IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-6, Issue-11, Nov 2018 E-ISSN: 2347-2693

Empirical Studies on COTS Methodology

Reena[*1], **Pradeep kumar Bhatia**[1]

GJU&ST[1]**, Hisar/India**

Available online at: www.ijcseonline.org

Accepted: 20/Nov/2018, Published: 30/Nov/2018

Abstract- Commercial Off-The-Shelf (COTS) components are being used in increasing number to reduce cost and delivery time of software. A commercial Off-The-Shelf (COTS) component is becoming more and more important since it promotes reuse to higher levels of abstraction. As a consequence, many components are available either being open-source software (OSS) or commercial-off-the shelf (COTS). Component - Based Software Development has evolved as a popular software development technique since the introduction of Microsoft’s Component Object Model (COM) in the early 90s. This paper presents review literature that has been published on empirical research of COTS. We were interested to describe the empirical research on COTS to see that if there are any areas of CBSD that are yet to be touched in the research process. Empirical studies are the proofs of the hypothesis about the industry’s perception of CBSD process.

Keyword- CBSD, Component object model (COM), COTS.

I. INTRODUCTION

We are in the 21st century presently and two decades will be finished soon. A century ago has been considered as the ascending of Information and correspondence innovation. We are witnessing an enormous expansion in the use of software in business, industry, administration, defense and research. Software is now become a central factor in many fields. System character based on software functionality, rather than other characteristics, are becoming the most important factor in competing on the market, for example in car industry, the service sector and in schools. These trends place new demands of software. Usability, functionality, robustness, simple installation and integration become the most important features of software. As a consequence of the wider area of software utilization, the need for the integration of different areas has increased. This requires established technique and tool support covering the entire component and system lifecycle including technological, organizational, marketing, legal, and other aspects. Component-based software engineering (CBSE) represents a new development paradigm: assembling software systems from components. This new research area has raised a significant amount of interests both in the research community and in the software industry-a rare phenomenon in the field of software engineering. This technique shift from building software from scratch to the engineering of assembling components to build software involves many issues and it demands more and more precision in every aspect to make this new paradigm as the most convenient way to resolve the software crisis [1][2][3]. CBSE attempts to answer the question Can a software be built by assembling components like assembling a bike? Following the success of the structured design and Object oriented technique, Component-Based Software Engineering (CBSE) has emerged as the next revolution in software development. CBSE emerge as a popular methodology in the early 90s and was expected to produce high quality software at reduced cost and lesser time as development of software need not be done from scratch. Reusability was considered as the backbone of CBSE. This paradigm started becoming popular after the introduction of middleware technology support like Microsoft‘s Component Object Model (COM), CORBA, and JavaBeans etc. Component based software engineering plays a vital role in increasing the productivity of an organization. There is a demand for rich set of components in the repository which can be reused. In most of the projects, once the requirements are collected, the development activity starts from scratch and this may lead to overtime and over budget anomaly. If the existing COTS component is reused rather than the developing the entire system from the scratch, not only the time is saved but quality product is produced [4].

In this paper, we wanted to review the literature that has been published on empirical research of CBSD since 1995 through March 2017. The goal of this research to see the areas of CBSD that is yet to be touched in the research process. We follow kitchenham’s guidelines to conducting a systematic literature review.

The paper is structured as follows: section 2 provides the background or related work of this study which covers an overview of the process CBSD/COTS, section 3 describes the methodology that we followed for this study at length, section 4 explains the results of our study, section 5 presents a analysis or discussion of the results, section 6 presents limitation of work, section 7 presents the conclusion and future work.

II. BACKGROUND

Spiral Models characteristics are involved in the Component Based software Development (CBSD) Model. The CBSD Model consists of the applications from the grouped software components (called Classes) stated by [5]. The component model started with the identification of the candidate components from repository and this is achieved by identifying the data to be changed by the application and the relevant algorithms that will be applied. The data and algorithms are encapsulated into the classes. The components created in the software projects are stored in component repository. Once requirement are identified, the repository is mined to check whether the desired components are present in the repository. If available, they are retrieved and are reused in the project development. If the component does not exist in the repository, it is engineered using the object-oriented methodology. The first iteration of the application to be build is composed of components retrieved from the repository and new components engineered to meet the requirement of the particular application. The process Flow is reversed back to the spiral model and will ultimately continue the component assembly looping during subsequent passes through the component life cycle as shown in figure 1. Software Reusability can be achieved by CBSD model which is highly useful to the Software Engineers. Yourdon .E in [6] shows about the usage of software reusability by QSM Associates Inc., and he reports component assembly moves to a reduction in software development life cycle, 84% reduction in project cost, and a productivity index of 26.2, compared to an industry norm of 16.9.With these results, the robustness of the component repository and CBSD model provides many advantages to the software engineers.

