FUZZY BASED COMPONENT REUSABILITY EVALUATION APPROACH TO SUPPORT COMPONENT BASED SOFTWARE DEVELOPMENT

SURYANI ISMAIL

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy (Computer Science)

Faculty of Computing
Universiti Teknologi Malaysia

JULY 2017

ALHAMDULILLAH…..

Thankfully.....

BLESSINGS AND GREETINGS TO OUR Great Prophet…

Dedicated to my beloved family in memoriam,
My husband, Zainal bin Selamat
My sons Ahmad Hasanuddin, Muhammad Hakimi, and Muhammad Muttaqin
My daughters Nur Aufa and Nurul Najibah

Thanks for giving me the strength and courage.

ACKNOWLEDGEMENT

In the Name of Allah, the Most Gracious, the Most Merciful, peace and blessings be upon our Prophet Muhammad's (pbuh), his family and his companions.

First and foremost, I would like to thank and express my gratitude to my supervisor, Prof. Madya Dr. Wan Mohd Nasir bin Wan Kadir and co supervisor Prof.Madya Dr. Noor Maizura binti Mohamad Noor for their assistance, guidance, advice and ideas given throughout and up to the completion of my thesis. Also, special thanks to Prof. Dr. Mohd Yazid bin Md Saman for his contributions in terms of knowledge and expertise related to this research. May God bless both of you and I will never forget your kindness for life.

Sincere appreciation to the Ministry of Higher Education (MOHE), which has funded this study, in Faculty of Computing (FC), UTM, where it has helped me throughout this study.

Thank you to all the friends, particularly in the School of Informatic and Applied Mathematic (SIAM), and UMT has much to contribute their ideas and criticism throughout this learning process.

Thanks to my beloved husband, Zainal bin Selamat and children for their support, encouragement, trust, and care during the process of completing this thesis. Hopefully everything will get rewarded by Allah. Aamiin.

ABSTRACT

One of the contributions of Component Based Software Development (CBSD) is the reuse of software components across multiple systems by software developers. However, the developers often face a difficulty to determine the reusability of the components during the component selection process. Similarly, the component developers also have a problem to measure the component reusability during component development. Nowadays, even though many studies have been conducted in this field, which the researchers suggested many approaches with metrics but they still lack in empirical confirmation and evidences. Therefore, the aim of this study is to investigate and develop the component reusability evaluation approach to support CBSD. The proposed approach, which is called Component Reusability Evaluation Approach (CREA), is supported by the developed automated tool (CREATool) that may automate the reusability evaluation. CREA is then evaluated by applying five Java component in this approach and CREATool to the selected software components. The results from the application approach and then validated with results from the controlled experiment using statistical analysis. The results indicated that CREA able to provide an acceptable reusability measure, which it is confirmed by similarity results between evaluation using statistical analysis through the controlled experiment and by applying the CREATool. It shows that the proposed approach could be used as an alternative approach in component reusability evaluation. Although the developed approach are not intended to make a holistic and an ultimate decision whether the components can be reused or not, but it is useful enough to be considered as a guide for both component users and developers in making decisions related to reusable components

ABSTRAK

Salah satu sumbangan pembangunan perisian berasaskan komponen (CBSD) adalah penggunaan semula komponen perisian merentas pelbagai sistem oleh pembangun perisian. Walau bagaimanapun, pembangun perisian sering menghadapi kesukaran untuk menentukan kebolehgunaan semula komponen semasa proses pemilihan komponen. Selain itu, pembangun komponen juga mempunyai masalah untuk mengukur kebergantungan komponen semasa pembangunan semula komponen. Pada masa kini, walaupun terdapat banyak kajian telah dijalankan dalam bidang ini oleh penyelidik yang telah mencadangkan banyak pendekatan dengan pelbagai jenis metrik, kajian ini tidak mempunyai bukti dalam pengesahan empirikal. Oleh itu, kajian ini mengkaji dan membangunkan pendekatan penilaian kebolehgunaan semula komponen (CREA) dalam menyokong CBSD. Pendekatan yang dikenali sebagai penilaian kebolehgunaan semula komponen (CREA) disokong oleh peralatan automatik yang dibangunkan. Dalam kajian ini, CREATool telah dibangunkan untuk kajian yang boleh mengautomasikan penilaian kebolehgunaan. CREA telah dinilai dengan menggunakannya kepada komponen Java yang dipilih menggunakan CREATool yang dibangunkan. Hasil daripada eksperimen ini selanjutnya disahkan dengan menggunakan eksperimen terkawal berdasarkan analisa statistik. Berdasarkan ketekalan hasil daripada kedua-dua eksperimen, keputusan menunjukkan bahawa CREA menghasilkan ukuran kebolehgunaan semula yang dapat diterima. Oleh itu, berdasarkan daripada keputusan CREA boleh dianggap sebagai pendekatan alternatif dalam penilaian kebolehgunaan semula komponen. Pendekatan yang dibangunkan tidak dapat membuat keputusan yang holistik dan keputusan muktamad untuk pemilihan komponen sama ada boleh digunakan semula atau tidak, tetapi ia boleh menjadi panduan bagi pengguna komponen dan pemaju dalam membuat keputusan yang berkaitan dengan komponen yang boleh diguna semula.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AACD	-	Average Active Component Density
ACD	-	Average Component Density
ANAC	-	Average Number Active Components
CAID	-	Component Average Interaction Density
CB	-	Component Based
CBD	-	Component Based Development
CBO	-	Coupling Between Objects
CBSD	-	Component Based Software Development
CBSD	-	Component Based Software Engineering
CCC	-	Component Cyclomatic Complexity
CDC	-	Component Dynamic Complexity
CID	-	Component Interaction Density
COTS	-	Common-off-the-Shelf
CPC	-	Component Plain Complexity
CPD	-	Component Packing Density
$\mathrm{CRIT}_{\mathrm{ALL}}$	-	Tall Criticality Metric
$\mathrm{CRIT}_{\mathrm{BRIDGE}}$	-	Bridge Criticality Metric
$\mathrm{CRIT}_{\mathrm{INHERITANCE}}$	-	Inheritance Criticality Metric
$\mathrm{CRIT}_{\mathrm{LINK}}$	-	Link Criticality Metric
$\mathrm{CRIT}_{\mathrm{SIZE}}$	-	Size Criticality Metric
CSC	-	Component Static Complexity
DIT	-	Depth Of Inheritance Tree
FCM	-	Factor-Criteria-Metrics
IDE	-	Integrated Development Environment
LCOM	-	Lack Of Cohesion Method
LOC	-	Line of Code

