International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.3, No.6, June 2014 DOI:10.15693/ijaist/2014.v3i6.81-84

# Modifiability: A Key Factor To Testability

**Abdullah**
*Research Scholar/ School of Computer Application*
*BBD University, Lucknow, India*

**Dr. Reena Srivastava**
*Dean / School of Computer Application*
*BBD University, Lucknow, India*

**Dr. M. H. Khan**
*Associate Professor/Department of Compute Science & Engineering*
*I.E.T., Lucknow, India*

*Abstract*— **Modifiability is an important key factor to testability estimation of object oriented software at an early stage of software development life cycle. Software testability is a collection of quality attributes that bear on the effort needed for validating the modified software. To design and deliver quality products within time and budget modifiability plays a vital job. This paper appreciates the need and importance of modifiability at design phase and the statistical inference establishes modifiability as an influencing factor for software testability. In this paper an endeavor has been made to establish a correlation between object oriented design constructs and testability factors as a first contribution. A model has been proposed for modifiability quantification of object oriented design by establishing multiple linear regressions. Finally the proposed model has been validated using experimental tryout.**

***Index Terms* – *Modifiability Model, Software Testability, Testability Factors, Object Oriented Design, Class Diagram, Quality Factors, Modifiability, Software Metrics.***

## I. Introduction

Improvement in software technology continues to be dynamic. New tools and techniques are regularly introduced in rapid succession. The most important inspiring issue in the development of object oriented approaches is to remove some of the flaws encountered in the procedural approach. This has forced the software developers and industry to continuously look for new approaches to high quality software design and development.

The overall intention of software engineering is to create quality oriented software that is maintainable, committed, delivered within time, budget and also satisfied its specified requirements [30, 36]. It is well known that quality is simple to feel but it is very hard to define and nearly impossible to measure accurately. Effective software testing will significantly donate to the delivery of higher quality software products, satisfied customers and will give perfect and consistent results. Testing is becoming more and more critical in view of the increasing complexity of software system as well as the highly competitive nature of the industry. It is understandable from the literature analysis; regretful to say nearly all of the industries not only fail to deliver quality software to their purchaser but rather do not recognize the appropriate quality attributes [11, 34, 35]. Producing quality software is no longer an advantage but a required factor.

The proposed study on software modifiability by us and the expert's view on this area agree on vigilant appraisal of software products for enhanced quality. Some of the quality issues that must be considered for critical evaluation are correctness, maintainability, modifiability, testability, and traceability etc.

### *A. SOFTWARE TESTABILITY*

According to ISO 9126 quality model, testability is defined as the effort needed for validating the modified software [30]. Software testability plays a key role in software design and testing. It is evident from literature survey that testing is one of the most expensive, and time taking process of development life cycle. It has been observed that the budget incurred in software testing range from 40% to 50% of the overall development budget [11, 29]. Testing is a major economic problem that requires appropriate resources and skills during the process of test design, preparation, execution and test result analysis. Testability is a quality factor, its correct estimation or evaluation can be used to predict the effort required for software testing and facilitates in distributing required resources.

## II. Related Work

### *A. Testability at Design Phase*

Testability is an important key factor and attributes for delivering high quality testable and persistent software [8]. Practitioners constantly support and advocate that testability should be planned near the beginning in the development process generally at design phase [32]. Keeping in view of the above fact, our main focus is on testability measurement at design phase that yield the highest payoff in terms of money, time and effort. It is well understood fact that a decision to modify the design in order to get better testability once the coding has started may be very costly, time consuming and error prone [26]. On the other hand, estimating testability early in the development process may significantly decrease the overall cost and attempt in measuring testability of object oriented design [11]. After taking into consideration the above actuality, reducing effort and improving software testability is an essential objective in order to reduce the total number of fault and produce high class maintainable and steady software inside time and plan. The central objective of increasing software testability is not just to detect defects but more importantly, to detect and eliminate defects as soon as they occurred [12]. Many experts in the area suggested that testability measurement should be done at design phase and the same is summarized in Table 1.