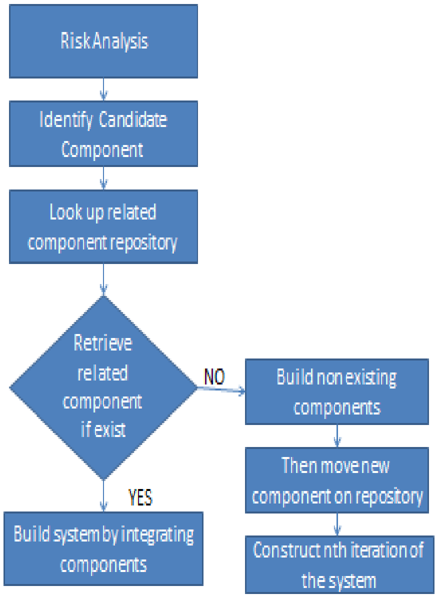
Risk Analysis
Identify Candidate Component
Look up related component repository
Retrieve related component if exist
NO
Build non existing components
Then move new component on repository
Construct nth iteration of the system
YES
Build system by integrating components

Figure1. Component Based Software Development Model

III. METHODOLOGY

This section describes the method we followed for our study in detail is taken from [7]. This section covers the research questions of our study, the search process we followed in order to accumulate the relevant literature, the selection criteria of the literature, quality assessment of the selected literature and finally the procedure we followed for extracting the data from the literature.

3.1 Research questions

The three research questions for our study are related to since 2017:

RQ1: What kind of empirical research has been done on CBSD?

RQ2: What research topics are being addressed in CBSD by these empirical studies?

Relative RQ1, we were interested in knowing the types of empirical studies that were conducted from 1995 to March 2017 with respect to CBSD or CBSE or Component Based Software (CBS).

Regarding RQ2, we are interesting to explore the areas concerning CBSD or CBSE or CBS that were covered by the empirical studies, for instance, the number of studies that were published on the issue of reusability of the components.

3.2 Search process

The keywords were – CBSE, Software development, organization, CBSD, component, component based software (CBS), empirical study, and empirical research.

The five databases used for our search are SpringerLink, ACM Digital Library, IEEE Xplore and Elsevier. Some important papers also included from other journal. These databases that contain much of the literature related to Software Engineering that was published by various international conferences, journals.

3.3 Study selection criteria

This section present in what follows is the criteria which we followed in including and excluding the studies for our study.

Inclusion criteria

- Number of Papers that were published from January 1995 to March 2017 was selected.
- Number of Papers that describe their study as an empirical study in their titles or has at least one type of empirical study as part of their study.
- Number of Studies that focus on aspects of CBSD, Component Based Software (CBS) or COTS was selected

•Number of Papers included the keywords ‘empirical’, ‘case study’ or ‘experiment’ relating to CBSD, CBS or COTS was selected.

Exclusion criteria

- position papers, view point and studies that are not empirical or do not contain an empirical study as part of their whole study were excluded
- Papers that are not related to CBSE or COTS or CBS were not included, for example, a paper discussing about reusability but not in regard to software components.
- Duplicate papers of the same study were excluded.

3.4 Procedure followed

We started searched for papers on CBSE without including the keyword ‘empirical’ or any of its synonyms in the search string. The total number of empirical studies on CBSE or COTS which includes literature reviews, opinion papers, position papers, etc. from the 4 databases which resulted in 366,723 papers. After applying selection criteria, we were left with total 40 studies.

3.5 Quality assessment criteria

Quality assessment criteria extracted from [8] and modified so as to make them adaptable for our study. The criteria which is based on three questions:

Q1 – Were the findings and analysis of CBSD clearly presented?
Q2 - Is the type of research method explained?
Q3 – Are the result justified by the data?

