

Optimal performance model investigation in component-based software engineering (CBSE)

Muhammad Osama Khan, Ahmed Mateen, Ahsan Raza Sattar

Department Of Computer Science, University of Agriculture Faisalabad, Pakistan

Email address:

osamagoldpk@hotmail.com (M. O. Khan), ahmedmatin@hotmail.com (A. Mateen)

To cite this article:

Muhammad Osama Khan, Ahmed Mateen, Ahsan Raza Sattar. Optimal Performance Model Investigation in Component-Based Software Engineering (CBSE). *American Journal of Software Engineering and Applications*. Vol. 2, No. 6, 2013, pp. 141-149.

doi: 10.11648/j.ajsea.20130206.13

Abstract: Commercial off-the-shelf (COTS) technologies have emerged over the past decade. COTS technology gained significant popularity by developing the optimal, efficient, economically and quickly software system that mapping business requirement. As a consequence, the need for designing effective strategies for enabling large scale reuse, whilst overcoming the risks involved in the use of a particular technology, still remains. In this situation, the use of “COTS” technology introduces many problematic factors that still have not been fully solved; some of them are the lack of inclusive tools, efficient methods to manage and collect the required information for supporting COTS software selection. Keeping in view all these issues in this research report present an Optimal Performance Model (OPM) for gathering the information that is needed to define COTS market segments in a way that would make software components selection more effective and efficient. Mostly the information we collect possess huge diversity therefore suggest OPM’s that will certainly help to cover different aspects and fields of COTS software selection. This design model will base on several software quality standards. Commercial of the shelf software has gained considerable popularity as approach that quickly and economical creates software system that address business requirement. This research work will presents an approach for defining assessment principles for reusable software components.

Keywords: COTS, CBSE, ERP, DBMS, BPR

1. Introduction

An efficient and effective software components selection method is necessary to deliver full Capable to the COTS technology. There are many COTS selection techniques, processes and methodologies that have been articulated. COTS-based systems development requires specific activity like desire component selection, composition and last but not least the most important is component integration. After the successful working of this technique it became more popular as a result of this today a large number of software components are commercially available, components selection a to fulfill the all system requirements which minimize the cost of the system software [21]. In COTS for large system are become common a days. Due to low budget, enhance the system requirement, accelerating rates of COTS improvement defined by software Engineering Institute (SEI). Building cost optimal components is very difficult job. Selecting the best components in cots selection presented many challenges to software engineer to

retain in new arena. They should consider many constraints. Two major constraints, first selecting optimal components and then their integrity should be considered [1]. In this research, the optimal model for component selection is described. Informally, first choose a set of software components from accessible component set which can satisfy all system requirements for the final product or set of products. To optimize and analyse a set of requirements should assign to each component and should allocate a cost which is the overall cost of selection and integration of that components. Many organizations have claimed for improving the productivity and quality of software development using COTS Selection [28]. Therefore many software organizations have introduces COTS technologies in the software industry. A number of COTS components supporting tools have been introduce by software engineers and consultants .The popularity increase of commercial and embeddable software components and software environments following standardization [17]. The quality and standard components more available so more potential reuse candidates. Many devel-

oping organizations are spending a large amount of time in COTS selection since the choice of the suitable components has a major effect on the project and final product [17]. We have developed OPM that addresses the selection process, Components software package. OPM has been introduced which supports the selection, search, and evolution of reusable software components and provides specific procedures for defining the development quality standards. If OPM for selection component is not defined then each project finds its own procedure to complete it. This model compares the cost of individual components for cost optimization and analyses the benefits of alternative selection components [28]. Software development organization initiated process for definition of requirements for customer business process in COTS Enterprise Resource Planning (ERP). The software modules are available for commercial vendors that support business processes such as production planning, finance, human resource, accounting and procurement. COTS-ERP integrating the data required for these business processes in a logically single database [7]. COTS ERP systems are based on the principle that the software vendor can map the common business processes more effectively and efficiently than customer organizations. Because ERP software is maintained by the vendor and developing of business processes defined by the vendor. Return on the investment (ROI) of customer and Total cost savings are based on the developing organization adopting the business process [8]. Mostly ERP system the required limited customization through changes to configuration settings in the specific module. Configuration setting of the components up to standard then software required less maintenance costs, if configuration settings is unsupported then software maintenance costs is increase, consequently reducing the overall return on the consumer's investment [13]. More and more organizations going toward standardization regarding their process then resulting several organizations like public, state, local government and private are either implementing or planning to implement COTS-ERP systems. COTS -ERP system integrating the required data from these modules in a logically single database [28].

2. Objectives

OPM will satisfy the consistent and relevant system requirement with high quality and reliability for software development. OPM will select and test the candidate components. Component development will lead to the final components that satisfying the requirements with correct

and expected results. The OPM are to select and test the candidate components and check whether they satisfy the system requirement with high reliability and quality.

The system architecture design are to collect the users requirement, identify the system specification, select appropriate system architecture, and determine the implementation details such as platform, programming languages, etc.

3. Materials and Methods

The increased commercial availability of embeddable software components, standardization of basic software environments (such as Microsoft Windows, UNIX), and the explosive popularity of the Internet has resulted in a new situation for reusable software consumers: there are many more accessible reuse candidates. Consequently, many organizations are spending much time in reusable component selection since the choice of the appropriate components has a major impact on the project and resulting product. The component selection process is not defined so each project finds its own approach to it. Here a method has been introduced which supports the search, evolution and selection of reusable software and provides specific techniques for defining the evolution criteria, comparing the cost and benefits of alternatives and consolidating the evolution and selection and benefits of alternatives and consolidating the evolution results in decision making.

