IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 10, October 2019

A Methodology to Evaluate Extensibility of Object Oriented Design: A Product Transition Perspective

Dr. Abdullah[1], Dr. Mahfuzul Huda[2], Hagos Yirgaw [3]

Assistant Professor, Adigrat University, Adigrat-Tigray, Ethiopia-Africa [1]

Assistant Professor, Saudi Electronic University, Riyadh -Saudi Arabia[2]

Lecturer, Adigrat University, Adigrat-Tigray, Ethiopia-Africa[3]

Abstract: A system is said to be extensible; if any modifications can be made to any of the existing system functionalities in addition of new functionalities with minimum impact. To attain extensibility, it has to be scheduled properly starting from the initial stage of the system development life cycle. Keeping in mind all the probable upcoming modifications to be made, the software designer should select the appropriate design patterns and finish the design for the application. As soon as the application design is finished, it must be examined to make sure that the application is extensible.

This research paper recommends a research framework for Extensibility evaluation process and does an extensive review on extensibility of object oriented software. A metrics based Extensibility Evaluation Model for Object Oriented Design" has been recommended by creating the relationship in the middle of design properties. Consecutively researchers justifying the model correlation with the help of statistical measures, which shows that for all the System, Coupling, Cohesion, Inheritance, Polymorphism are highly correlated with Extensibility. The developed model has been authenticated by experimental tryout. Finally, it incorporates the empirical validation of the extensibility evaluation model as the author's most important contribution. In Conclusion, Hypothesis testing is performed to test the significance of r (Correlation Coefficient) using 2-tailed test for a 95% confidence interval with different degrees of freedom. As a result, the researcher's claim of correlating Extensibility with Coupling, Cohesion, Inheritance and Polymorphism at design phase is statistically extremely justified.

Keywords: Extensibility, Reusability, Flexibility, Extensibility, Maintainability, Scalability, Design Phase, Object Oriented Design, Software Quality, Product Transition.

I. INTRODUCTION

Extensibility is a software engineering and system design principle where the implementation takes upcoming development into consideration [8]. The term extensibility can also be seen as a systemic measure of the ability to extend software and the level of effort required to implement the extension at the design phase of the development life cycle. Extensions can be through the addition of new functionalities or through modification of existing functionalities [12, 26, 27]. The central theme is to provide for change typically enhancements, while minimizing impact to existing system functions.

An extensible system is one whose internal structure and dataflow are minimally or not affected by new or modified functionalities. Since software systems are extended lived and will be modified for new features and additional functionalities required by end users, extensibility allows system designers and developers to grow or add to the software's capabilities and facilitates systematic reuse. Several of its methodologies include facilities for permitting users' individual program routines to be inserted and the capabilities to describe new data types as well as to describe new formatting markup tags.

II. EXTENSIBILITY

With the developing technologies, requirements are also varying and increasing day by day.The revolutions in the software industry forces the system designers and tool builders to improve their design and products to be well-suited with these revolutions [13, 14]. Making modifications to an already deployed code might not be easy all the time. It may be easy for a small application to be recompiled and redeployed. But for large application with many end users, recompilation and redeployment may take a more time and outcomes in consumption of resources. Modern software need to be extended by other developers or programmers to fit in the customer requirements. Software teams do not want to touch the code base for each and every modification since it is error prone. It is at this condition that, the

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 10, October 2019

designers and developers start thinking about software extensibility and extensible designs come to the assistance of software developers. The significant feature of extensibility is to make any modification in existing system functions with minimum impact. By extensions, it means whichever the adding of new functionality or modification of existing functionality [15, 16, 29].

In Programming Languages, it appears as a set of mechanisms and concepts that make it easy to extend the software [19]. When the software is extended there will be some added features along with the functionalities that were available previously. Most of the time, extensibility is misunderstood for reusability. Code reusability is copy/paste of the code that already exists for a similar application. So the resultant application will be a newer or more efficient version of the existing version. But in case of extensibility, it is not reuse of the available code. We define extensibility as the ability of a system to be extended with new functionality with minimal or no effects on its internal structure and data flow [28]. The software quality attributes presented below are the essential contributing factors to software extensibility as a system property. Each design principle suggested in this study is related to one or more of those quality attributes in order to achieve the final purpose of extensibility. The sections below extensibility core contributors will further explain the quality attributes and their definitions. It will also present what design properties map to every single quality attribute.

III. Software Extensibility at Design Phase

Extensible design supports the iterative development principles. It permits functionality to be implemented in small steps as needed. To attain extensibility, it has to be scheduled appropriately starting from the initial phase of the system development life cycle. The system designer must have an idea of probable upcoming requirements and how the application will have to be changed in future [19]. For instance, if it is an application for a restaurant, provision should be there to increase more variety items in the menu and calculate the bill accordingly.