With respect to Q1 we find the results obtained from the study were clearly presented in terms of appropriateness. This means that if the study was qualitative one, then we considered the way the results were presented i.e. they may explanatory or not. If the study was quantitative one, then we looked out for the results numerically or statistically presented or not. For papers with results of complex type, we read and interpreted only certain parts of those results and assessed the quality, for example, papers with mathematical results or results with heavy usage of various mathematical formulae, etc.

With Q2 we find the result obtained from study was clearly explain in term of appropriateness. It mean if the study was stated to be qualitative one, then we considered result were presented but if the study was quantitative one, then we look out the result whether they are numerically or statistically presented or not.

With respect toQ3 it find the conclusions drawn from the study had a mapping to the data presented in the results section. This is done by reading the conclusion section of all the studies

- The kind of study that would be followed – whether it would be a qualitative or quantitative study
- Description of the study context – whether it could takes place in an industrial or in an academic environment.
- Description of the subjects involved in the study – whether humans or systems were the subjects.

IV. RESULT ANALYSIS

This section describes the result of the study. The result is presented in the form question.

4.1Q1 Kind of empirical studies

Q1 what types of empirical studies from 1995 to 2017 has been done with respect CBSD/COTS?

The answer of this question is figure 2.the most preferred methodology is experiment and case studies, they constitutes 55% and 35% of total number of studies (40) respectively. Survey constitutes 7% and correlation studies constitute 3% of the total number of studies.

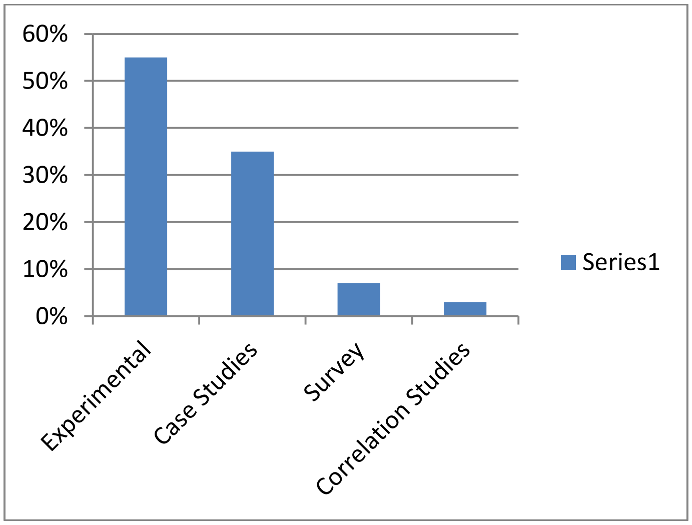

Figure2. Types of empirical study

4.2 Q2 Research topics being discussed

The result of Q2 i.e. what research topics are discussed by empirical studies?
The most discussed topics are reusability of components, selection of components and design and implementation of components.
Reusability of components constitutes (15%), selection of components constitutes (15%), reliability of components constitutes (14%) and design implementation constitutes (12%). The next most discussed topics are integration of components (10%), testing of components (9%), CBSD process (9%), maintenance of component (8%), and Cost reduction of components (3%) contribution.