Component-Based Software Engineering (CBSE) is concerned with composing, selecting and designing components. As this approach become the more popular of and consequently several commercially available software components grows, selecting group of required components to full fill the requirements even though minimizing cost is becoming more hard. Optimal component systems selection is not an easy task. It requires not only optimal components selection, although also to take their relationship into account. Selecting component problem described in this research paper. To accomplish this goal, we must to assign each component a set of requirements which satisfies. Each component is assigned a unit cost which is the overall cost of acquirement and integration of that component. Several organizations have supported their reuse with E technologies. Component Based ERP systems have packages which offered by commercial vendors. These packages support main administrative processes such as accounting, procurement, budgeting and human resource management. These integrated System in a single database.

Table 1. Difference between Traditional software development and component based Software development

Traditional Software Development	Custom Development	Component based Software development
Requirements	Creation of system requirements to create a Software system that meets these requirements (the engineers are producers).	Creation of a set of flexible requirements followed by the COTS marketplace exploration for selecting components that best fit these requirements. The engineers are consumers who then integrate the products.
Design	Analyze requirements to produce a depiction of the internal structure and integration of the system that	To integrate the products into a software system that meets the requirements. It implies an iterative

Traditional Software Development	Custom Development	Component based Software development
	will serve the foundation for its construction.	trade-off process of requirements analysis, architecture, COTS availability, prioritization and negotiation.
Construction	Coding the detailed design to implement the system requirements	Some requirement functionalities that were not addressed by any COTS are usually developed in-house. In any case, usually glue code is used to mediate components interactions; as well as bridges or adaptors to smooth over incompatibilities in the component interfaces.
Testing	Integration and evaluation of the product quality by verifying its behavior by a finite set of test cases	Although COTS are tested by the component provider, they should be retested by the user to assure their suitability and their good system integration
Maintenance	Modification to code and associated documentation due to a problem or need for improvement	Due to maintenance effects, COTS-Based systems undergo a technology refresh and renewal cycle that has many implications.

Table 2. Advantages and disadvantages of using COTS software

Advantages	Disadvantages
Immediately available; earlier payback	Licensing, intellectual property procurement delays
Avoids expensive development	Up-front licensing fees
Avoids expensive maintenance	Recurring maintenance fees
Predictable, confirmable license fees and performance	Reliability often unknown or inadequate; scale difficult to change.
Better functionality	Too-better functionality compromises performance and usability.
Repeated upgrades often expect organization's needs	No control over maintenance and upgrades.
Broadly mature and use technologies	Constraints on efficiency and functionality.
Devoted support organization	Dependence on vendor
roadway technology needs	Synchronizing multiple-software vendors.
Hardware /software independent.	Integration not always trivial; incompatibilities among vendors.

Table 3. ERP implementation models

Author(s)	ERP implementation model	Notes
Bancroft et al.(1998)	(1) Focus, (2) Creating As-Is picture, (3) Creating of the To-Be design, (4) Construction and testing, and (5) Actual Implementation	A model of IT projects
Kuruppuarachchi et al. (2002)	(1) Initiation, (2) Requirement definition, (3) Acquisition/development, (4) Implementation, and (5) Termination	
Markus and Tanis (2000)	(1) Project chartering, (2) The project, (3) Shakedown, and (4) Onward and upward	
Mäkipää (2003)	(1) Initiative, (2) Evaluation, (3) Selection, (4) Modification, Business Process Reengineering, and Conversion of Data, (5) Training, (6) Go-Live, (7) Termination, and (8) Exploitation and Development	Three parallel phases in phase number 4.
Parr and Shanks (2000a)	(1) Planning, (2) Project: a. setup, b. reengineer, c. design, d. configuration & testing, e. installation, (3) Enhancement	Applied from Kwon and Zmud's (1987) model of IT implementation
Rajagopal (2002)	(1) Initiation, (2) Adoption, (3) Adaptation, (4) Acceptance, (5) Reutilization, and (6) Infusion	
Ross (1999)	(1) Design, (2) Implementation, (3) Stabilization, (4) Continuous improvement, and (5) Transformation.	
Shields (2001)	Rapid implementation model of three phases and 12 major activities	Model of the ERP Acquisition Process (MERPAP)
Umble et al. (2003)	(1) Review and process to date before implementation, (2) test and Install any new hardware (3) Install the software. (4) Get system training, (5) Training on the conference room pilot, (6) Establish security and necessary permissions, (7) Ensure that all data bridges are sufficiently robust and the data are sufficiently accurate, (8) Document policies and procedures, (9) Improve continually. Bring the entire organization on-line, either in a total cutover or in a phased approach.	
Verville and Halingten (2003)	(1) Planning, (2) Information search, (3) Selection, (4) Evaluations, and (5) Negotiation	

The Critical Success Factors in ERP implementation are described in addition to of the general literature on IT implementation, project management and Business Process Reengineering (BPR). This list was rated by IT managers. The consensus concerning ERP implementation is that multiple factors affect success. They have defined implementation project success as a function of critical success factors (CSFs). ERP implementation research has widely tried to specify the CSFs of ERP implementation projects.