Extensible system design in software engineering is to accept that not the whole thing can be designed in advance. A light software framework which permits for modifications is provided in its place. Acceptance change is essential to the extensible design, in which additions will be continual [23]. Each portion of the application will be workable with any modifications, and the notion of modification through addition is the midpoint of the entire system design. Extensibility imposes fewer and cleaner dependencies throughout development, as well as reduced coupling and more cohesive concepts, plus well-defined interfaces.

IV. Diagnosing of Extensibility

During the diagnosing phase we had already established good relationship between extensibility core contributors and design properties. Furthermore, we identified Reusability, Flexibility, Extensibility, Maintainability and Scalability as good quality attributes to use when designing for extensibility. Most of the studies focused their attempt to examine the impact of object oriented characteristics and have successfully established relationships with quality factors. However, we examined and assessed their impact on the particular aspect of study i.e. extensibility and by associatively and similarity perspective, concluded on identifying extensibility factors affected by object oriented characteristics. It was observed that each of these characteristics, either have positive or negative impact on the factors that affect extensibility of object oriented software. After an exhaustive review of available literature on the topic [4, 5, 6, 7, 8, 9, 10, 11, 22] the relation between OO software characteristics and extensibility factors (as depicted in Figure1) has been established. Based on the relationship shown below, a model has been developed for evaluate software extensibility. Further, the relative significance of individual design properties that influence software extensibility is weighted proportionally.

V. Extensibility Core Contributors

The purpose of this section is to identify and assess a number of suitable design principles and extensibility core contributors to minimize the risk for code deterioration, as a proof of concept enabling the obtaining of both practical experiences and theoretical reflection. We are however aware of the fact that software systems cannot be kept sane by the influence of design principles alone, and wish to make clear that we are not suggesting this method as a Silver Bullet [7] solution. Other factors also influence the life of a software project, such as its developer's dedication to follow the set design or its manager's willingness to allow for the design to be followed. Even so, we believe that the application of and dedication to a reasonable set of design principles may well increase the success probability of a software project in terms of Extensibility and code deterioration avoidance.

VI. Extensibility Quality Factors and Quality Criteria

Criteria are the characteristics which define the quality factors. The criteria for the factors are the attributes of the software product or software production process by which the factor can be judged or described. The relationships between the factors between the criteria can be found in Table 4.1.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 10, October 2019

There are four reasons for developing a list of criteria for each factor:

1. Criteria offer a more complete, real definition of quality factors.
2. Criteria common among factors support to clarify the interrelation between the factors.
3. Criteria permit audit and review metrics to be developed with more easiness.
4. Criteria permit us to identify that area of quality factors which may not be up to a predefined acceptable standard.

Table 4.1: Extensibility core contributors with description [1, 2, 3, 4-11, 22]

Core Contributors	Description
Reusability	The extent to which a program can be reused in other applications.
Flexibility	The effort required to modify an operational program.
Extensibility	The effort needed for modification, fault removal or for environmental change.
Maintainability	Characteristics related to the effort needed to make modifications, including corrections, improvements or adaptation of software to changes in environment, requirements and functions specifications.
Scalability	Scalability is the ability of a system to expand in a chosen dimension without major modifications to its design.

Table 4.2: Extensibility Core Contributors with criteria of software quality [1, 2, 4-11, 22]

Core Contributors	Criteria of Software Quality
Reusability	Generality, Hardware Independence, Self-Documentation, Modularity, Software Independence
Flexibility	Complexity, Concision, Consistency, Self Documentation, Simplicity, Generality, Modularity, Expandability
Extensibility	Structured, Augment ability
Maintainability	Consistency, Modularity, Self-Documentation, Software Independence, Concision, Instrumentation
Scalability	Modifiability, Expandability, Simplicity

VII. MAPPING BETWEEN OO SOFTWARE CHARACTERISTICS AND EXTENSIBILITY CORE CONTRIBUTOR

In order to establish a contextual impact relationship between object oriented software characteristics and extensibility core contributors, the influence of object oriented characteristic on each extensibility core contributors were examined by several researchers. Most of the studies focused their attempt to examine the impact of object oriented characteristics and have successfully established relationships with software quality core contributors. However, we examined and assessed their impact on the particular aspect of study i.e. extensibility and by associatively and congruence perspective, concluded on identifying extensibility core contributors affected by object oriented characteristics. It was observed that each of these characteristics, either have positive or negative impact on the core contributors that affect extensibility of object oriented software.

After an exhaustive review of available literature on the topic [2, 4, 5, 6, 7, 8, 9, 10, 11, 22, 17, 18, 20], the relation between object oriented software characteristics and extensibility core contributors (as depicted in Figure1) has been established. Based on the relationship shown below, a model has been developed (equation 3) for estimating extensibility core contributors. Further, the relative significance of individual design properties that influence software extensibility is weighted proportionally.