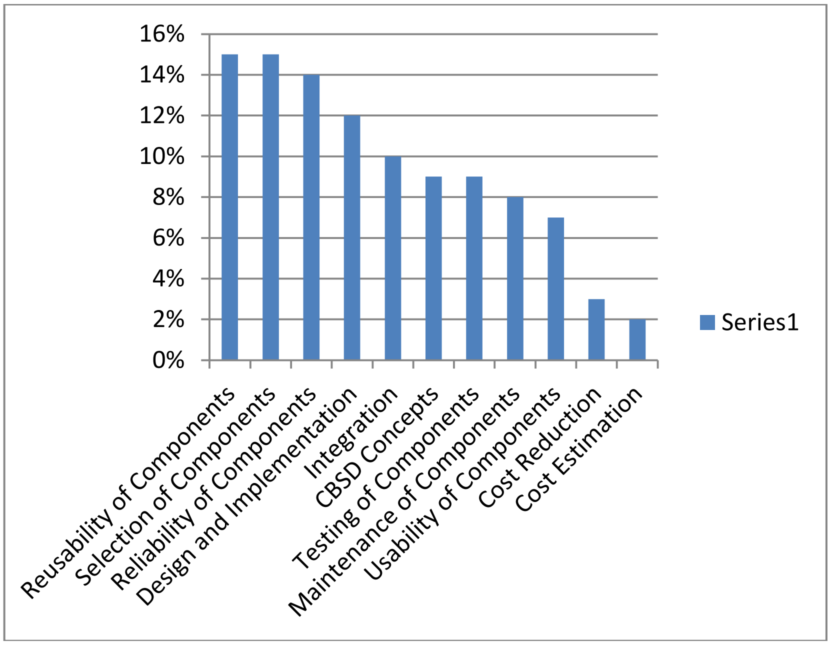

Figure3. Research topics being addressed

V. DISCUSSION OF RESULT

In this section, we discussed about research methodology that are most preferred in empirical research of CBSD/COTS.

5.1 Case studies

Table 1 present lists of all case studies that were conducted in the industry. This table presents the summary of the study and area of CBSD.

Table 1 Case Studies

Paper ID	Description	Research area of CBSD/COTS
P[9]	The object that needs to be managed in CBSD is analyzed and a component-based SCM is presented.	Management of components
P[10]	This paper presents a sample application of component- oriented programming concept for CBSD.CBSD process and some of the potential risk and challenge in CBSD are also presented	CBSD process
P[11]	This article presents the process of modeling control algorithms as means to increase reliability of software components. The propose approach developing EmbeddedControl Software (ECS) is tailored to Component- based Software Development (CBSD) and such tailoring allow reusing the ECS development process tools in a development process for robotics software.	Reliability of components
P [12]	This paper describes a pattern story that shows how a component-based design has been implemented using periodic	Design of components

	concurrent tasks with RT requirements.	
P[13]	Evaluates the reliability of component-based software using adaptive Neuro-Fuzzy inference system. This model considers the factors particular to component based software that affects its reliability and hybrid neural network used in ANFIS is trained using the data set obtained from a survey	Reliability assessment of component
P[14]	This article provide an overview of the actual state of the OSS marketplace and report preliminary findings about how companies interact with this marketplace to reuse OSS components and such data was gathered from interviews in software companies in Spain and Norway. These results identify some challenges aimed to improve the industrial reuse of OSS components.	Reuse of components
P[15]	This paper discussed an overview of a distributed architecture for the deployment of applications based on business components.	Design & implementation
P[16]	This paper suggests functional testing strategy and test case generation technique for component- based software. When two components are joined or integrated then they generate some specific effect. This strategy is called integration-effect graph.	Integration of components
P[17]	This article present a reusable component model-FLP model for reusable component which describes components from three dimensions form, level and presentation) views components and their relationships from the perspective of process and management.	Reusability of components
P[18]	This article present proposed model that determines the sphere of reusable components, the time points of reusing components in the development process and the needed means to present components in terms of the abstraction level, logic granularity and presentation media.	Reusability of components
P[19]	It proposes a component framework for supporting the architecture-based design and development of self –adaptive application.	Design of components
P[20]	This paper present using CAD tools that support an ideal separation between component, system development, experiments were conducted to	Testing of components

	investigate two related question: (1) To what extent can unit (component) testing replace system testing? (2) What properties of software influence the quality of sub domain testing?	
P[21]	The component-based software development (CBSD) to identify Design and to build the components and interface of the system that provides and requires services. The proposal took a study case based on development of an academic management system basic module with a persistence model dependent a distributed database system.	Application of components
P[22]	This article presents a model for estimating CBSS reliability, known as an Adaptive Neuro Fuzzy Inference System (ANFIS) that is based on these two basic elements of soft computing and we compare its performance with that of a plain FIS (Fuzzy Inference System) for different data sets.	Reliability of components
P[23]	This approach is validated with the reporting service of a document exchange server, by modeling the reliability, conducting a reliability prediction and sensitivity analyses and demonstrating its ability to support design decision.	Reliability of components
P[24]	A study of the safety properties of HTS using a calculus of component composition has provided solid foundations for the design of configuration languages for the safe specification and deployment of parallel components	Performance of components
P[25]	It presents a self- adaptive software architecture (model-based) approach for supporting seamless integration of COTS components	Integration of components
P[26]	It present a component management system to manage distributed software components for large- scale software system development and component repository was introduced as the basic logic unit for storage and management of the software components in accordance with development stage.	Management of component

Commercial Off-The-Shelf (COTS) software components From the table1 it is clear that much of research that has been done focused on reusability of component, selection of components and design of components. This gives a direction that other area such quality, testing and storage of components which are vital in the process of CBSD were less researched area.