Table 4. Critical success factors of ERP implementation

	Critical success factor	Mean	Std. Dev
1	Top management support	4.29	1.16
2	Project team competence	4.20	1.07
3	Interdepartmental cooperation	4.19	1.20
4	Clear goals and objectives	4.15	1.14
5	Careful package selection	3.89	1.06
6	Interdepartmental communication	4.09	1.33
7	Dedicated resources	4.06	1.37
8	Project champion	4.03	1.58
9	Project management	4.13	0.96
10	Vendor support	4.03	1.6
11	Data analysis & conversion	3.83	1.27
12	Management of expectations.	3.81	1.25
13	Use of steering committee	3.79	1.95
14	Minimal customization	3.79	1.16
15	Education in new business processes	3.76	1.18
16	Business Process Reengineering, BPR	3.68	1.26
17	User software training	3.68	1.45
18	Architecture choices	3.44	1.19
19	Use of vendors' tools	3.43	1.34
20	Partnership with vendor	3.39	1.21
21	Change management	3.15	1.57
22	Use of consultants	2.90	1.20

3.1. Optimal Performance Model

We suppose that components are provided with interfaces that explicitly declare the component estimated performance properties. The estimate performance which better fulfills the non-functional and performance-related requirements could be used by the Software Architect. Items in the factor “Project Success” are divided into two groups “Progress” and “Quality”. The new factor “Progress” includes questions about project completion on time and within budget, while “Quality” has questions related to system quality and the scope matched with the company’s needs. The other variables remained the same as they were in the pilot survey. “Function” is the most important factor for “Quality” of the ERP system. It indicates that selecting the right software and defining the necessary functions should be given the most consideration to enhance the overall quality of the ERP system. “Consultant Support” can also impact on “Quality”, but there is no impact expected from “Internal Support”.

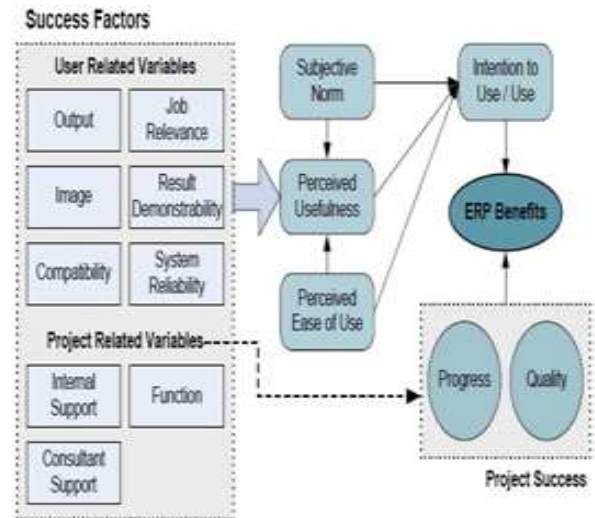


Figure 1. Optimal Performance Model

3.2. Survey Questionnaire

This questionnaire (which starts on the following page), gives you an opportunity to tell us your reactions to the SAP-ERP applications you used. The user responses help us to understand different aspects of the ERP system. The results have been used for research purpose only and we will not identify any Organization or individual in any publication.

Please read each statement and indicate how strongly users agree or disagree.

Table 5. Evaluating Oracle ERP Application

1	Name (Optional):
2	Current Position:
3	Work Experience with SAP-ERP Applications:
4	Work Experience with other ERP Applications:
5	No of ERP Implementation Project done & Roles:
6	Names of ERP Implementation Organization

The survey tool is planned based on the OPM model, and most relevant parameter in the survey are primarily adapted from the relevant earlier research in the IS contexts. The main survey was conducted to analyse ERP success before proposed model well developed. The main survey to examine whether or not the Data analysis including correlation, factor analysis and reliability test was conducted to adjust survey items and extract factors associated with the success of ERP systems. The new survey instrument which has been used in the main survey and the revised ERP success model after adjustment with the pilot survey are proposed in this chapter. The revised model looks much simpler than the conceptual model.

QUESTIONNAIRE

- 1 It is easy to learn and understand several transactions involves for completing one business process?
- 2 It is easy to learn and understand the total number of transactions involves for completing whole Module level business process?
- 3 Overall, ERP application requires minimum number of transactions to complete business processes?
- 4 I am satisfied the way transactions and business processes are organized in ERP application?
- 5 Business processes wise training or reference material for different module is easily available from ERP Vendor or Consultant?
- 6 It is easy to perform data cleansing & harmonization in ERP application?
- 7 It is easy to perform data transformation task in ERP application?
- 8 It is easy to perform testing and data loading in ERP application?
- 9 It is easy to perform data migration by using API method?
- 10 Overall, it is easy to learn the process of data migration in this ERP application?
- 11 Once learned, it is easy to remember data migration process after a period of non-use (at least one or few months)?
- 12 Service response time on error messages from ERP vendor is too slow?
- 13 Many peoples handling the same error for service request causes slow responses time?
- 14 I like using the User interface of ERP application?
- 15 ERP application has all the functionalities and capabilities that I expect it to have?
- 16 Overall, I am satisfied with this ERP application?
- 17 In ERP implementation projects the degree of top management team contributions in support, planning and training, with respect to respondents.
- 18 ERP software can support its business processes as well as the functionality of the respondent's company.

