IROTS: A Proposed COTS Evaluation & Selection Methodology for Component Based Software Engineering in Under-Development Countries

M. Shakeel Faridi & Zahid Javed, M. Haris Abid, Mudassar Ahmed, Dr. Md Asri Bin Ngadi
Department of Komputing
Universiti Teknologi Malaysia,
Johor Bahru, Malaysia
shakeelfaridi@gmail.com, zahidjaved.uaf@gmail.com, harisabid@hotmail.com, mudassar.utm@gmail.com, dr.asri@utm.my

***Abstract*—Component Based Software Engineering (CBSE) methodology deals with software systems by evaluation and selection of appropriate components. CBSE offers software developers a compact software development life cycle, reduce development times, cost effective and less strenuous efforts. In under development countries (UDCs) most of the IT developers are unaware about the COTS evaluation and selection methodology for component-based software engineering. The objective of this research work is to facilitate the IT Professionals for the selection of the best COTS component on the basis of software quality model ISO/IEC 25010. The proposed Idealize Recommendation Off-The-Shelf (IROTS) method will facilitate the IT developers for selection of COTS components on basis of software quality model ISO/IEC 25010. This optimized components evaluation and selection technique that brings the best software in market which will fulfill the stakeholders' requirements.**

Keywords----Component Based Software Engineering, Idealize Recommendation Off-The-Shelf (IROTS), COTS evaluation and selection, ISO/IEC 25010

I. INTRODUCTION

Component Based Software Engineering (CBSE) means making up, assortment and manipulative of software component. Commercially available software components are increased with the popularity of CBSE [1]. In the majority of the research work and its literature, the researcher find the solutions and methodologies have focused on the aspects of the development of components [2]. Optimal solution regarding cost of CBSE is not an easy task. Development using CBSE is not including selection or relating them but to assure requirements.

The traditional method to this new condition caused problems like loose the deadline, economical cost and good quality requirements. So the researchers are to find way to develop program in a quicker and cost effective way using COTS component [3].

There are many methods have been proposed by earlier studies to evaluation and selection COTS component, but under development countries some issues and problems that are still not considered by these proposed methods such as recognizing the discrepancies between user requirements [4] and selection COTS, cost criteria, technologically advancement, component specification, vendor selection and COTS as "Black box" natures etc. In addition, previous studies do not exploit previous COTS selection attributes and ignored its significant role to support both of evaluators and decision makers during the evaluation and selection process.

II. AVAILABILITY OF COTS COMPONENTS IN MARKET

There are different categories of COTS components that are available in market according to specific user requirements shown as figure 1.

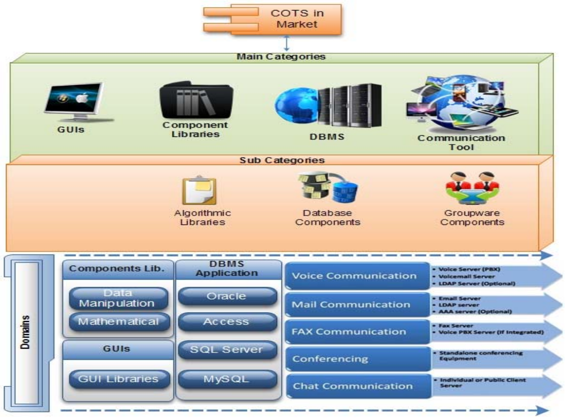

Fig. 1: AVAILABILITY OF COTS COMPONENTS IN MARKET

III. COMMON ISSUES OF COMPONENT-BASED SOFTWARE ENGINEERING

Table 1 is showing the common issues which are facing by IT developers during COTS evaluation and selection process.