5.2 Experiments: A Research Methodology

Table 2 Presents list of all 23 experiments that were conducted with respected to the CBSD approach. Studies of CBSD are evaluated by full text reading.

Table2. Experiment in the Empirical Research of CBSD/COTS

Paper ID	Description	Research area of CBSD/COTS
P[27]	This article describes the use of CAD tools that support an ideal separation between component and system development. Experiments were conducted to investigate two related question: (1) To what extent can unit testing replace system testing? (2) What properties of software influence the quality of sub domain testing?	Testing of components
P[28]	This article describe the CBSD theory and tools, lists insight gained and it suggest new ways to think about testing using test –based specifications.	Testing of components
P[29]	This paper presents the application of the CBSE paradigm for the development of a university ERP- specifically an e-Administration System and the result of the case study yielded a usable ERP for a Nigerian university and concrete empirical data confirmed the superiority of CBSE over traditional software development.	Application of components
P[30]	This paper describes the effects of AOP (Aspect – Oriented Programming) on the maintainability of two COTS based system: Openbravo POS and Jasper report. The effects were measured using the maintainability metrics of the ISO/IEC 9126 model.	Maintainability of components
P[31]	This paper presents result of an empirical study concluded to quantify the implementation, testing and knowledge requirement costs of building a self – adaptive software system using control engineering methods. Our objective is to find, whether these costs can be significantly reduced if a library of pre - packaged control components is available to software engineers and the findings of the study indicate that the aforementioned costs can be significantly reduced when supporting libraries are available.	Cost reduction using components
P[32]	This article can develops a theoretical framework based on formally certified semantic preserving graph- theoretic transformations that allows us to associate to each concrete component	Maintainability of components

	repository a much smaller one with a simpler structure so that we call strongly flat, with equivalent co-installability properties.	
P[33]	The objective of this paper is to describe the characteristics of some selected state of the art CBSD models and a new reusable software process model has been designed for the optimal selection of components based on the new optimal algorithm	Selection of components
P[34]	This paper presents a methodology to decouple the tasks of component and application developers, who will be able to share information asynchronously as well as independently and communicate implicitly by developing and deploying what we call healing connectors.	Selection of components
P[35]	The objective of this paper is to select the suitable mix of components using build-or-buy strategy or considering fabrication and to propose a multi-objective model for software modular system with objective of maximizing reliability while simultaneously minimizing the cost.	Selection of components
P[36]	This paper presents a system development life cycle model which has incorporated most importantly customer participation, customer support, customer feedback analysis and new component development phases by providing fully complete customer requirement oriented framework. The propose model for CBSD named as Customer Requirement Oriented CBSD life cycle Model(CROM)	CBSD concept and propose model
P[37]	It introduces concepts and mechanism that allow to model security specifications and derive automatically corresponding security implementations by transforming the original component model into a secured one taking into account sensitive data flow in the system and the resulted architecture ensures security requirements by construction and is expressed in the original meta model form.	Application of components
P[38]	This paper present V& V when developing software components which are helpful in improving the functionality and quality of component and Component-Based System (CBS) by using a new X Component-Based Model.	CBSD proposed model

P[39]	It introduces a new reliability analysis technique applicable to high- level design. The technique is named Reliability Analysis Based on Rewrite Logic (RABRL). RABRL is specific for component-based software whose analysis is strictly based on its operational profile and specification.	Reliability of components
P[40]	This paper proposes metrics for measuring the complexity, customizability and reusability of software components.	Quality of components
P[41]	For a component-based approach, two steps may be used to estimate the overall size of object-oriented(OO) software: a designer uses metrics to predict the size of the software components and then ultimate the sizes to estimate the overall project size.	Cost estimation of components
P[42]	The propose optimization model is presented to solve component selection problem considering reusability and compatibility simultaneously. This model can be used to assist software developers in selecting software components when multi-application are undertaken concurrently. Four experiments are conducted with the purpose to provide some insights in management perspective.	Selection of components
P[43]	This paper describe the characteristics of some selected state of art CBSD models and a new reusable software process model has been designed for the optimal selection of components based on the new optimal algorithm.	Selection of components
P[44]	This paper describes a set of measure to assess the usability of software components and it also describes the method followed to obtain and validate them.	Usability of components
P[45]	This paper describes a function point-like measure named Component Point (CP). For measuring the system level size of a CBSS specified in the Unified Modelling Language. The proposed approach integrates the three software measure and extends an existing size measure from the more matured Object Oriented paradigm to the related and relatively young CBSS discipline.	Cost estimation of components
P[46]	It evaluates the impact of three popular CBAs, namely Enterprise Java Beans,	Reusability of components