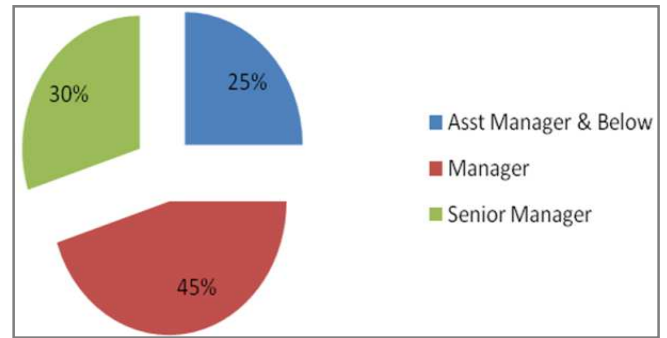


Figure 2. Survey Respondents' Position in their Company

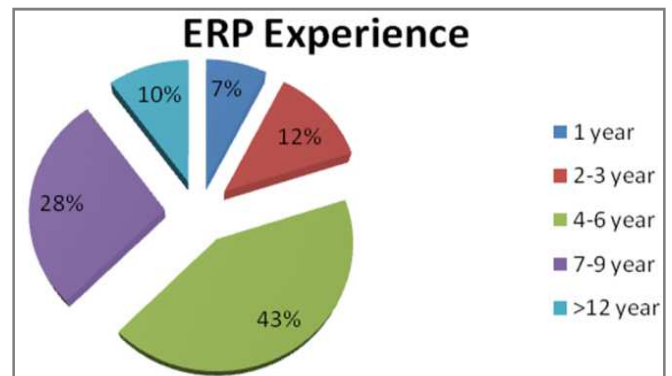


Figure 3. No of year SAP ERP Experience

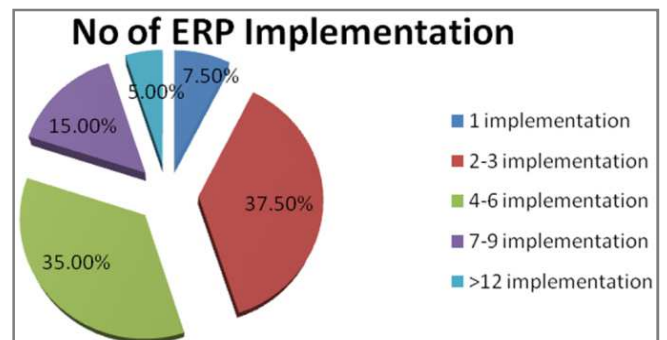


Figure 4. No of year SAP ERP Implementation

4. Results and Discussion

In last four decades information systems have been developed, There are so much research has been conducted to comprehend the develop systems customized to a specific organization. During the last 10 years it however became clear that a different practice had slowly emerged in the area of Enterprise Systems Software. Thus over the years it had become more and more common for software providers to reuse existing code (code developed for prior customer) whenever developing an Information System for a new customer. Thus most Enterprise Software had become pre-developed software targeted at specific lines of businesses instead of tailor-made software for a specific customer. This phenomenon can be observed for different kinds of software e.g. Enterprise Resource Planning (ERP) software, Supply Chain Management (SCM) software, and Project Management (PM). In this thesis I will focus on ERP package software.

The pilot survey was developed by using Google docs and was conducted as a web-based survey. The link to the survey was sent to the contacted individuals so that they can distribute it to other possible participants.

In ERP software consists of number of modules. In the organization, each module deal with a specific functional area e.g. Purchase Inventory, Quality Management, Plant Maintenance, human resource, finance, warehousing, sales or project management, and all the modules in the ERP System are integrated. ERP have common database. The final product in the Enterprise Package Software is marketed "best practice". Nevertheless, "best practice" does not necessarily mean that only one version of a process is supported by the software. Many functional areas have businesses or different practices experienced. This research originated the conceptual ERP success model

based on theories and knowledge gained from several industry practitioners. DeLone and McLean's IS success model was used for recognized success indicators. The OPM model adapted the TAM as the Opening point for the arrangement of relationships between factors and indicators. In conclusion, the fundamentals of project management were incorporated into the OPM model for investigated the success of ERP implementation. As a result, for better understanding about the success of ERP systems this OPM model is theoretically sound and can be helpful.

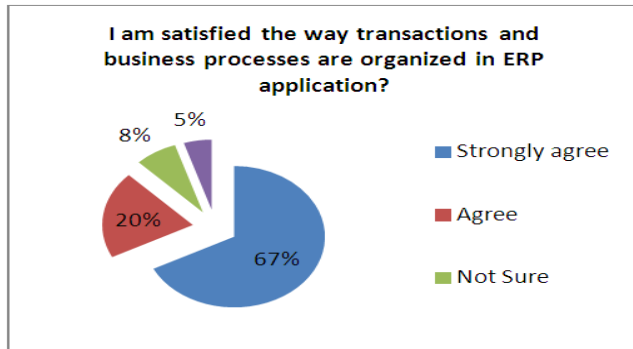


Figure 5. Survey Question

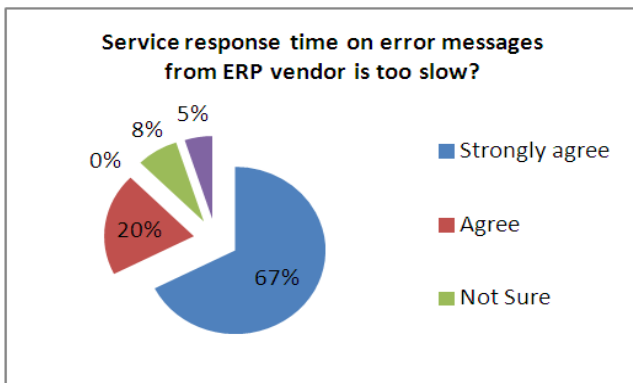


Figure 6. Survey Question

The main research result here is that the new factor proposed from this research, "Function" is the most significant factor to be absolutely linked with "perceived usefulness". Most of the users understand that if the functionality of their ERP system is good enough for business process.