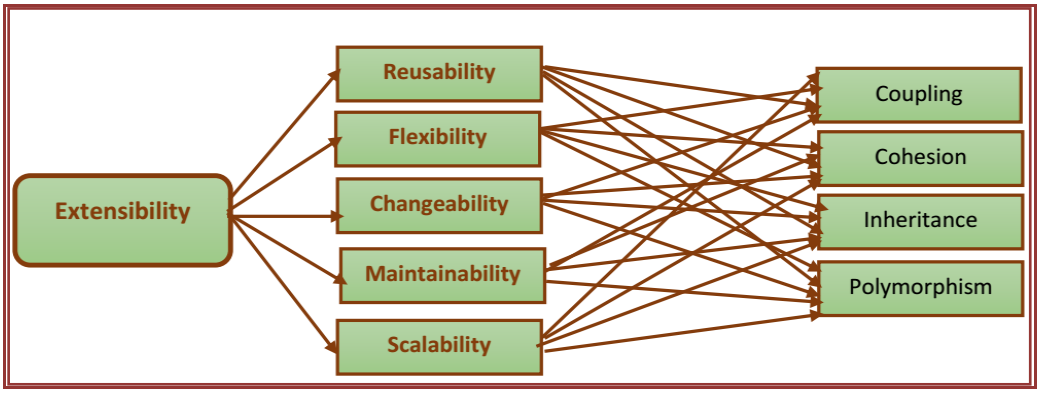

Fig1: Research Framework to Evaluate Software Extensibility of a Class Hierarchy

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 10, October 2019

VIII. ACTION PLANNING

This phase was used primarily to identify which design properties we saw fit for the purpose of providing extensibility and thus counter code deterioration. We spent a large amount of time studying the related research and trying to come to terms with where the design constructs would fit in, and to what level they would be useful [3].

In order to establish a relationship between design constructs and extensibility, the influence of design constructs quality attributes are being examined with respect to SATC's attributes [20]. It was observed that each design constructs affects certain quality attributes [20-22, 24, 25]. The extensive review of object oriented development books and publications [7, 22, 30-32] indicate that the encapsulation is viewed to promote efficiency and complexity. Inheritance design property has a significant influence on the efficiency, complexity, reusability maintainability. While low coupling is considered good for understandability, complexity, reusability and maintainability. Upper measures of coupling are observed to unfavorably affect these quality attributes. Cohesion is viewed to have a significant effect on a design's understandability and reusability. Based upon the design property to extensibility relationship, the relative significance of individual design properties that influence software extensibility is weighted proportionally.

Extensibility of a class depend upon one or more number of design attributes , component-wise effect may be speculated and respective component weightage (CW) may be fixed using regression equation. Thereby, the CWs of individual design attributes have been calculated in terms of regression coefficient β. Twenty eight (28) medium sizes Java SE projects were used to fit the regression line and model validation by acquiring real data from commercial projects of Software Company based in Bangalore, India. Name of the Projects and actual source data is being concealed, as per wishes of company's management. We have assured authenticity of source data, to the best possible extent. These projects include the number of classes (15-25) and the metrics value of every class. In addition, the mean value of the expert's rating of extensibility of these class diagrams is also known and termed as 'Known Value' in this research paper. The projects were independently completed over a period six months. Using these data, the CW coefficient calculated for Coupling, cohesion, inheritance and polymorphism to show the complexity relationship with design properties (Coupling, cohesion, inheritance and polymorphism). Equation 3 summarizes the computational formula for Extensibility with the component weightage.

IX. MODEL DEVELOPMENT FOR SOFTWARE EXTENSIBILITY

An In order to establish a model for Extensibility, multiple linear regression technique has been used. Multivariate linear model is given as follows.

$$Y=a0\pm a1 \times y1\pm a2 \times y2\pm a3x\ y3\ldots\ldots an \times yn \quad \text{Eq. (1)}$$

$$\textbf{Extensibility}= a0 \pm a1 \times \text{Coupling} \pm a2 \times \text{Cohesion} \pm a3 \times \text{Inheritance} \pm a4 \times \text{Polymorphism} \quad \text{Eq. (2)}$$

The relationship amongst Extensibility key factors Reusability, flexibility, extensibility, maintainability, scalability and object oriented design properties has been established as depicted in Fig. 1. Using SPSS, values of coefficient are calculated and extensibility model is formulated as given below.

$$\textbf{Extensibility= 7.442 -.459 x Coupling -.040 x Cohesion -.677 x Inheritance +1.128 x Polymorphism} \quad \text{Eq. (3)}$$

Table 6.1 shows the coefficients for Extensibility model. The Unstandardized coefficients component of the table 6.1 gives us the values that we require in order to develop the regression Equation (3). The experimental assessment of Extensibility is very hopeful to get Extensibility index of object oriented design for low cost software Extensibility and maintainability.

Table 6.1: Coefficients values for Extensibility Evaluation Model

Coefficients[a]						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	7.442	1.456		5.112	.004
	Coupling	-.459	.168	-.644	-2.738	.041
	Cohesion	-.040	.165	-.049	-.242	.818
	Inheritance	-.677	.393	-.402	-1.721	.146
	Polymorphism	1.128	.370	.866	3.050	.028
a. Dependent Variable: Extensibility						

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 10, October 2019

The model summary table 6.2 results are most helpful when performing multiple regressions. Capital R is the coefficients determinant that tells us how powerfully the multiple independent variables are associated to the dependent variable. The value of R Square is very helpful as it provides us the coefficient of determination.