NOC	-	Number Of Child
OC	-	Original Component
PLC	-	Product Line Component
QC	-	Quality Component
RC	-	Reusable Component
REBOOT	-	Reuse Based Object Oriented Technology
RFC	-	Response For Class
SIAM	-	School Of Informaticsapplied
UMT	-	University Malaysia Terengganu
WMC	-	Weight Method Per Class

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This chapter provides an introduction to the research work presented in this thesis. It describes the research background and reviews numerous important related aspects of this research. In the first part, it explains the problem background followed by the problem statements and objectives of the study. Next, it describes the scope of the study and explains the importance of the study. Finally, it presents the outline of the thesis.

1.1 Research Background

In this section, issues related to the main areas of the research are presented. They are categorized into three parts, namely software reuse and component reuse, component reuse problem and research direction for component reuse. The category will be explained in the following sections.

1.1.1 Software Component Reuse

In software engineering, the trend is changing from the traditional software development approach to the extension and integration with existing systems [3, 4]. An ideal software component reuse technology would enable software developers to quickly use and adapt components in software development. In addition, this technology could be used in all application domains, to reduce the time and effort required to build and maintain software systems and to enhance the quality of software systems by reusing quality and reusable software components.

Software component reuse is considered as an important solution to many software engineering problems. It has been claimed to improve the productivity and the quality of software development [5, 7]. Many organizations have benefited from using reusable components in reducing the time and cost of software development [8-10].

From exhaustive study in software component reuse, it can be concluded that software component reuse is one of the important factors in facilitating software reuse in new software development. Software become hard to be developed, understood, managed, controlled and maintained without any software component reuse. Software component reuse plays an important role as it always keeps track of the relationships among the artifacts to help developers or system analysts in performing their tasks. It helps ensuring that software development time could be cut down and upon a change has been made and all of impacted components have been reused, plug in into the new system and tested effectively. Finally, the new software development cannot be developed from scratch.

1.1.2 Software Component Reuse Problem

Among the problems faced by software engineers in component reuse is the difficulty to determine which set of components are suitable to use in new software development. Problem of feature and component selection, if these are given a set of such components, it is hard to determine a subset that it is minimize the risk and maximize the commercial return

In addition, third-party users always face the problem on how to test software component when its source code is unavailable [12].

1.1.3 Directions for Software Component Reuse

Software component reuse is considered as an important solution to many software engineering problems. It is claimed to help improve the productivity and quality of software development [4, 6, 7]. Many organizations have benefited from using reusable components because it can reduce the time and cost of software development [5, 11, 13]. Deng [14], suggested to reuse the component in software development when the components that will be used exactly fit the need of the software developer. It can be used without any modification or learn how to use it [14].

Since software reuse is widely accepted as a solution to improve the quality of both software products and processes, there are many research efforts devoted to this area [3, 15, 16]. Such examples are the management of reusable component reuses metrics, and the composition of reusable artifacts.

However, regarding software development using components based approach, it requires more effort by the developers to reuse the component in new systems developments. It needs the study of component evaluation approach that may assist software engineers in developing and measuring their own reusable components [17]. On the other hand, using this evaluation approach; it is easier for software developers to determine which set of components that are useful to produce new quality software in their future software project development.