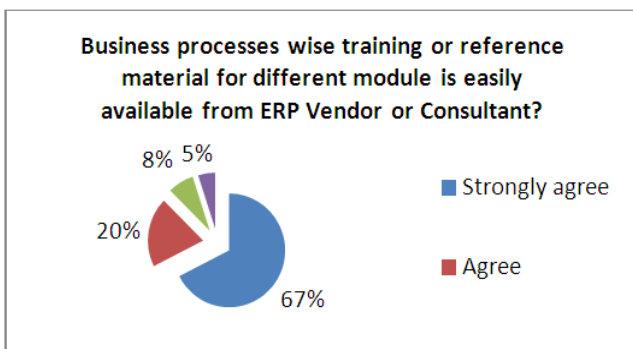


Figure 7. Survey Question

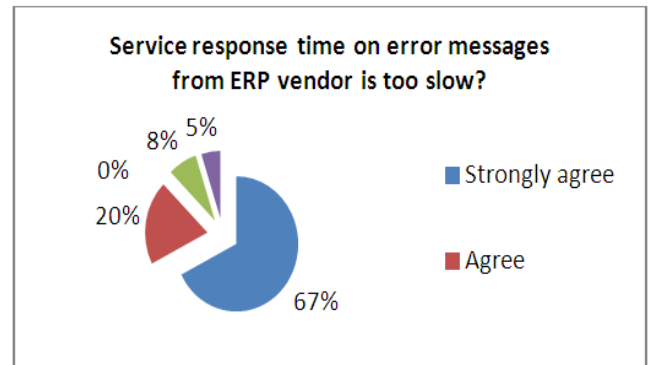


Figure 8. Survey Question

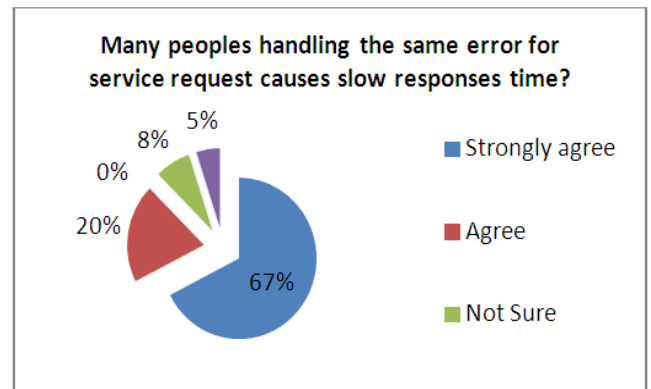


Figure 9. Survey Question

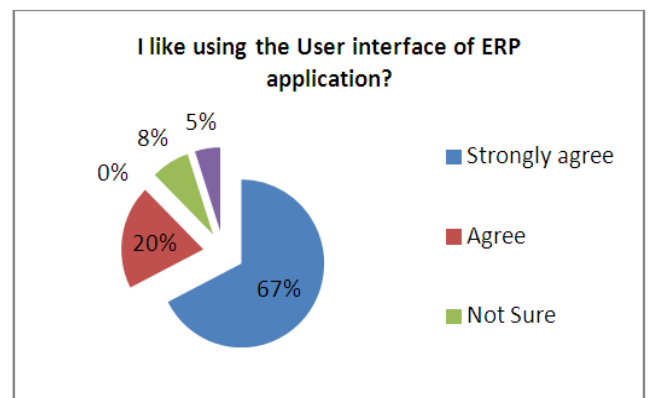


Figure 10. Survey Question

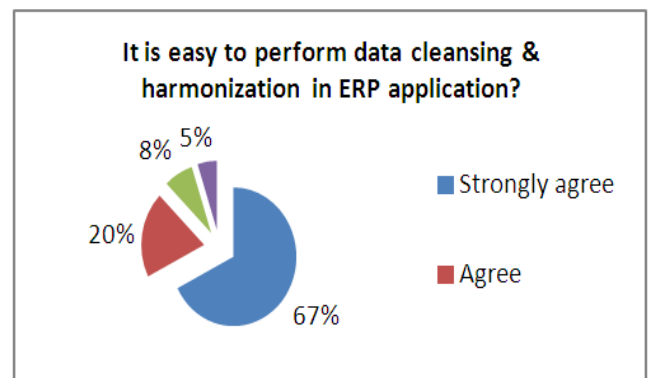


Figure 11. Survey Question

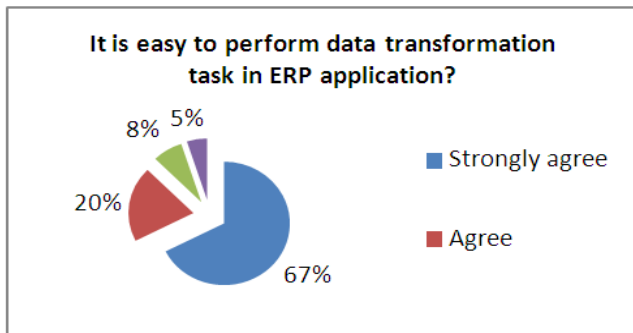


Figure 12. Survey Question

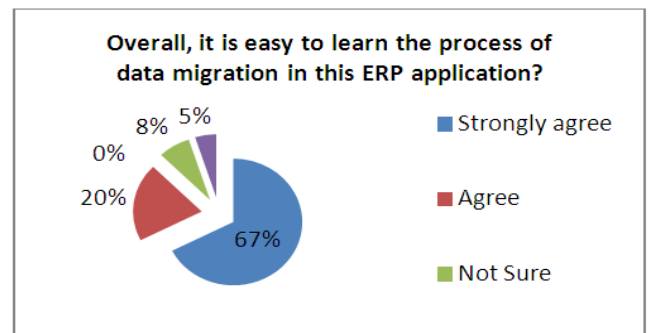


Figure 16. Survey Question

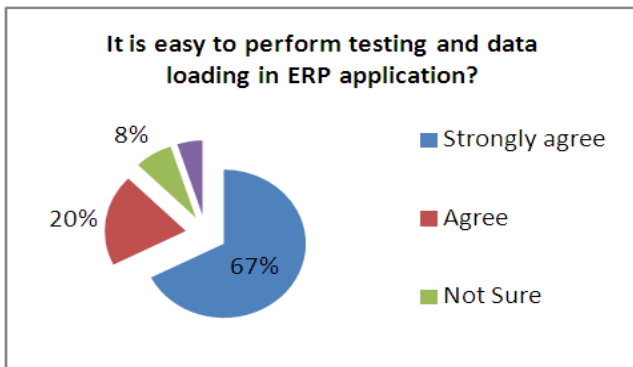


Figure 13. Survey Question

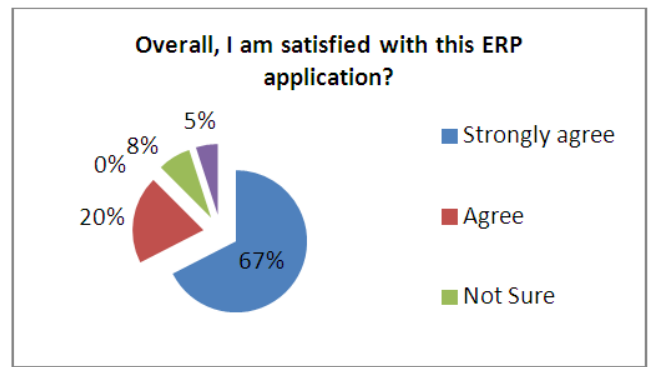


Figure 17. Survey Question

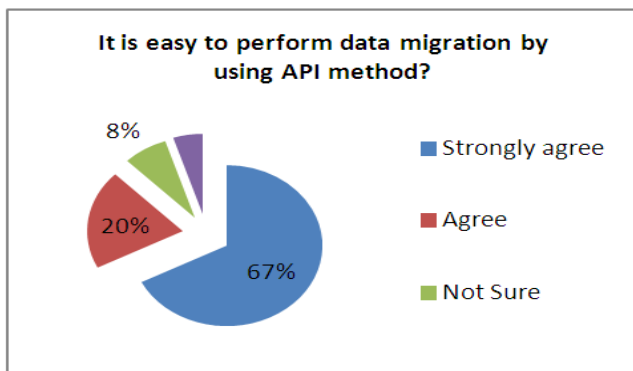


Figure 14. Survey Question

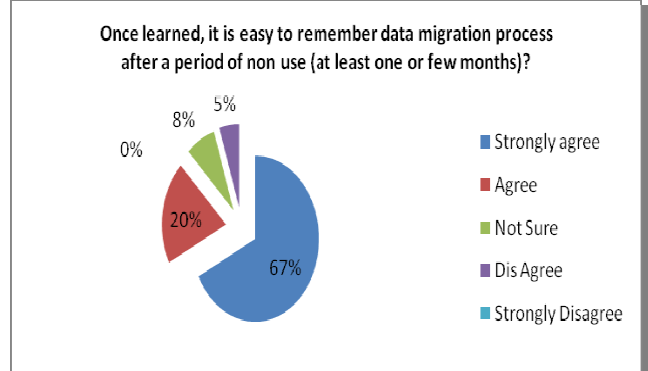


Figure 18. Survey Question

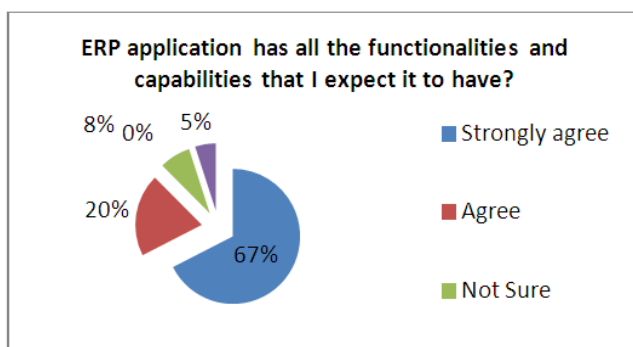


Figure 15. Survey Question

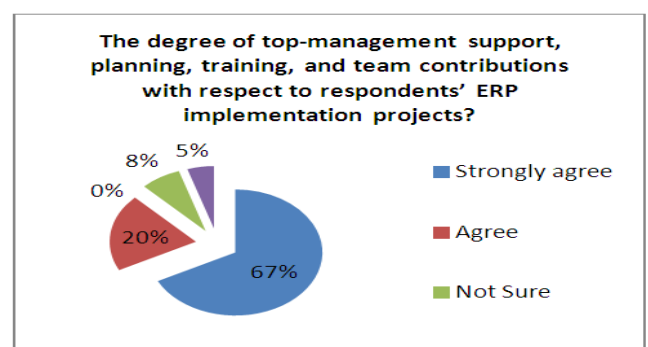


Figure 19. Survey Question

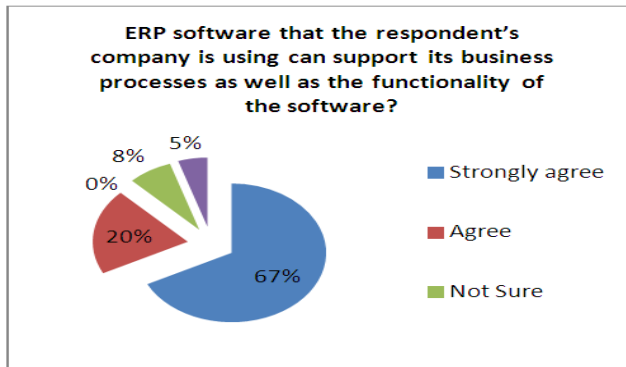


Figure 20. Survey Question

The Mostly Users think ERP system is useful. Moreover, how business requirements match the functions of the ERP system. Mainly make the ERP system is more useful. Dissimilar other information traditional systems, necessary functions across the departments in the ERP system needs to integrate within an organization to be fully valuable. This exact feature causes “Perceived Usefulness” have the major impact on “Function” to. It should be renowned that the ERP system must be considered part of business functions and processes sooner than an information system.

It is analyzed that results point to that more use and better quality of ERP systems can add to the benefits of ERP. The two project success indicators (i.e. “Quality and Progress”) have a positive impact on success indicator (i.e. “Intention to Use / Use”) and on the final dependent variable, “ERP Benefits”.

5. Conclusion