TABLE 1COMMON ISSUES OF COMPONENT-BASED SOFTWARE ENGINEERING

Problem	Key point
Component specification	What a module is, and how it would be measured [5, 6].
Component Models	Software strategy and descriptions of the record on the whole used reproductions [7, 8].
Requirements Negotiation	Critical needs that should always be satisfied [9].
Domain coverage	required capabilities compulsory to meet

© 2013. The authors - Published by Atlantis Press

	core essential customer requirements
Time restriction	Software development time constrains & schedule
Cost rating	It includes features like Licensing cost, training cost, COTS adaptation and maintenance prices
Vendor guaranties	It must verifying the customer support provided by the vendor, vendor status and market position.
Technologically Advancement	knowledge come forward time to time so our developers, users, and project managers must be conscious
License Type	constituent licensing that industrial support provided by the vendor
Unawareness of well-defined process	unskilled person know about the well-define process[10]
Evaluation criteria	Lake of predefined criteria i.e. confusing know-how with product attributes [11].
COTS as "Black box" natures	Mismatch and integration problem. No way to test and no access to the COTS internals matter so this reasons difficult to recognize COTS components.

IV. Related Worked

In this research work we proposed Idealize Recommendation Off-The-Shelf (IROTS) method which facilitates the IT developers for selection of COTS components on basis of software quality model ISO/IEC 25010. This optimized components evaluation and selection technique fulfills the stakeholders' requirements and provides the best software in market.

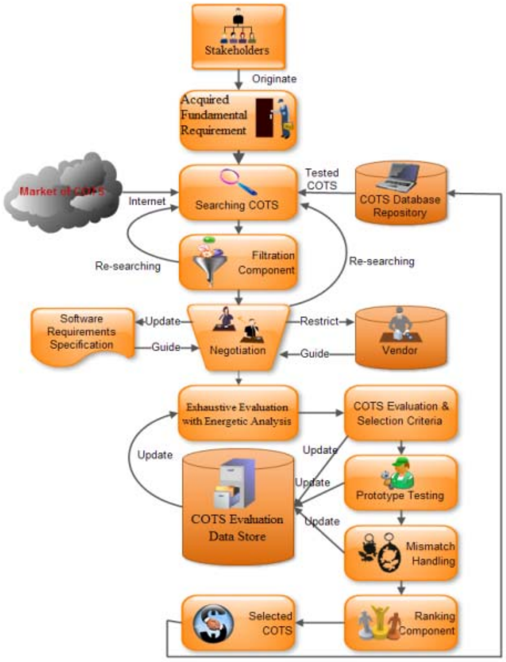

Fig. 2: The Overview of the Proposed IROTS Process

Some limited techniques of COTS selection have already been proposed. These are based on different decision making way for COTS evaluation and selection. Our COTS technique IROTS is based on ISO/IEC 25010. IROTS technique is especially proposed for under-developed countries.

The systematic process of the proposed COTS selection processes are Requirement Acquisition, Searching COTS (Internet & COTS Repository), Filtration, Negotiation, Exhaustive Evaluation with Energetic Analysis, COTS Evaluation & Selection Criteria like Prototype Testing, Mismatch Handling, Ranking the Components, COTS Selection as shown figure 2.

A requirement structure is the root of any useful COTS selection procedure as they supply the reception criterion of any COTS.

While requirement elicitation phase all stakeholders like end users, system administrators, operators, maintainers, integrators, architects, sponsors program managers and vendors are involved in interview and brain storming sessions. After completing the requirement elicitation the deliverable is vision document. Vision document is a software process document which is responsible for showing vision or plan for specific unit of software. It is written description of stakeholder’s requirements. It has both the outline and detail of requirements.

In searching COTS phase, it is observed that what should be reusable. Selection process begins with keyword searching technique in which a keyword is being searched from COTS sources. These sources can be own created libraries of in-house organizations or internet.

In filtration phase, we find out most accurate selection among selections. List of explored result is sort out for saving time and effort to made evaluation easy. COTS filtration is done according to requirements like system software, environment or cost. Requirement checklist has a major influence while searching out COTS.