Many component evaluation approaches that have been proposed such as product line component approach [16-18], original component approach [17], quality component approach [3, 18, 19] and reusability component approach [20, 21]. Most of components evaluation approaches focus only on some limited aspects in evaluating the components**.** The aspects only cover for components with reuse such as a black box components that the source codes of the components are unavailable. However, in order to support software component evaluation, a component evaluation approach must take as much as possible important influencing aspects that cover components with reuse and components for reuse in component evaluation. This study focuses on component reusability evaluation of components for reuse. This approach has many characteristics that minimizes software development time, effort and cost in the development new systems, such as portability, adaptability, understandability and confidence [20, 21, 106].

1.2 Statement of the Problem

This study covers software reuse in general and software component reuse in specific. Shambhu and Mishra [71] stated that software component reuse helps reducing production cost and time in a new software development. Component Based Software Development (CBSD) is one of the techniques used by researchers and practitioners to improve the quality of software systems with lower cost and shorter time to market, where it uses existing reusable components instead of writing from scratch [72].

It posits that utilizing a reusable software components reusability evaluation approach to provide significant support for facilitating component for reuse in CBSD. There are many characteristics of component reusability such as portability, adaptability/legibility, understandability and confidence that are mentioned by previous researchers [20, 21, 106]. The metrics of component reusability were proposed based on these characteristics and sub characteristics.

From the study on the reusability component evaluation approach, component evaluation is very important in order to select the suitable component for new software development. Although, there are many software component reusability evaluation approaches were proposed, but most of them lack empirical validation [80]. The hypothesis leads to the following research questions.

The main research question is "*How to evaluate component reusability in component reuse for Component Based Software Development (CBSD)?*"
The sub-questions about the main research question are as follows:

(i) What is the gap in the current component reusability evaluation approaches in CBSD?

(ii) How to define the characteristics and sub characteristics of software component reusability evaluation for CBSD?

(iii) How to develop a set of metrics suite of software component reusability evaluation approach for CBSD?

(iv) To evaluate and validate the reliability of the proposed approach using controlled experiments and appropriate supporting tools for CBSD?

1.3 Objectives of the Study

Based on the problem statements mentioned above, this research encompasses a set of objectives of this research as follows:

(i) To investigate and identify the gap in the current component reusability evaluation approaches in CBSD.

(ii) To define the characteristics and sub characteristics of software component reusability evaluation for CBSD.

(iii) To develop a set of metrics suite of software component reusability evaluation approach for CBSD.

(iv) To evaluate and validate the reliability of the proposed approach using controlled experiments and appropriate supporting tools for CBSD.

1.4 Scope of the Study

This research developed a set of metrics suite for software component reusability evaluation approach to support software component for reuse in CBSD. In this study only Java components are considered. The evaluation of the component reusability was done using the development of the tool proposed approach. However, this research will only focus on the components for reuse in the software development.

Although numerous component evaluation approaches that supports many software component reuse activities are available, there is a lack of metrics suites for component reusability evaluation. The theory of fuzzy logic that is used in this study was coined by Lotfi A. Zadeh, Professor at the University of California, Berkley [67]. This research proposed a component reusability evaluation approach employing fuzzy rules to calculate the reusability level of the components. The proposed approach focuses on the following aspects:

(i) Selecting the Java components that can be used in the development of a software.

(ii) Evaluating the reusable components based on reusability level for those components.

(iii) Adopting the reusable components in new software development that will cover the process to select, reuse, evaluate and adopt the components that can be deployed in the development of the new software [22].

In this research, a supporting tool has been created to implement the approach and to simplify the component reusability evaluation. The tool will be integrated with a specific ready-to-use Integrated Development Application (IDE) in order to select the reusable components for software development processes.

In this study, four sample Java component/packages have been used as input for the evaluation subject, namely BookPackage, FruitPackage, GeogmetricPackage and PersonPackage. Every package was used as a subject for metrics calculation using proposed metrics and the developed tool. These packages were also used as a reuse component in simple Java programming development coupled with that a set of questionnaire that need to be answered by target users. Software component model for this study is waterfall software life cycle model. Software reuse can be applied at any stage in software development processes. The reusable software artifacts such as requirement documents, system specification, design patterns, software unit, test cases and development artifacts can potentially be reused at the different stages.

1.5 Significance of the Study

Software component reuse is about reusing existing components in software development rather than developing a software from scratch. Many literatures point out that there is a close relation between software component reuse and software development where in software component reuse the software developer develop software using existing reusable component, instead of writing the coding from the scratch [9, 13, 15].

Software component reuse plays an important role to support software in CBSD because it can reduce cost and time during the lifecycle of a software development. Thus, finding the approach for evaluating component reusability among components that takes multiple aspects into consideration is very important in order to simplify and minimize software development effort.

A more flexible and natural way of evaluating software component reusability is by employing everyday natural human language. Thus by incorporating fuzzy logic technique into software component evaluation, it will make the process become more flexible and friendly to user.