	Distributed Inter Network Architecture and object Management Architecture on reusability.	
P[47]	This paper discusses how an evidence-based approach to component evaluation can improve repeatability and reproducibility of component selection under the following conditions: (1) functional homogeneity of candidate component (2) high number of components and selection problem instances.	Selection of components
P[48]	It presents a systematic approach that makes it possible to derive architectural models with structural descriptions and behavior from Teleo – Reactive Programs and therefore benefits significantly from a combination of two approaches.	Reusability of components
P[49]	This article formalizes an evolution management model that generates evolution plans according to a given architecture change request thus preserving consistency of architecture description and coherence between them.	Management of components
P[50]	This paper describes a set of measures to assess the usability of software components and describes the method followed to obtain and validate them.	Usability of components

From table 2 it is clear that most covered area of experiments is selection of components, implementation of component, and reusability of components being the most researched area. Other areas are CBSD process, management of components, and design of components as well as performance of components.

VI. LIMITATIONS OF THE STUDY

We basically review articles of four databases, also some important article collected from some others journal from January 1995 to march 2017 based on COTS or CBSD or CBSE. But there are other good journals which are not used in this study. This is one of the limitations of our work. Other limitation may be partially based papers on CBSD/COTS are not considered.

VII. CONCLUSION AND FUTURE WORK

We describe the systematic review of article from January 1995 to March 2017. This study shows almost all areas of CBSD are covered. We found 40 studies of empirical research on CBSD. Most discussed research topics were:

- Selection of component.
- Reusability of component
- Quality of component through different model
- Implementation of component

Apart from that, we found certain topics that should be focused in future.

- There is a need of standard quality model to build COTS component.
- No single algorithm enough to find most appropriate component from repository.
- The main challenge of COTS is achieving a best conversion procedure from requirements to components and then from components to system.

The future work of this study could be possible by adding some new rule in the method section and it could help in extracting more analytical information.

References

[1] Gao, J., Tsao, H. S., Wu, Y., "Testing and quality assurance for component-based software", Artech House, 2003.

[2] Bhatt, P., Thaker, B., Shah, N., "A Survey on Developing Secure IOT Products", International Journal of Scientific Research in Computer Sciences and Engineering, Vol. 6(5), pp.41-44, 2018.

[3] Singha, P., Aditya, "Toolkit for Web Development Based on Web Based Information System", International Journal of Scientific Research in Computer Sciences and Engineering, Vol. 6(5), pp.1-5, 2018.

[4] Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., Linkman, S., "Systematic literature reviews in software engineering–a systematic literature review: Information and software technology", Vol. 51(1), pp.7-15, 2009.

[5] Clemens, S., D.G.S.M., "Component Software: Beyond Object-Oriented Programming", 2nd ed. London: Addison-Wesley and ACM Press, 2002.

[6] Crnkovic, I., M. Chaudron, S. Larsson., "Component-Based Development Process and Component Lifecycle", In International Conference on Software Engineering Advances (2006).

[7] Tekumalla, B., "Status of Empirical Research in Component Based Software Engineering-A Systematic Literature Review of empirical studies", 2012.

[8] Ajila, S.A., D., Wu, "Empirical study of the effects of open source adoption on software development economics", Journal of Systems and Software, Vol. 80(9): pp. 1517-1529, 2007.

[9] Höfer, A., W. Tichy, "Status of Empirical Research in Software Engineering Empirical Software Engineering Issues", Critical Assessment and Future Directions, V. Basili, et al., Edit, Springer Berlin / Heidelberg, pp. 10-19, 2007.

[10] Mei, H., Zhang, L., Yang, F., "A component-based software configuration management model and its supporting system", Journal of Computer Science and Technology, Vol. 17(4), pp.432-441, 2002.