In this research discusses relationships between success factors and indicators in more detail, describing the reason behind their relationships. Interpretations of the results about each dependent variable are presented along with its independent variables based on its analysis and other relevant data from the survey. The section discusses why a difference exists in a particular variable, and interprets its meaning with respect to ERP success. In conclusion, the research proposed several suggestions for the accomplishment of ERP systems based on the results of recognizing indicators and the relationships between indicators. These suggestions should allow software engineering have a better understanding of ERP success and help them to abstain failure considering critical factors recognized to successful ERP completion.

Most IT related research in the area of construction business management generally proposes research models without theories. Furthermore, since this kind of research is still comparatively new to construction related research, a lot surveys have been formed without sound theoretical background. They usually recognize the factors which compare the mean values of factors, and rank factors in with their importance showing the higher mean value as the more important factor. This research projects first time

point out the factors affecting ERP success with strong background theories in construction business related research. The OPM adapted three theoretically validated models including D&M IS Success Model and TAM, the fundamentals of project management in ERP implementation. Therefore, the professional contribution of this research project can be found in a deliberate attempt to develop the ERP success model.

There have been few studies attempting to validate empirically the factors affecting both ERP implementation and user adoption. The factors identified in literature were mostly based on the experiences of IT professionals or senior managers involved in ERP implementation projects. For these reasons, this research focused on