1.6 Thesis Organization

This thesis is divided into seven chapters. Chapter 1 contains the overview of the study, problem statement, objectives and scope and significance of the study and finally, outline of the thesis. Chapter 2 to Chapter 7 are organized as follows:

Chapter 2: reviews the literature of software reuse and component reuse. It presents the definitions of software reuse as well as discuss issues related to software component reused as pointed out by some researchers. It is followed by component reuse that will be used as a tool to compare some existing evaluation approaches. It also reviews four existing component evaluation approaches, then explains comparative approaches for various component evaluations. The comparative approaches were focused primarily on their capability to support software component evaluation. In addition, comparison of various metric suites was also performed. This chapter is concluded with a discussion based on the evaluation results and a summary of the approaches.

Chapter 3: discussions on the research methodology that describes the research design and formulation of the research problems and validation considerations. This chapter also presented procedures that were carried out in this research. It also describes the experiments that were conducted to evaluate the development tool as well as describing the validation process for this research. Lastly, it explains some assumptions and limitations of this research.

Chapter 4: presents the details of the proposed component reusability evaluation approach. It includes the proposed model for component reusability evaluation approach and its rationale. Next, it describes the conceptual framework of this study followed by explanations of the proposed approach.

Chapter 5: explains the design and functionality of Component Reusable Evaluation Approach (CREA) tools as well as its accomplishment. This chapter describes the design of the tool, the user interface, and the implementation.

Chapter 6: Provides evaluations and validation of CREA. The evaluations focused on component reusable metrics, the methods used and the experiments that had been conducted. Research findings based on the results of the analysis are provided at the end of this chapter.

Chapter 7: Presents the achievements of the research objectives, the research contributions, recommendations and future work of this study.

REFERENCES

1. Hong, S., and Koelzer, B. (1995). A Comparison of Software Reuse Support in Object Oriented Methodologies. *the 30th International Symposium on Computer Architecture.*
2. Basili, V. R., Briand, L. C., and Melo, W. L. (1996). *How reuse influences productivity in object-oriented system.* Communications of ACM, 39(10): p. 104-116.
3. Alvaro, A., E.S. Almeida, and S.R.L. Meira. (2005). Quality Attributes for a Component Quality Model. in *10th International Workshop on Component-Oriented Programming (WCOP) in Conjuction with ECOOP*. Glasgow, Scotland.
4. Hooper, J.W. and R.O. Chester. (1991). *Software Reuse: Guidelines and Methods*, ed. R.A.D. (Ed). New York: Plenum Press.
5. Poulin, J.S., J.M. Caruso, and D.R. Hancock. (1993). The business case for software reuse. *IBM Systems Journal*. 32(4): 567 - 594.
6. Prieto-Díaz, R. (1991). Implementing faceted classification for software reuse. *Communications of the ACM*. 34(5).
7. Schäfer, W., R. Prieto-Díaz, and M. Matsumoto.1(994). *Software Reusability*, (Eds). Ellis Horwood.
8. Tracz., W. (1988). Software Reuse : Motivators and Inhibitors In: Tutorial: Software Reuse:Emerging Technology, Washington:. *IEEE Computer Society*. 62-67.
9. Kruger, C.W. (1992). Software Reuse. *ACM Computing Surveys*, 24(2): 132-138.
10. Lung, C.-H. (1994). *An Analogy-based Domain Analysis Mehodology*. in Thesis (PhD). Arizona State University.

11. Harman, M., A. Skaliotis, and K. Steinh. (2006). Search-Based Approaches to the Component Selection and Prioritization Problem. *GECCO '06,. ACM Press*.

12. Cai, K.Y., et al. (2005) Adaptive Testing of Software Components. *ACM Symposium on Applied Computing*. 1463-1469.

13. Griss, M.L. (1993). Software reuse: From library to factory. *IBM Systems Journal*. 32(4): 548-566.

14. Deng, G. (2005). Supporting Configuration and Deployment of Component-based DRE Systems Using Frameworks, Models, and Aspects. *ACM OOPSLA* '05 : p. 152-153.

15. Frakes, W.B. and K. Kang. (2005). Software Reuse Research: Status and Future. *IEEE Transaction On Software Engineering*. 31(7): 529-536.

16. Goula˜o, M. (2004). Software Components Evaluation : an Overview. *ACM Computing Surveys*.

17. Andreou, A.S. and M. Tziakouris. (2006). A quality framework for developing and evaluating original software components. *Information and Software Technology Journal*. 122-141.

18. Goula˜o, M.A. and A.F. Brito. (2002).Towards a Component Quality Model. in *28th EUROMICRO Conference*. Dortmund, Germany.

19. Abreu, F.B. (2001). *Comparing Software Quality Models*. INES Technical Report (in Portuguese).

20. Washizaki, H., H. Yamamoto, and Y. Fukazawa. (2003). A Metrics Suite for Measuring Reusability of Software Components. in *Proceedings of the Ninth International Software Metrics Symposium* (METRICS'03).