Prioritization is a more difficulty phase where Stakeholders are always fails to prioritize the requirements and also fail to entertain the least priority.

In Exhaustive Evaluation with Energetic Analysis phase, we used to analyze behavior of output result of filtration phase by pilots, trial version, hands on demonstrations or execution. The results are documented after assessment. Black box natured dynamic analysis performs well with COTS because it provides .exe files, design requirement with no source code and testing demonstrations. The technique relates openly to COTS components, which are mainly difficult to tackle with fixed testing and analysis practice.

Best fit selection can be done by given evaluation criteria. Stakeholder’s requirements provide a way to select best COTS component [12]. Consent with Kontio are recommended for evaluation at equal time at given COTS valuation. The practical prose is also having COTS estimate according to superiority Model. For example, French *et. al.*, [13] and Carvallo *et. al.*, [14] projected quality model contained 6 steps for appraisal COTS. ISO/IEC 9126 quality model have definite actions for assessment COTS.

In prototype phase, a pre developed or pre suggested component is selected for real scenario. Prototyping is used to help out developer for better understandings of customers’ requirements. By using prototyping developer can understand user’s point of satisfaction. Prototyping is normally done at in-house development scenario.

Mismatch handling development is subdivided on two actions 1.Classify Mismatches 2. Suggest Mitigation Action.

There are four mismatches categorizes i.e ascertainable mismatches (Resolvable), alternatively feature, ignorable mismatches and improvable mismatches. COTS mismatches can be moderate by taking following two improvement exploit i.e Refining Requirements and Resolvable Mismatches. Analyze new evaluation position of every most gifted candidate. Standing of COTS on the basis of the COTS attribute, less error, cost, and Vendor service after behavior the mismatches is afterward done. In selecting the COTS component phase developers must assessment the COTS component and update the COTS evaluation data store repository that will help next time to select a COTS component.

V. MAPPING OF COTS SELECTION ATTRIBUTES WITH ISO 25010 QUALITY MODEL

In fifth phase of IROTS, we comparisons our own proposed COTS selection attributes and ISO/IEC 25010 software quality standard for characteristics, Sub-characteristics and its metrics. The ISO/IEC 25010standards emphases on software quality that can be used to work out own software value invention and software quality attributes. The ISO 25010 standard is made up of eight main attributes as shown table 2:

TABLE 2MAPPING OUR PROPOSED ATTRIBUTES WITH QUALITY MODEL 25010

ISO-25010	**(Sub)Characteristic**	**Our Attributes**
Functional suitability	Functional	---
	Functional Completeness	API Breakage,
	Functional correctness	---
	Functional Appropriateness	Upgrades system,
Performance suitability	Performance	Performance
	Time-behavior	---
	Resource utilization	---
	Capacity	Dormant Functionality,
Comp-atibility	Compatibility	Inter-component Compatibility,
	Co-existence	---
	Interoperability	Version Compatibility, Interoperability
Usability	Usability	Usability
	Appropriateness Recognisability	Documentation
	Learnability	Learnability,
	Operability	---
	User error protection	---
	User interface aesthetics	---
	Accessibility	---
Reliability	Reliability	Evolutionary period, Catastrophic,
	Maturity	Product maturity
	Availability	Alternative COTS,
	Fault tolerance	Uncertainties, Failure Probability
	Recoverability	---
Security	Security	Safety/Security
	Confidentiality	---
	Integrity	---
	Non-repudiation	---
	Accountability	---
	Authenticity	---
Maintainability	Maintainability	Alternative Plans,
	Modularity	---
	Reusability	---
	Analyzability	---
	Modifiability	Modifiability,
	Testability	Software Requirements Specification, Integrating Testing, User Acceptance Testing, User Acceptance Testing, Stress Testing, Performance Testing
Portabi-lity	Portability	Cross Platforms, Portability,
	Adaptability	---
	Install ability	Installation / Uninstallation,
	Replace ability	---
No Sub-characteristic correspondent attributes		Freeze creeping user requirement, Bad Standards, Change frequency, License Type, COTS development, Vendor Attitude, Prototyping, COTS Type, Bayesian Belief Networks, Analytic Hierarchy Process, Ordinal Metric, Binary Metric, Software License Cost, Integration Cost, Training Cost, Hardware Upgrades Cost, Glue Code Cost, COTS Upgrade Cost, Testing Cost, Modification Cost, Vendor Concessions, Delivery, Geographic location, Customer support, Reputation and position in industry, Domain specific, Market share, Warranties and claims,