Table 6.2: Extensibility Evaluation Model Summary

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.911[a]	.829	.693	.31066
a. Predictors: (Constant), Polymorphism, Cohesion, Inheritance, Coupling				

X. VALIDATING THE EXTENSIBILITY MODEL

A. STATISTICAL SIGNIFICANCE BETWEEN EXTENSIBILITY AND OBJECT ORIENTED DESIGN PROPERTIES

The 18 software projects that are used in presenting the statistical importance amongst Extensibility and object oriented design properties, we classified the projects as: System A, System B and System C. All the systems are commercial software projects implemented in Java SE with the number of classes as shown in Table 5.

Table 5: Group and Projects for proposed EEMOOD

Group	Projects
System A	5
System B	6
System C	7

Table 5.1 provides the descriptive statistics for System A and Table 5.2 provides the correlation analysis for System A.

Table 5.1: Descriptive Statistics Summary for System A

	Minimum	Maximum	Mean
Extensibility	6.00	9.80	7.7000
Coupling	2.50	3.60	3.2800
Cohesion	1.30	2.70	1.9400
Inheritance	.40	.80	.6000
Polymorphism	1.90	2.90	2.4000

Table 5.2: Correlation Analysis Summary for System A

	Extensibility	Coupling	Cohesion	Inheritance	Polymorphism
Extensibility	1	.906	.949	.868	.996
Coupling	.906	1	.927	.990	.882
Cohesion	.949	.927	1	.887	.920
Inheritance	.868	.990	.887	1	.842
Polymorphism	.996	.882	.920	.842	1

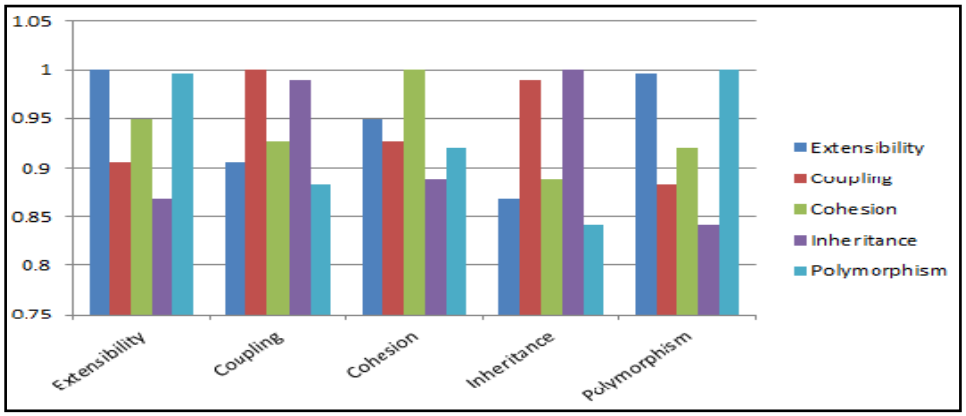

Fig. 2: Correlation Analysis Summary for System A

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 10, October 2019

Table 5.3: Descriptive Statistics Summary for System B

	Minimum	Maximum	Mean
Extensibility	5.90	9.80	8.1000
Coupling	2.50	3.80	3.2000
Cohesion	1.30	2.40	1.7800
Inheritance	.40	.90	.6800
Polymorphism	1.90	2.90	2.4600

Table 5.4: Correlation Analysis Summary for System B

	Extensibility	Coupling	Cohesion	Inheritance	Polymorphism
Extensibility	1	.930	.929	.880	.975
Coupling	.930	1	.820	.890	.868
Cohesion	.929	.820	1	.895	.870
Inheritance	.880	.890	.895	1	.753
Polymorphism	.975**	.868	.870	.753	1