21. Cardino, G., Francesco, and A. Valerio. (1998). The Evaluation of Framework Reusability. *ACM Computing Surveys*.

22. Barnes, B., Durek, T., Gaffney, J., and Pyster, A. (1988). A Framework and Economic Foundation for Software Reuse In: Tutorial: Software Reuse: Emerging Technology, ed. W. Tracz. Washington *IEEE Computer Society*. 77-88.

23. Sommerville, I. (2001), *Software Engineering*. Sixth Editioned. Unted States of America: Addison Wesly.

24. Pressman, R.S. (2001). *Software Engineering : A practitioner's Approach*. Mc Graw Hill International Edition.

25. Frakes, W.B., V. Tech, and Terry. (1996). Software Reuse: Metrics and Models. *ACM Computing Surveys*. 24(2): p. 415-435.

26. Mili, H., Ah-Ki, E., Godin, R., and Mcheick, H. (2003). *An experiment in software component retrieval.* Information and Software Technology. 45(2003): 633-649.

27. Biggerstaff, T. and C. Richter. (1989). Reusability Framework, Assessment, and Directions. *IEEE Software*. 4: 41- 49.

28. Jacobson, I., M.L. Griss, and P. Jansson. (1997). Software Reuse: Architecture, Process, and Organization for Bussiness Success. New York: *ACM Press/Addision Wesley*.

29. *Building a Fuzzy Inference System* Available [cited 2007] from: http:// mathworks.com/products/fuzzylogic/description3.html.

30. Heinman, G.T. and W.T. Council. (2001). *Component Based Software Engineering: Putting the pieces together*. Addison-Wesley.

31. Frakes, W. and S. Isoda. (1994)., Success factors of systematic reuse. *IEEE Software.* 11(5): 14-19.

32. Poulin, J.S. (1997). *Software Reuse Principles, Practices, and Economic Models*. Addison Wesley.

33. Jawawi, D.N.A. (2006). *A Framework for Component-Based Reuse For Autonomous Mobile Robot Software.* PhD Theses. Universiti Teknologi Malaysia.

34. Bieman, J. (1992). *Deriving measures of software reuse metrics in object-oriented systems. In BCS-FACS Workshop on Formal Aspects of Measurement, Springer-Verlag. BCS-FACS Workshop on Formal Aspects of Measurement, Springer-Verlag.*

35. Bieman, J. and S. Karunanithi. (1993). Candidate reuse metrics for object oriented and ADA software. *EEE-CS First International Software Mertics Symsiposium.*

36. Chidamber, S.R. and K.C. F. (1994). *A* Metrics Suit for Object Oriented Design. *IEEE Transaction of Software Engineering*. 20: 476-493.

37. Gill, N.S., Grover, P. S. (2003). Component-Based Measurement: Few Useful Guidelines. *ACM SIGSOFT Software Engineering Notes.*. 28(6).

38. McIlory, M. (1969). Mass Produced Software Components. in *NATO Conference Software Engineering*. Petrocelli/Charter, New York.

39. Freeman, P. (1987). "Reusable Software Engineering: Concepts and Research Directions." in Tutorial: Software Reusability. *Computer Society Press of the IEEE*. (b): 10-23.

40. Jacobson, I. (1993). *Object-Oriented Software Engineering: A Use Case Driven Approach.* Addison-Wesley Publishing Company.

41. Kotonya, G., I. Sommerville, and S. Hall. (2003). *Towards A Classification Model for Component-Based Software Engineering Research*. Computing Department Lancaster. *EUROMICRO Conference.*

42. Clements, P. (2002). *Software Product Lines*. Addison-Wesley.

43. Her, J.S., Kim, J. H., Rhew, S. Y., and Kim, S. D. (2006). *A framework for evaluating reusability of core asset in product line engineering,* Information and Software Technology. 740-760.

44. Brahnmath, G., G. J., Raje, R. R., Olson, A. M., Auguston, M., Bryant, B. R., and Burt, C. C. (2002). *A quality of service catalog for software components*. *Southeastern Software Engineering Conference*. Huntsville, Alabama.

45. Bertoa, M. and A. Vallecillo (2002). *Quality Attributes for COTS Components.* . in *6th International Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'2002)*. Málaga, Spain.

46. Jezequel, J.M. and B. Meyer. (1997). Design by Contract: The Lessons of Ariane. *IEEE Computer Society*. 30(02): 129-130.