identifying the factors for the ERP success from both implementation project and user adoption perspectives. Then, identified factors were examined to verify their relationships with success indicators associated with the redefined ERP success, i.e. the success of the project and the success of use. Moreover, the research study proposed recommendations for the ERP success explaining how to approach ERP implementation to avoid failure and what to do more considering the importance of each factor to a given dependent variable based on the findings of the study. These recommendations can give valuable information to engineering firms when they want implementing or upgrading their ERP systems.

References

- [1] Aberdour, M. 2007. Achieving Quality in Open Source Software. *IEEE Software*, 24(1): 58–64.
- [2] Balbo, G., and G. Serazzi. 1997. Asymptotic Analysis of Multiclass Closed Queueing Networks: Multiple Bottlenecks. *Performance Evaluation*, 30(3): 115–152.
- [3] Brownsword, L., T. Oberndorff and C.A. Sledge. 2000. Developing New Processes for COTS-Based Systems. *IEEE Software*, 17(4): 48–55.
- [4] Balsamo, S., A. D. Marco, P. Inverardi and M. Simeoni. 2006. Efficient Performance models in Component-Based Software Engineering. *IEEE Computer Society*, 38(3): 64–71.
- [5] Balsamo, S., A.D. Marco, P. Inverardi and M. Simeoni. 2004. Model-Based Performance Prediction in Software Development. *IEEE Transactions on Software Engineering*, 30(5): 295–310.
- [6] Becker, S., L. Grunske, R. Mirandola and S. Overhage. 2006. Architecting Systems with Trustworthy Components. *Lecture Notes in Computer Science*, 3938(1): 169–248.
- [7] Bertolino, A. and R. Mirandola. 2004. CB-SPE Tool: Putting Component-Based Performance Engineering into Practice. *Component Based Software Engineering*, 3054(1): 233–248.
- [8] Becker, S., H. Koziol and R. Reussner. 2009. The Palladio Component Model for Model-Driven Performance Prediction. *J. Systems and Software*, 82(1): 3–22.