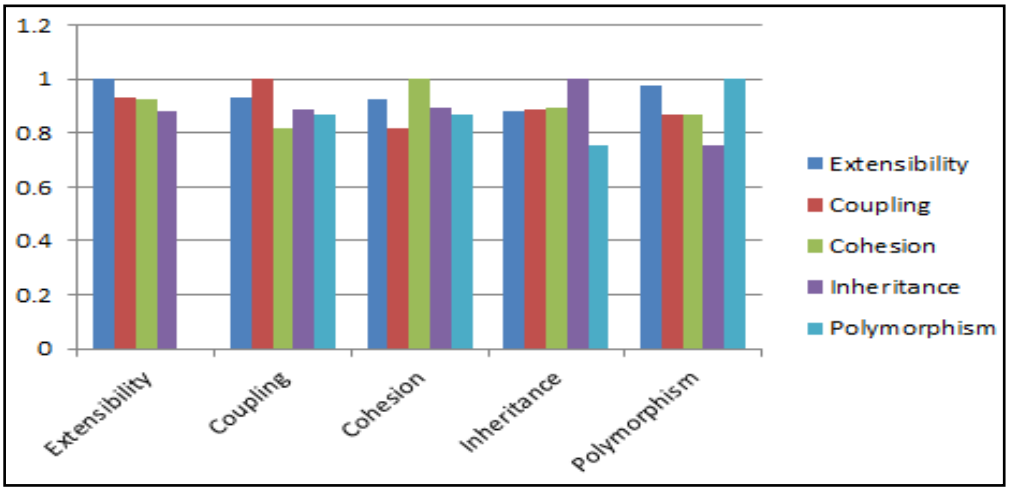

Fig. 3: Correlation Analysis Summary for System B

Table 5.5: Descriptive Statistics Summary for System C

	Minimum	Maximum	Mean
Extensibility	7.40	9.80	8.8000
Coupling	2.50	4.10	3.2400
Cohesion	1.30	2.70	1.8200
Inheritance	.40	1.20	.8000
Polymorphism	1.80	2.90	2.3000

Table 5.6: Correlation Analysis Summary for System C

	Extensibility	Coupling	Cohesion	Inheritance	Polymorphism
Extensibility	1	.919	.907	.925	.950
Coupling	.919	1	.998	.869	.826
Cohesion	.907	.998	1	.851	.806
Inheritance	.925	.869	.851	1	.982
Polymorphism	.950	.826	.806	.982	1

DOI 10.17148/IJARCCE.2019.81001

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 10, October 2019

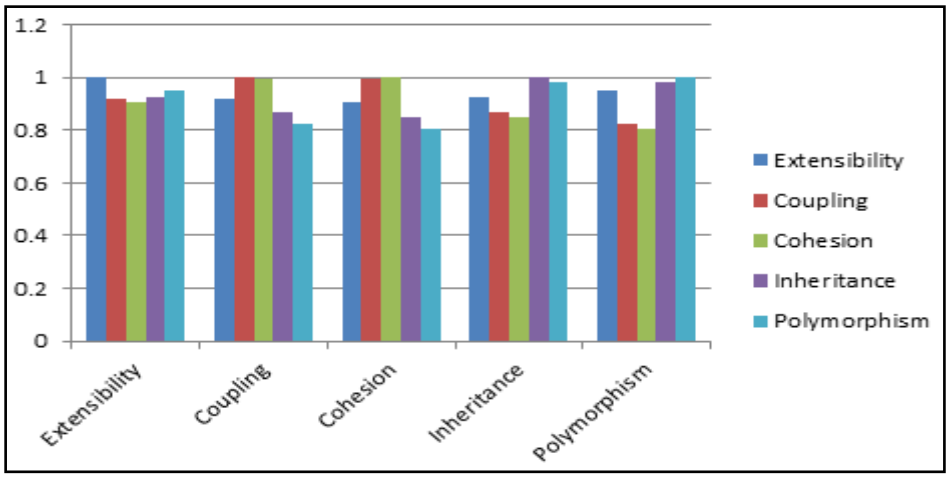

Fig. 4: Correlation Analysis Summary for System C

Table 5.7 summarizes the result of the correlation analysis for Extensibility evaluation model, which shows that for all the System, Coupling, Cohesion, Inheritance, Polymorphism are highly correlated with Extensibility. The value of correlation 'r' lies between ±1, positive value of 'r' in Table 5.7, designates positive correlation between the two variables. The value of 'r' close to +1 specifies high degree of correlation between the two variables in above Table.

Table 5.7: Correlation Analysis Summary for System A, System B and System C

	Extensibility × Coupling	**Extensibility × Cohesion**	**Extensibility × Inheritance**	**Extensibility × Polymorphism**
System A	.906	.949	.868	.996
System B	.930	.929	.880	.975
System C	.919	.907	.925	.950