[11] Shukla, R., Marwala, T., "Component Based Software Development Using Component Oriented Programming", In Proceedings of International Conference on Advances in Computing, Springer India, pp. 1125-1133, 2013.

[12] Brodskiy, Y., Wilterdink, R., Stramigioli, S., Broenink, J., "Fault avoidance in development of robot motion-control software by modeling the computation", In International Conference on Simulation, Modeling, and Programming for Autonomous Robots, Springer International Publishing, pp. 158-169, 2014.

[13] Pastor, J. A., Alonso, D., Sánchez, P., Álvarez, B., "Towards the definition of a pattern sequence for real-time applications using a model-driven engineering approach", In International Conference on Reliable Software Technologies, Springer Berlin Heidelberg, pp. 167-180, 2010.

[14] Dubey, S. K., Jasra, B., "Reliability assessment of component based software systems using fuzzy and ANFIS techniques", International Journal of System Assurance Engineering and Management, pp.1-8, 2017.

[15] Ayala, C., Hauge, Conradi, R., Franch, X., Li, J., Velle, K. S.,"Challenges of the open source component marketplace in the industry", In IFIP International Conference on Open Source Systems, Springer Berlin Heidelberg, pp. 213-224, 2009.

[16] Jain, H., Reddy, B., "Layered architecture for assembling business applications from distributed components", Journal of Systems Science and Systems Engineering, 13(1), pp.60-77, (2004).

[17] Tiwari, U. K., Kumar, S., "Components integration-effect graph: a black box testing and test case generation technique for component-based software", International Journal of System Assurance Engineering and Management, pp.1-15, 2016.

[18] Mei, H., "A component model for perspective management of enterprise software reuse", Annals of Software Engineering, Vol. 11(1), pp.219-236, (2001).

[19] Brown, A. W., "Model driven architecture: Principles and practice", Software and Systems Modeling, 3(4), pp. 314-327, 2004.

[20] Tyagi, K., Sharma, A., "An adaptive neuro fuzzy model for estimating the reliability of component-based software systems", applied Computing and informatics, Vol. 10(1), pp.38-51, 2014.

[21] Maldonado, C. D. A., Nieto, L. A. C., Cala, D. S. P., Diosa, H. A., "Methodological hybrid SOA+ CBSD for services oriented software development", 10th Computing Colombian Conference (10CCC), pp. 86-92, IEEE, 2015.

[22] Ulkuniemi, P., Araujo, L., Tähtinen, J., "Purchasing as market-shaping: The case of component-based software engineering", Industrial Marketing Management, Vol. 44, pp.54-62, 2015.

[23] Pham, T. T., Defago, X., "Reliability prediction for component-based software systems with architectural-level fault tolerance mechanisms", Eighth International Conference on Availability, Reliability and Security (ARES), 2013, pp. 11-20, IEEE, 2013.

[24] de Carvalho Junior, F. H., Rezende, C. A., deCarvalho Silva, J., Al-Alam, W. G., de Alencar, J. M. U., "Contextual abstraction in a type system for component-based high performance computing platforms", Science of Computer Programming, 132, pp.96-128, (2016).

[25] Tang, S., Liu, Q., "Supporting Integration of COTS Components from a Perspective of Self-Adaptive Software Architecture", In Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual, pp. 706-713, IEEE, (July, 2013).

[26] Li, Y., Yin, J., Dong, J., "A component management system for mass customization", First International Multi-Symposiums on Computer and Computational Sciences, IMSCCS'06, Vol. 2,pp. 398-404, IEEE, 2006.

[27] Yu, Z., Xiaoxing, M., Xianping, T., Jian, L., "Towards a component framework for architecture-based self-adaptive applications", Wuhan University Journal of Natural Sciences, 11(5), pp.1227-1232, 2006.

[28] Hamlet, D.., "Test-Based Specifications of Components and Systems", Seventh International Conference on Quality Software, QSIC'07, Seventh, pp. 388-395, IEEE, 2007.

[29] Okewu, E., Daramola, O., "Component-based software engineering approach to development of a university e-administration system", 6th International Conference on Adaptive Science & Technology (ICAST), pp. 1-8., IEEE, 2014.

[30] Mguni, K., Ayalew, Y., "Improving maintainability of cots based system using aspect oriented programming: An empirical evaluation", African Conference on Software Engineering and Applied Computing (ACSEAC), pp. 21-28, IEEE, 2012.