47. Alvaro, A., E.S. Almeida, and S.R.L. Meira (2005). Towards a Software Component Quality Model. *5th International Conference on Quality Software (QSIC).*

48. Alvaro, A., E.S. Almeida, and S.R.L. Meira. (2006). A Software Component Quality Model : A Preliminary Evaluation. *32nd EUROMICRO Conference on Software Engineering and Advanced Application (EUROMICRO-SEAA'06).*

49. Sedigh-Ali, S., A. Ghafoor, and R.A. Paul. (2001). Software Engineering Metrics for COTS-Based Systems. *IEEE Computer Society,*

50. Bertoa, M. and A. Vallecillo. (2004). *Usability metrics for software components. QAOOSE 2004*. Oslo.

51. Dumke, R. and A. Schmietendorf. (2000). *Possibilities of the Description and Evaluation of Software Components*, in *Metrics News.*

52. Hoek, A.v.d., E. Dincel, and N. Medvidovic (2003). *Using Service Utilization Metrics to Assess and Improve Product Line Architectures. Ninth International Software Metrics Symposium (Metrics'03)*. Sydney, Australia: IEEE Computer Society Press.

53. Zelkowitz, M.V. (1998). Experimental Models for Validating Technology. in IEEE Computer. 23-31.

54. Hooper, J.W.a.R.O.C., ed. R.A.D. (Ed). (1991). *Software Reuse: Guidelines and Methods.* New York: Plenum Press.

55. Alvaro, A., E.S. Almeida, and S.R.L. Meira. (2007). Component Quality Assurance: Towards a Software Component Certification Process. *IEEE Computer Society Press*, 134-139.

56. Caldiera, G. and V.R. Basili. (1991). Identifiying and Qualifying Reusable Software Components. *IEEE Computer*, vol. 24, Feb.

57. McCall, J.A., P.K. Richards, and G.F. Walters. (1977). *Factors in Software Quality.*, US Rome Air Development Center Reports US.

58. ISO-9126, I.I.S. (1991). *Software Product Evaluation-Quality Characteristics and Guidelines for Their Use,*

59. J. Han. (1998). *A* Comprehensive Interface Definition Framework for software Components. 5th Asis Pacific Software Engineering Conference : *IEEE CS.*

60. Poulin, J.S. (1996). *The Search for a General Reusability Metrics*. *Workshop on Reuse and NASA Software Strategic Plan, Fairfax, VA.*

61. G. Sindre, R. Conradi, and E. Karlsson. (1995) *The REBOOT Approach to Software Reuse*, . Journal of System and Softtware, Elsevier. 30(3): 201-212.

62. Hitam, M.S. (2006). *Design and Implementation of Fuzzy Control for Industrial Robot*. Industrial Robotic: Theory, Modelling, and Control. 409-438.

63, Panwar, D. and P. Tomar. (2011). *New method to find the maximum number of faults by analyzing reliability and reusability in component-based* *software.* Proceedings of the 3rd International Conference on Trendz. Information Sciences and Computing, December 8-9, 2011, Chennai, India. 164-168.

64. Sagar, s., N.W. Nerurkar and A. Sharma. (2010). A soft computing based approach to estimate reusability of software components. *ACM SIGSOFT Software Eng. Notes,* 35: 1-465.

65. Ibraheem Y.Y. Ahmaro, Abdallah M. Abualkishik and Mohd Zaliman Mohd Yusoff. (2014). Taxonomy, Definition, Approaches, Benefits, Reusability Levels, Factors and Adaption of Software Reusability: A Review of the Research Literature. *Journal of Applied Sciences*. 14: 2396-2421.

66. Sommerville, I. (2004). *Software Engineering.* 7th Edition, Addison-Wesley, New York, ISBN: 0321210263.

67. L.A. Zadeh, (1965), *Fuzzy sets, Elsevier Information and Control*. Volume 8, Issue 3, 338–353.

68. Jalender, B., A. Govardhan and P. Premchand. (1996). *Designing code level reusable software components.* Int. J. Software Eng. Appl. 2012.3: 219-229.
49: 255-304.

69. S. Ismail1, W.M.N. Wan Abdul Kadir and N M Mohd Noor. (2012). *Penggunaan Semula Komponen Berorientasikan Objek dapat Membantu dalam Pembelajaran Pengaturcaraan dan Pembangunan Sistem.* International Conference on Quality of Teaching & Learning (ICQTL 2012).

70. Narasimhan, V. L. Parthasarathy, P. T., and Das, M. .(2009), *Evaluation of a Suites of Metrics for Component Based Software Engineering (CBSD),* Issues in Informing Science and Information Technology, Volume 6. p 731-740.

71. Shambhu Kr. Jha. and Dr.R.K. Mishra. (2015). Accessing So,ware quality For Component-Based Software through Trustworthiness and Dependenbability Analysis, Internal Journal of Development Research, Vol. 5, Issue, 04, pp. 4259-4261.

72. Khan A. I., Noor-ul-Qayyum, and Khan U. A. (2012). An Improved Model for Component Based Softwar Development, Software Engineering Journal, , 2940: 138-146.

73. Sommerville, I. (2006). Software Engineering. 8th Edition. Addison-Wesley, New York, ISBN-13: 978-0321313799.

74. Tyagi, K. and Sharma. (2014). A, Significant Factors for Reliability Estimation of Component Based Software Systems, Journal of Software Engineering and Application, 7, 943-942.