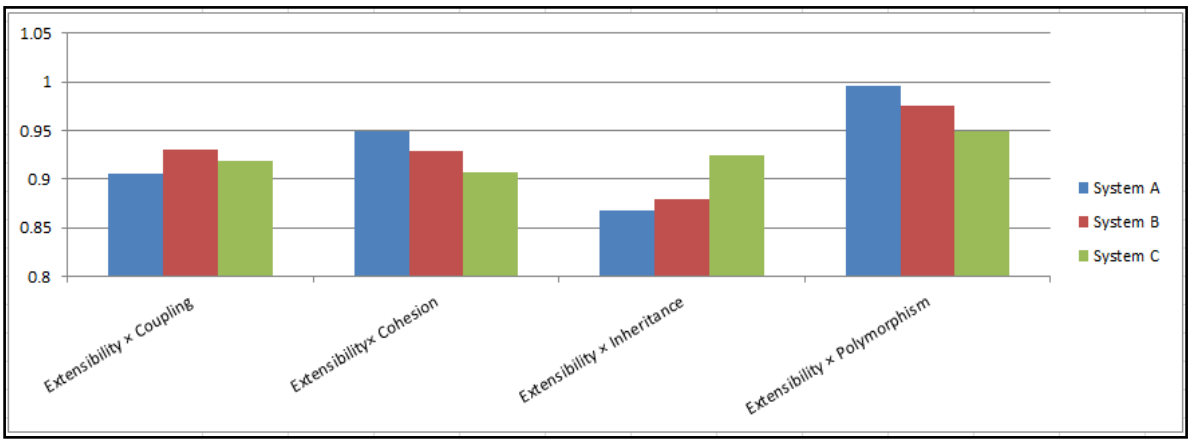

Fig. 5: Correlation Analysis Summary for System A, System B and System C

B. VALIDATION OF EXTENSIBILITY EVALUATION MODEL

This part of study paying consideration how the above developed extensibility model is able to evaluate the extensibility of object oriented software especially at design phase. The empirical model validation is an important step of proposed research to estimate Extensibility evaluation model for better and high level adaptability. In view of this truth practical validation of the Extensibility evaluation model has been performed, using experimental tryouts.

In order, to validate the developed extensibility evaluation model the projects viz. (P1, P2, P3, P4, P5, P6, P7, P18, P19 and P20) were taken. The projects were independently completed over a period of months. A suite of ten projects was developed by different individuals for the study. The values for Extensibility are computed using the equations 3.0 and shown in table 6.1. These testability scores are used to assign the Extensibility Benchmark (BC) to all the ten projects.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 10, October 2019

Table 6.1: Design Extensibility Indices

	Projects									
	P1	P2	P3	P4	P5	P6	P7	P18	P19	P20
Extensibility	2.5	2.2	3.8	3.4	3.7	1.4	0.4	4.4	6.7	5.8
BC	4	3	7	5	6	2	1	8	10	9

A group of ten independent evaluators was allotted to study the quality of the same ten projects in the authentication suite. All the evaluators had 9 to 12 years of rich experience in commercial software development, had knowledge of the object oriented technology, and had developed software using java SE. The study was done over a period of month. All the members analyzed each project's design and assigned the scores on an ordinal scale to the Extensibility by using their own traditional tool. These scores for all attributes were assigned to give the evaluator's Extensibility Benchmark (BC) for the project. The projects were ranked 1 through 10 based on decreasing Extensibility scores by each evaluator as shown in Table 6.2

Table 6.2: Evaluators Model and Extensibility Ranking

	Projects									
	P1	P2	P3	P4	P5	P6	P7	P18	P19	P20
Evaluators Ranking	5	1	6	4	7	3	2	9	8	10

C. STATISTICAL EXAMINATION

Charles Speraman's rank relation rs was used to test the impact of correlations between design time metric based Extensibility assessment and the evaluator's implementation based assessment of the project. It provides a nonparametric significance test that works well with ranked data without precise proportional scaling and can be used to detect relationships other than linear one. For the level of significance of α=95%, the threshold value calculated for n=10 projects was ±.781.

The 'rs' was calculated using the technique given as under:
Speraman's Coefficient of Correlation (rs) –

$$rs = 1 - \frac{6 \Sigma d2}{n (n2-1)} \qquad -1.0 \leqslant rs \leqslant +1.0 \qquad \text{Eq. (4)}$$

'd' = Difference between, Calculated Rank and Known Rank of Extensibility.
'n' = Number of total Projects used in the experimentation.

Table 6.3: Computed Rank and Evaluators Ranking Correlation

Projects	P1	P2	P3	P4	P5	P6	P7	P18	P19	P20
Computed Ranks	4	3	7	5	6	2	1	8	10	9
Evaluators Ranking	5	1	6	4	7	3	2	9	8	10
$\sum \mathbf{d^2}$	16									
$r_{s\ Calculated}$	.903									
$r_s > \pm .781$	✓									

The correlation value among calculated Extensibility ranks using proposed model and known ranks are shown in Table 6.3. A correlation value r undoubtedly shows that the Extensibility model is greatly significant. The correlation is up to standard with high level of confidence i.e. at the 95%. It is evident from the correlation values shown in table that the Extensibility indicators evaluated using proposed EEM^{OOD} are highly correlated with the measurements done by human evaluators.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 10, October 2019

D. HYPOTHESIS TESTING OF COEFFICIENT OF CORRELATION

An experimental coefficient of correlation of Coupling, Cohesion, Inheritance and Polymorphism with Extensibility strongly indicates the higher importance and significance of taking into consideration all four design properties (Coupling, Cohesion, Inheritance and Polymorphism) for making an evaluation of software Extensibility at design phase. Additionally, to defend the claim, a test to justify the statistical significance of the correlation coefficient is performed. For the motivation, Hypothesis testing is performed to test the significance of r (Correlation Coefficient) using the given below formula:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}}.$$

With N-2 degree of freedom, a coefficient of correlation is calculated as statistically significance when the t value equivalents or go above the t critical value in the t distribution critical values.