[31] Patikirikorala, T., Colman, A., Han, J.., "Can control-component libraries reduce the costs of developing control engineering-based self-adaptive systems", 20th Asia-Pacific on Software Engineering Conference (APSEC), 2013 Vol. 1, pp. 42-49, IEEE, 2013.

[32] Vouillon, J., Cosmo, R. D., "On software component co-installability", ACM Transactions on Software Engineering and Methodology (TOSEM), Vol. 22(4), pp. 34, 2013.

[33] Kumar, A., "A Design Based New Reusable Software Process Model for Component Based Development Environment", Procedia Computer Science, Vol. 85, pp. 922-928, 2016.

[34] Chang, H., Mariani, L., Pezze, M., "Exception handlers for healing component-based systems", ACM Transactions on Software Engineering and Methodology (TOSEM), Vol. 22(4), pp. 30, 2013.

[35] Kaur, R., Arora, S., Jha, P. C., Madan, S., "Fuzzy Multi-criteria Approach for Component Selection of Fault Tolerant Software System under Consensus Recovery Block Scheme", Procedia Computer Science, 45, pp.842-851, 2015.

[36] Talukder, S. C., Rahman, M. M., "Customer requirements oriented component based software development life cycle model", International Conference on Computers, Communications, and Systems (ICCCS), pp. 61-68, IEEE, 2015.

[37] Saadatmand, M., Leveque, T., "Modeling security aspects in distributed real-time component-based embedded systems", Ninth

International Conference on New Generations (ITNG), pp. 437-444, IEEE, 2012.

[38] Tomar, P., Gill, N. S., “Verification & Validation of Components with New X Component-Based Model”, 2nd International Conference on Software Technology and Engineering (ICSTE), Vol. 2, pp. V2-365, IEEE, 2010.

[39] Wang, D., Huang, N., “Reliability analysis of component-based software based on rewrite logic”, 12th IEEE International Workshop on Future Trends of Distributed Computing Systems, FTDCS'08, pp. 126-132, IEEE, 2008.

[40] Cho, E. S., Kim, M. S., Kim, S. D., “Component metrics to measure component quality”, Eighth Asia-Pacific Software Engineering Conference, APSEC 2001, pp. 419-426, IEEE, 2001.

[41] Pendharkar, P. C.., “An exploratory study of object-oriented software component size determinants and the application of regression tree forecasting models”, Information & management, Vol. 42(1), pp.61-73, 2004.

[42] Tang, J. F., Mu, L. F., Kwong, C. K., Luo, X. G., “An optimization model for software component selection under multiple applications development”, European Journal of Operational Research, 212(2), pp.301-311, 2011.

[43] Kumar, A., “A Design Based New Reusable Software Process Model for Component Based Development Environment”, Procedia Computer Science, 85, pp. 922-928, 2016.

[44] Bertoa, M. F., Troya, J. M., Vallecillo, A., “Measuring the usability of software components”, Journal of Systems and Software, Vol. 79(3), pp.427-439, 2006.

[45] Wijayasiriwardhane, T., Lai, R., “Component Point: A system-level size measure for component-based software systems”, Journal of Systems and Software, Vol. 83(12), pp.2456-2470, 2010.

[46] McArthur, K., Saiedian, H., Zand, M., “An evaluation of the impact of component-based architectures on software reusability”, Information and Software Technology, Vol. 44(6), pp.351-359, 2002.

[47] Becker, C., Rauber, A., “Improving component selection and monitoring with controlled experimentation and automated measurements”, Information and Software Technology, 52(6), pp.641-655, 2010.

[48] SáNchez, P., Alonso, D., Morales, J. M., Navarro, P. J.., “From Teleo-Reactive specifications to architectural components A model-driven approach”, Journal of Systems and Software, Vol. 85(11), pp.2504-2518, 2012.

[49] Hjertström, A., Nyström, D., Sjödin, M., “Data management for component-based embedded real-time systems: the database proxy approach”, Journal of Systems and Software, Vol. 85(4), pp.821-834, 2012.

[50] Bertoa, M. F., Vallecillo, A.,“Usability measures for software components”, IEEE Latin America Transactions, Vol. 4(2), pp.136-143, 2006.