75. Land R., Alvaro A., and Crnkovic I. (2009). Towards Efficient Software Component Evaluation, Component Selection and Certification Swedish Foundation for Strategic Research (SSF).

76. Goula˜o, M., and Abreu F. B. e. (2011). An Overview of Metrics-based Approaches to support software Components Reusability Assessment. *ACM Computing Surveys.*

77. Szypers C. Gruntz D. and Murer S. (2002). Component Software Beyond Object Oriented Programming, 2nd edition. New York: Press – Adision Wesley.

78. Goulão M., and Abreu F. B. (2004). "Cross-Validation of a Component Metrics Suite," presented at IXJornadas de Ingeniería del Software y Bases de Datos, Málaga, Spain.

79. Diomidis D. Spinellis, http://www.spinellis.gr/sw/ckjm/doc/indexw.html (acess 7 Mac 2016)

80. Hristov D., Hummel O., Huq M., and Janjic W. (2012). Structuring Software Reusability Metrics for Component-Based Software Development
The Seventh International Conference on Software Engineering Advances (ICSEA), ISBN: 978-1-61208-230-1, page 421- 429.

81. http://www.aptronix.com/fide/whyfuzzy.htm, (acess 10 Mac 2016)

82. Singh Y., Bhatia P. K. and Sangwan O. (2011). Software Reusability Assessment Using Soft Computing Techniques, ACM SIGSOFT Software Engineering Notes, Volume 36 Number 1, page 1-7.

83. Parvinder S. Sandhu, and Hardeep Singh. (2006). A Reusability Evaluation Model for OO-Based, International Journal of Computer Science, Volume 1 Number 4 page 259-264.

84. Nerurkar N.W., Kumar A. and Shrivastava P. (2010). Assessment of Reusability in Aspect-Oriented Systems Using Fuzzy Logic, ACM SIGSOFT Software Engineering Notes, Volume 35 Number 5, page 1-5.

85. M. Halstead. (1977). Elements of Software Science, Amsterdam: Elsevier North-Holland.

86. T.J. McCabe. (1976). A Complexity Measure, IEEE Transactions on Software Engineering, , vol. 2, pp. 308-320.

87. J. Barnard. (1998). A new reusability metric for object-oriented software, Software Quality Journal, vol. 7, Jan , pp. 35-50.

88. Y. Mao, H. Sahraoui, and H. Lounis. (1998). Reusability Hypothesis Verification using Machine Learning Techniques: A Case Study, Proceedings of the International Conference on Automated software engineering, *IEEE*, pp. 84-93.

89. Y. Lee and K.H. Chang. (2000). Reusability and maintainability metrics for object-oriented software, Proceedings of the 38th annual on Southeast regional conference (ACM-SE 38), ACM, New York, NY, USA, pp. 88-94.

90. E.S. Cho, M.S. Kim, and S.D. Kim. (2001). Component Metrics to Measure Component Quality, Proceedings of the Eighth Asia-Pacific on Software Engineering Conference (APSEC '01), IEEE Computer Society, Washington, DC, USA, pp. 419-426.

91. L.H. Etzkorn, W.E. Hughes Jr., and C.G. Davis. (2001). Automated reusability quality analysis of OO legacy software, Information and Software Technology. vol.43, pp. 295-308.

92. S. Bhattacharya and D.E. Perry. (2005). Contextual reusability metrics for event-based architectures, Intern. Symposium. On Empirical Software Engineering, pp. 459-468.

93. G. Gui and P.D. Scott. (2008). New Coupling and Cohesion Metrics for Evaluation of Software Component Reusability, Proc. Of the Intern. Conf. for Young Computer Scientists, pp. 1181-1186.

94. N. Gill and S. Sikka. (2011) Inheritance Hierarchy Based Reuse & Reusability Metrics in OOSD, International Journal on Computer Science and Engineering (IJCSE), vol.3, pp. 2300-2309.

95. M.A.S. Boxall and S. Araban. (2004). Interface Metrics for Reusability Analysis of Components, Australian Software Engineering Conference (ACWEC’04), Melbourne, Australia, pp. 40-50.

96. A. Sharma, P.S. Grover, and R. Kumar. (2009). Reusability assessment for software components, SIGSOFT Software Engineering Notes, vol.34, No.2, pp. 1-6.

97. O.P. Rotaru and M. Dobre. (2005). Reusability Metrics for Software Componenents, ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'05), Washington DC, USA, pp.24-31.

98. "Basic Concepts of Component-based software", Online Available:http://www.idt.mdh.se/kurser/cdt501/2008/lectures/book%20Basic %20Concepts%20of%20CBSE.pdf

99. Asif Irshad Khan1, Noor-ul-Qayyum2, Usman Ali Khan2, An Improved Model for Component Based Software Development, DOI: 10.5923/j.se.20120204.07 Published online at http://journal.sapub.org/se Copyright © 2012 Scientific & Academic Publishing, Software Engineering 2012, 2(4): 138-146

100. A. K. Gupta, V. K. Yadav, S. Kumar and Vishal. (2012). A Robust Retrieval Scheme for Software Component Reuse, International Journal of Engineering and Innovative Technology (IJEIT) ISSN: 2277-3754 Volume 2, Issue 1.