A. H0 (T^M): EXTENSIBILITY AND COUPLING ARE NOT HIGHLY CORRELATED

Table 7.1: Correlation Coefficient Test for Extensibility and Coupling

	System A	System B	System C
Extensibility ^ Coupling	.906	.930	.919
tr	3.70	5.06	5.21
tr-Critical Value	2.57	2.45	2.37
tr >tr-Critical Value	√	√	√
H0(E^C)	Rejected	Rejected	Rejected

B. H0 (T^M): EXTENSIBILITY AND COHESION ARE NOT HIGHLY CORRELATED

Table 7.2: Correlation Coefficient Test for Extensibility and Cohesion

	System A	System B	System C
Extensibility ^ Cohesion	.949	.929	.907
tr	5.21	5.02	4.81
tr-Critical Value	2.57	2.45	2.37
tr >tr-Critical Value	√	√	√
H0(E^C)	Rejected	Rejected	Rejected

C. H0 (T^M): EXTENSIBILITY AND INHERITANCE ARE NOT HIGHLY CORRELATED

Table 7.3: Correlation Coefficient Test for Extensibility and Inheritance

	System A	System B	System C
Extensibility ^ Inheritance	.868	.880	.925
tr	3.02	3.70	5.44
tr-Critical Value	2.57	2.45	2.37
tr >tr-Critical Value	√	√	√
H0(E^I)	Rejected	Rejected	Rejected

D. H0 (T^M): EXTENSIBILITY AND POLYMORPHISM ARE NOT HIGHLY CORRELATED

Table 7.4: Correlation Coefficient Test for Extensibility and Polymorphism

	System A	System B	System C
Extensibility ^ Polymorphism	.996	.975	.950
tr	19.30	8.77	6.80
tr-Critical Value	2.57	2.45	2.37
tr >tr-Critical Value	√	√	√
H0(E^P)	Rejected	Rejected	Rejected

Through using 2-tailed test at the 0.025 for a 99% confidence interval with different degrees of freedom, it is evidence from the tables 7.1, 7.2, 7.3 and table 7.4, the null hypothesis is strongly rejected. As a result, the researcher's claim of

correlating Extensibility with Coupling, Cohesion, Inheritance and Polymorphism at design phase is statistically extremely justified.

CONCLUSION

In this research work, object oriented design constructs and extensibility core contributors are identified and their impact on Extensibility evaluation and improvement at design phase has been analyzed. Table 5.7 summarizes the result of the correlation analysis for Extensibility evaluation model, which shows that for all the System, Coupling, Cohesion, Inheritance, Polymorphism are highly correlated with Extensibility. Considering both, the design constructs and extensibility core contributors, an Extensibility evaluation model for object oriented design has been developed (EEM^{OOD}), and the statistical inferences are validated for high level better acceptability. The research paper furthermore authenticates the computing ability of developed model. The developed model to evaluate extensibility of object oriented software design is extremely reliable and correlated with object oriented design constructs. Extensibility evaluation model has been validated theoretically as well as empirically using experimental try-out. In Conclusion, Hypothesis testing is performed to test the significance of r using 2-tailed test for a 95% confidence interval with different degrees of freedom. As a result, the researcher's claim of correlating Extensibility with design constructs at design phase is statistically extremely justified. The applied validations on the Extensibility evaluation model conclude that proposed model is highly consistent, acceptable and reliable.

