlin Heidelberg, 21–30, 2002.

[6] Mohan Lakshmi and Belardo Salvatore and Bjorn-Andersen Niels, "A contigency approach to managing information technology in developing countries: Benefiting from lessons learned in developed nations chapter in Information Technology in Developing Countries, Bhatnagar S.C and Bjorn-Andersen N. (editors)," Elsevier Science Publishers, North-Holland, 15-22, 1990.

[7] W'O Okot-uma Rogers, "A perspective of contextual, operational and strategy: Problems of Infomediation in Developing Countries a chapter in Social Implications of Computers in Developing Countries, Bhatnagar S C and Odedra Mayuri (editors)," Tata McGraw-Hill Publishing Company Limited, New Delhi, 10-25, 1992.

[8] Avgerou, Chrisanthi, "Information systems in developing countries: a critical research review," Journal of Information technology 23, 133–146, 2008

[9] Bhatnagar S C (editor), "Social Implications of Computers in Developing Countries," Tata McGraw-Hill Publishing Company Limited, New Delhi, 1992.

[10] Woherem Evans E., "IT Manpower Development Strategy at the Organisational level chapter in Information Technology Manpower: Key Issues for Developing Countries, Bhatnagar S.C. (editor)," Tata McGraw-Hill Publishing Company Limited, New Delhi, 147-157, 1992.

[11] 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.

[12] Brownsword, Lisa, Carney, D., Oberndorf, Tricia., "The opportunities and complexities of applying commercial-off-the-shelf components." Crosstalk 11, 4–6., 1998.

[13] Dean John C. and Vigder Mark R., "System Implementation Using Commercial Off-The-Shelf Software, Proceedings of the 1997 Software Technology Conference (STC 97)," Salt Lake City, Utah, May 1997.

[14] Torchiano, Marco, and Maurizio Morisio. "Overlooked Aspects of COTS-Based Development." Software, IEEE 21 (2): 88–93, 2004.

[15] Frankfort-Nachmias Chava and Nachmias David, "Research Methods in the Social Sciences," 7th Edition, Worth Publishers, 2007.

[16] Robson Colin, "Real World Research: A Resource for Social Scientists and Practitioner-Researchers" 2nd Edition, Wiley-Blackwell, 2002.

[17] Kunda Douglas, "A social-technical approach to selecting software supporting COTS-Based Systems," Doctoral Thesis, Department of Computer Science, University of York, York, 2001.

[18] Curtis Bill and Krasner Herb and Iscoe Neil, "A field study of the software design process for large systems," Communication of the ACM, 31(11):1268-1286, November 1988.

[19] Friedman Batya and Kahn, Jr. Peter H., "Educating Computer Scientists: Linking the social and technical," Communication of the ACM, 37(1):65-70, January 1994.

[20] Maiden Neil A. , H. Kim, and Ncube Cornelius,

ISSN:2319-7900

"Rethinking Process Guidance for Selecting Software Components." In COTS-Based Software Systems, 151–64. Springer, 2002..

[21] Brown Alan W. and Wallnu Kurt C., "Engineering of Component-Based Systems, in Proceedings of the 2nd IEEE International Conference on Engineering of Complex Computer Systems," IEEE Computer Society Press, Los Alamitos, California, October 1996.

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

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

[24] Mumford Enid, Hickey Steve and Holly Matthies, "Designing Human Systems: An agile approach to ETHICS," Lulu.com, S.l. 2006.

[25] Brooks Laurence and Kunda Douglas, "Social-Technical COTS Development: The STACE Contribution," Journal of Intelligent Systems 15 (1-4), pages 177-202, 2006

[26] Laudon K.C. and Laudon J. P., "Essentials of Management Information Systems," Prentice Hall, 10th Edition, 2012.