- [9] Chen, S., I. Gorton, A. Liu and Y. Liu. 2002. Performance Prediction of Component-based Applications. *Journal of Systems and Software*, 74(1): 35-43.
- [10] Crnkovic, I., M. Chaudron and S. Larsson. 2005. Component-based Development Process and Component Lifecycle. *Journal of Computing and Information Technology*, 13(4): 321-327.
- [11] Craig, D. C. (2007). Compatibility of Software Components-Modelling and Verification. Ph.D Thesis, Univ., Memorial, Newfoundland.
- [12] Denaro, G., A. Polini and W. Emmerich. 2004. Early Performance Testing of Distributed Software Applications. *ACM SIGSOFT Software Engineering Notes*, 29(1):94-103.
- [13] Fox, M. R., D. C. Brogan and F. Reynolds. 2004. Approximating component selection, *Winter Simulation Conference*, 25(5): 429-434.
- [14] Glenn A., A. Thomas and S. Jajodia. 2004. Commercial-Off-The- Shelf Enterprise Resource Planning Software Implementations in the Public Sector: Practical Approaches for Improving Project Success, *The Journal of Government Financial Management*, 53 (2): 10-15.
- [15] Grassi, V., R. Mirandola, and A. Sabetta. 2007. Filling the Gap between Design and Performance/Reliability Models of Component-Based Systems: SA Model-Driven Approach, *J. Systems and Software*, 80(4): 528-558.
- [16] Grozev, N. 2011. A comparison of component-based software engineering and model-driven development from the ProCom perspective. M.S thesis. Univ., Mälardalen, Sweden.
- [17] Hofmeister, C., P. Kruchten, R.L. Nord, H. Obbink, A. Ran and P. America. 2007. Generalizing a Model of Software Architecture Design from Five Industrial Approaches, *Journal of Systems and Software*, 80(1):106-126.
- [18] Kim, I.Y. and O. Weck. 2005. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*, 29(2):149-158.
- [19] Kontio, j. 1996. A Case Study in Applying a Systematic Method for COTS selection. *IEEE Computer Society*, 32(4):201-209.
- [20] Kuz.I., Y. Liu, I.Gorton and Gernot Heiser. 2007. CAMKES: A component model for secure microkernel-based embedded systems, *The Journal of Systems and Software*, 80(1): 687-699.
- [21] Kaur, A. and K. S. Mann. 2010. Component Selection for Component based Software Engineering. *International Journal of Computer Applications*, 2(1):12-18.
- [22] Lau, K. and Z.Wang. 2007. Software Component Models. *IEEE Transaction on Software Engineering*, 33(10): 3-8
- [23] Madanmohan, T.R. and R. De. 2004. Open Source Reuse in Commercial Firms. *IEEE Software*, 21(6): 62-69.
- [24] Mohamed, A. 2007. Decision Support for Selecting COTS Software Products Based on Comprehensive Mismatch Handling. Ph.D. Thesis, univ., Calgary, Alberta.
- [25] Mahmood, S. and A. Khan. 2011. An industrial study on the importance of software component documentation: A system integrators perspective. *Information Processing Letters*, 111(12): 583-590.
- [26] Navas, J. and J. Babau. 2009. Efficient and Adapted Component-Based Strategies for Embedded Software Device Drivers Development. *IEEE Computer Society*, 2(1): 514-519.
- [27] Peslak, A.L R., G.H. Subramanian and G.E. Clayton. 2008. The Phases of ERP Software Implementation and Maintenance: A Model for Predicting Preferred ERP Use. *Journal of Computer Information Systems*, 48(2): 25-33
- [28] Rawashdeh, A. and B. Matalakah. 2006. A New Software Quality Model for Evaluating COTS Components, 2 (4): 373-381.
- [29] Runeson, P. and M. Höst. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2): 131-164.
- [30] Torchiano, M. and M. Morisio. 2004. Overlooked Aspects of COTS-Based Development. *IEEE Software*, 21(2): 88-93.
- [31] Tekumalla, B. 2012. Status of Empirical Research in Component Based Software engineering. M.S thesis, Univ. Gothenburg, Sweden.
- [32] Vilpola, I. 2008. Applying User-Centred Design in ERP Implementation Requirements Analysis. Ph.D Thesis, Univ. Technology, Tampere.
- [33] Victor, S., C. Becerra and G. Valdes. 2010. Empirical Validation of Component-based Software Systems Generation and Evaluation Approaches. *Celi Electronic Journal*, 13(1): 2-9
- [34] Wu, X. and M. Woodside. 2004. Performance Modeling from Software Components, *ACM/SIGSOFT Software Eng. Notes*, 29(1): 290-301.
- [35] Wanyama, T. and B. Far. 2008. An Empirical Study to Compare Three Methods for Selecting Cots. Software Components. *International Journal of Computing and ICT Research*, 2(1): 34-41.
- [36] Winkelmann, A. 2012. Reference Model Maintenance Based On ERP System Implementations. *AIS Transactions on Enterprise Systems* 3(1):1-9