REFERENCES

[1]. McCall JA, Richards PK, Walters GF. Factors in software quality, RADC TR-77-369: (Rome: Rome Air Development Center) 1977.
[2]. Boehm BW, Brow JR, Lipow M, McLeod G, Merritt M. Characteristics of software quality. North Holland Publishing. Amsterdam, the Netherlands; 1978.
[3]. Grady, RB. Practical software metrics for project management and process improvement, Prentice Hall; 1992.
[4]. Dromey RG. Concerning the Chimera (software quality). IEEE Software.; 1:33- 43, 1996.
[5]. Lee, Ming-Chang. "Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance." British Journal of Applied Science & Technology 4.21 (2014).
[6]. Boehm BW, Brown JR, Lipow M. Quantitative evaluation of software quality, In Proceeding of the 2nd International Conference on Software engineering.; 592- 605, 1976.
[7]. Dromey RG. A model for software product quality. IEEE Transaction on Software Engineering. 21:146-162; 1995.
[8]. Abdullah, Dr, M. H. Khan, and Reena Srivastava. "Testability Measurement Model for Object Oriented Design (TMMOOD)". International Journal of Computer Science & Information Technology (IJCSIT), Vol. 7, No 1, February 2015, DOI: 10.5121/ijcsit.2015.7115.
[9]. ISO 9001:2005, Quality management system Fundamentals and vocabulary; 2005.
[10]. Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Quantifying Reusability of Object Oriented Design: A Testability Perspective. Journal of Software Engineering and Applications, 8, 175-183. http://dx.doi.org/10.4236/jsea.2015.84018
[11]. Abdullah, Dr, Reena Srivastava, and M. H. Khan. "Modifiability: A Key Factor To Testability", International Journal of Advanced Information Science and Technology, Vol. 26, No.26, Pages 62- 71 June 2014.
[12]. Tomar AB, Thakare VM. A systematic study of software quality models, International Journal of software engineering & application. 12(4):61-70; 2011.
[13]. Al-Qutaish RE. Quality models in software engineering literature: An analytical and comparative study. Journal of American Science. 6(3):166-175; 2010.
[14]. Shanthi PM, Duraiswamy K. An empirical validation of software quality metric suits on open source software for fault-proneness prediction in object oriented system, European journal of Scientific Research. 5(2):168-1812011.
[15]. Abdullah, Dr, Reena Srivastava, and M. H. Khan. "Testability Estimation of Object Oriented Design: A Revisit". International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 8, pages 3086-3090, August 2013.
[16]. Jeremy D. Miller, Designing for Testability, The Shade Tree Developer, Jun 29 2007.
[17]. Huda, M., Arya, Y.D.S. and Khan, M.H. (2014) Measuring Testability of Object Oriented Design: A Systematic Review. International Journal of Scientific Engineering and Technology (IJSET), 3, 1313-1319.
[18]. Bansiya Jagdish & Devis Carl, "Automated Metrics and Object Oriented Development", Dr. Dobb's Journal December 1997.
[19]. http://www.ddj.com/documents/s=934/ddj9712d/9712d.htm
[20]. Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Testability Quantification Framework of Object Oriented Software: A New Perspective. International Journal of Advanced Research in Computer and Communication Engineering, 4, 298- 302. http://dx.doi.org/10.17148/IJARCCE.2015.4168
[21]. Allan, K.: The Philosophy of Extensible Software. (August 2002), http://accu.org/index.php/journals/391
[22]. Metsker, S.J., Wake, W.C.: Design Patterns in Java. Addison-Wesley Professional, Reading (2006)
[23]. R. A. Khan & K. Mustafa, A review of SATC research on OO Metrics, proceedings, National Conference on Software Engineering Principles and Practices, SEPP-04, March 5-6, 2004
[24]. Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Metric Based Testability Estimation Model for Object Oriented Design: Quality Perspective. Journal of Software Engineering and Applications, 8, 234-243. http://dx.doi.org/10.4236/jsea.2015.84024
[25]. Abdullah, Dr, M. H. Khan, and Reena Srivastava. "Flexibility: A Key Factor To Testability", International Journal of Software Engineering & Applications (IJSEA), Vol.6, No.1, January 2015. DOI: 10.5121/ijsea.2015.6108.
[26]. R. A. Khan & K. Mustafa, High Level Design Quality Assessment of Object Oriented Codes, accepted for publication in the proceedings in 2nd International Workshop on Verification and Validation of Enterprise Information System VVEIS Porto, Portugal, April 13, 2004.
[27]. R. A. Khan, K. Mustafa, & S. Yadava, Quality Assessment of Object Oriented Code in Design Phase, Proceedings, QAI 4th Annual International Software Testing Conference, Pune, India Feb. 20-21, 2004.
[28]. Ellen, A., Aino, C.: How to Preserve the Benefits of Design Patterns. In: OOPSLA (1998)
[29]. Linda Rosenberg, Software Quality Metrics for Object Oriented System Environments, A report of SATC's research on OO metrics.
[30]. http://ourworld.compuserve.com/homepages/qualazur/$swmesu2.htm

[31]. Spiteri, Staines, A.: A Fundamental Modelling Concept Approach for Modelling UML Design Patterns. International Journal Of Computers 2(3) (2008)
[32]. Matthias, Z.: Evolving Software with Extensible Modules, vol. 17(5). ACM, New York (September 2005)
[33]. Raju, P., Browne, J.C.: Support for Extensibility and Reusability in a Concurrent Object Oriented Programming Language. In: IEEE-Proceedings of IPPS (1996)
[34]. I. Sommerville, Software Engineering 8. AddisonWesley Professional, 2007.
[35]. M. Zenger, "Evolving software with extensible modules," in International Workshop on Unanticipated Software Evolution, 2002.
[36]. B. Kitchenham, & Pfleeger, S. L., Software Quality: The Elusive Target, IEEE Software, 1996, 13(1): 12-21.
[37]. S.R. Chidamber, C.F. Kemerer, A metrics Suites for Object Oriented Design, IEEE Transaction on Software Engineering, Vol.20, No. 6, pp. 476-493, Jan 1994.
[38]. B. Basili, L. Briand, and W. L. Melo, A validation of Object Oriented Metrics as Quality Indicators, IEEE Trans. Softwaree Engineering, Vol.22, No. 10 pp. 751 -761, Oct-1996.
[39]. Abdullah, Dr, Reena Srivastava, and M. H. Khan. "Testability Measurement Framework: Design Phase Perspective". International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 11, Pages 8573- 8576 November 2014.
[40]. Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Evaluating Effectiveness Factor of Object Oriented Design: A Testability Perspective. International Journal of Software Engineering & Applications (IJSEA), 6, 41-49. http://dx.doi.org/10.5121/ijsea.2015.6104