101. B.Jalender, N.Gowtham, K.Praveenkumar, K.Murahari, K.sampath. (2010). Technical Impediments to Software Reuse, International Journal of Engineering Science and Technology (IJEST), Vol. 2(11), p. 6136-6139.

102. M. R. J. Qureshi, S. A. Hussain. (2008). A reusable software component-based development process model, Advances in Engineering Software, v.39 n.2, p.88-94.

103. Peterson, R. A. (1994). Meta-analysis of Alpha Cronbach's Coefficient. Journal of Consumer Research, 21(2), 381–391.

104. Jedlitschka, A., M. Ciolkowski, and D. Pfahl. (2008). Reporting Experiments in Software Engineering, in Guide to Advanced Empirical Software Engineering, F. Shull, et al., Editors. Springer. Springer London. p. 201-228.

105. http://www.cs.kent.edu/~jmaletic/cs63901/lectures/SoftwareMetrics.pdf (acess 15 Mei 2016)

106. J. M. Morel, http://www.idi.ntnu.no/grupper/su/courses/dif8901/presentations2003/r01.pdf (acess 15 Mei 2016)

107. Lazaro, M. and E. Marcos. (2005). Research in Software Engineering: Paradigms and Methods, in PHISE'05.

108. Sjoeberg, D.I.K., Hannay, J.E., Hansen, O., Kampenes, V.B.,Karahasanovic,

A., Liborg, N.-K., Rekdal, A.C. (2005). "A survey of controlled experiments in software engineering," *IEEE Transactions on Software Engineering*, vol.31, no.9, pp.733-753, .

109. Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B. and Wesslén, A. (2012). *Experimentation in Software Engineering*. Berlin Heidelberg. :Springer Science & Busines Media.

110. Zelkowitz, M.V. and Wallace, D.R. (1998). Experimental Models for Validating Technology, Computer, 3 Quality (5), 23-31, pp. 23–31.

111. Jatain A. and Gaur D. (2012). Estimation of Component Reusability by Identifying Quality Attributes of Component-A Fuzzy Approach, CCEIT, ACM, Combatore, India, page 738-742.

112. Singh A. P. and Tomar P. (2014). Estimation of Component Reusability through Reusability Metrics, International Journal of Computer, Electrical, Electrical, Automation, Control and Information Engineering Vol: 8, No 11, page 1729-1736.

113. Koteska B. and Velinov G. (2013). Component-Based Development: A Unified Model of Reusability Metrics, ICT Innovations 2012, AISC 207, Springer-Verlag Berlin Heidelberg, pp. 335–344.

114. Borysowich C., http://it.toolbox .com/blog/enterprise –solution/design-principle-fanin-vsfanout-16088 (acess 15 Mei 2010).

115. Schroeder M., A Practical Duide to Object Oriented Metrics, IT Pro, (1999).

116. Zadeh, L.A. (2008). Is There A Need For Fuzzy Logic?. Information sciences, *178*(13), pp.2751-2779.

117. http://whatis.techtarget.com/definition/fuzzy-logic (acess 13 April 2017)

118. Li, P. and Yang, G. (2009). Backstepping adaptive fuzzy control of uncertain nonlinear systems against actuator faults. Journal of Control Theory and Applications, *7*(3), pp.248-256.

119. Hao, L.Y., Park, J.H. and Ye, D. (2016). Fuzzy logic systems-based integral sliding mode fault-tolerant control for a class of uncertain non-linear systems. IET Control Theory & Applications, *10*(3), pp.300-311.

120. Tong, S., Wang, T. and Li, Y. (2014). Fuzzy adaptive actuator failure compensation control of uncertain stochastic nonlinear systems with

unmodeled dynamics. IEEE Transactions on Fuzzy Systems, *22*(3), pp.563-574.

121. Li, Y.X. and Yang, G.H. (2015). Robust fuzzy adaptive fault-tolerant control for a class of nonlinear systems with mismatched uncertainties and actuator faults. Nonlinear Dynamics, *81*(1-2), pp.395-409.

122. Hong, Y., Pasman, H. J.,Sachdeva, S.,Markowski, A. S., and Mannan, M.S. (2016). A Fuzzy Logic And Probabilistic Hybrid Approach To Quantify The Uncertainty In Layer Of Protection Analysis, Journal Of Loss Prevention In The Process Industries, Volume 43, Pages 10–17.

123. Singh, C., Pratap, A. and Singhal, A. (2014). July. An estimation of software reusability using fuzzy logic technique. In Signal Propagation and Computer Technology (ICSPCT), 2014 International Conference, pp. 250-256.

124. Hameed IA, Sorensen CG. (2010). Fuzzy systems in education: a more reliable system for student evaluation. INTECH Open Access Publisher.