In addition to the above core qualities, we have comprised these qualities that are not straight related to the excellence in custom, Market segment, Geographic location Reputation and position in industry produce provision, permit kind, gaining charge, and area precise etc. All these above given attribute emphasis on exterior software qualities that are frequently used to define software machineries when purchasing (COTS) commercial off-the-shelf. These qualities are imperative when gauging different COTS for influential the definite cost of software purchase, Vendor selection and Decision Making Analysis.

VI. CONCLUSION

The purpose of this research work is to aware the IT developers about the COTS evaluation and selection methodology on the basis of software quality model ISO/IEC 25010 for component-based software engineering in UDCs. The proposed Idealize Recommendation Off-The-Shelf (IROTS) method (a seven-step COTS variety process in which the appraisal of COTS component) will facilitate the IT developers for selection of COTS components on basis of software quality model ISO/IEC 25010. This optimized components evaluation and selection technique that will bring the best software in market to fulfill the stakeholders' requirements. In future work we will try to implement this process model in IT industry that will facilitate the IT developers to select the best COTS component as per stakeholders' requirements.

REFERENCES

[1] J. Kaur and K.S.Mann," Component Selection for Component based Software Engineering", International Journal of Computer Applications, volume 2, issue 1, 2010, pp. 109 – 114

[2] Geisterfer, C. J. M. and S. Ghosh, 2006. Software component specification: a study in perspective of component selection and reuse. IEEE, pp. 9.

[3] Fahmi, S. A. and H. J. Choi, 2008. Life Cycles for Component-Based Software Development. IEEE, pp. 637-642.

[4]Tarawneh, F., F. Baharom, J. H. Yahaya and A. Zainol, 2011. COTS software evaluation and selection: A pilot study based in Jordan firms. IEEE, pp. 1-5.

[5] Cheesman J. and Daniels J., UML Components - A Simple Process for Specifying Component-Based Software, ISBN 0-201-70851-5, Addison- Wesley, 2000.

[6] Shaw M. and others, Abstractions for Software Architecture and Tools to Support Them, IEEE Transaction on Software Engineering, volume 21, issue 24, 1995

[7] Bosch J. and Stafford J., "Architecting Component-based Systems", in Crnkovic I. and Larsson M. (editors): Building Reliable Component-Based Software Systems, ISBN 1-58053-327-2, Artech House, 2002.

[8] Estublier J. and Favre J.-M., "Component Models and Technology", in Crnkovic I. and Larsson M. (editors): Building Reliable Component-Based Software Systems, ISBN 1-58053-327-2, Artech House, 2002.

[9] Korel B., Black-Box Understanding of COTS Components, In Proceedings of 7th international workshop on program comprehension, 1999.

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

[11] 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, Los Alamitos, 1997

[12] J. Kontio, “OTSO: a systematic process for reusable software component selection,” Institute for Advanced Computer Studies and Department of Computer Science, University of Maryland, Tech. Rep., 1995.

[13] X.Franch &J.P. Carvallo, "Using Quality Models in Software Package Selection, "Software,vol.20(1),Jan/Feb 2003. pp. 34-41

[14] J. P. Carvallo, X.Franch, G. Grau, &C. Quer,"COSTUME: a method for building quality models for composite COTS-based software systems," in QSIC'04, Germany, 2004, pp. 214 – 221