| Study/Author | Design Phase |
|---|---|
| [29]J Voas et al. (1992) | Design Phase |
| [14]JMC & Srinivas (1996) | Design Phase |
| [31]Jungmayr (2002) | Design Phase |
| [19]S. Mouchawrab (2005) | During Analysis and Design Phase |
| [6]Jerry et al. (2005) | Design Phase |
| [32]Kolb et al. (2006) | Design Phase |
| [8]D.Esposito (2008) | Design Phase |
| [9]Nazir et al. (2009) | Early in the development Process |
| [10]Nazir et al. (2010) | Design Phase |
| [12]Nazir &Khan (2012) | Design Phase |

Table: 1. Testability Measurement at Design Phase Consideration by Various Experts

***B. Testability Factor***

An accurate measure of software quality and reliability absolutely depends on testability measurement [12]. This is totally based on those factors that can influence software testability directly, especially at design phase. The testability of object oriented software should be evaluated as soon as possible, mainly as it is designed, not when coding is started or completed [6, 8, 9, 31]. Object oriented design characteristics greatly contribute to identify software testability factors that play key role to reducing effort in measuring testability of object oriented design at design phase during software development life cycle.

However, testability has always been an elusive concept and its correct measurement or evaluation is a difficult exercise [11]. It is very hard to produce a truthful view on all the factors that have impact in improving testability of object oriented software. It is evident from literature survey that there is an opposition among practitioners in taking into consideration testability factors for estimating software testability of object oriented design in general and at design phase [12]. A consolidated chart for the testability factors identified by various experts is concluded in Table 2. It is clearly evident from this Table that Observability, Controllability, Built-in-test, Traceability, Understandability and Modifiability are the commonly accepted testability factors at design phase. In this paper our focus will be exploring the effect of modifiability on testability.

| Testability Factors / Author/Study | Observability | Controllability | Built-in-test | Traceability | Understandability | Modifiability | Diagnostic capability | Fault locality | Simplicity | Complexity | Test support environment | Test suite | Development process | Implementation characteristics | Separation of concerns | Coupling |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [1]Binder (1994) | √ | √ | √ | √ | | √ | | | | | | √ | √ | √ | | |
| [2]Bach (1999) | | √ | | | √ | √ | | | √ | | | | | | | |
| [3]Jungmayr(2002) | √ | | √ | √ | | √ | √ | √ | | √ | | | | | √ | √ |
| [4]Wang(2003) | √ | √ | √ | √ | √ | | | | √ | | | | | | | |
| [5]Jungmayr (2003) | √ | | | √ | | | | | | | | | | | | |
| [6]Jerry(2005) | √ | √ | | √ | √ | √ | | | | | | | | | | |
| [7]E Mulo(2007) | √ | √ | | √ | | | | | | | | | | | | |
| [8]Dino Esposito(2008) | | √ | | √ | | √ | | | | | | | | | | |
| [9]Nazir &Khan(2009) | √ | √ | √ | √ | | √ | | | | | | | | | | |
| [10]Nazir et al. (2010 | √ | √ | √ | √ | | √ | | | | | | | | | | |
| [11]P.Malla &G(2012) | | | | √ | √ | √ | | | | | | | | | | |
| [12]Nazir et al. (2012) | √ | √ | √ | √ | | √ | | | | | | | | | | |
| [13]P.Nikfard(2013) | | √ | | √ | | √ | | | | | | | | | | |

Table 2.Testability Factors Consideration by Various Experts (Design Phase): A Critical Look

***C. Object Oriented Design Constructs***

During identification of design construct having most positive impact on testability estimation, a realistic view should be considered. If all design level parameters are considered, it may become complicated, fruitless, or insignificant." The literature analysis suggests that Abstraction, Encapsulation, Inheritance, Coupling and Cohesion are the design level constructs that quantify testability measures [10, 13]. Table 3 shows a consolidated view from the literature.

| Design Parameters / Author/Study | Cohesion | Coupling | Encapsulation | Inheritance | Abstraction |
|---|---|---|---|---|---|
| [14]MC Gregor et al. (1996) | | | √ | √ | |
| [15]Bruce & Shi(1998) | | √ | | √ | |
| [16]B.Pettichord(2002) | | √ | | | |
| [17]Baidry et al.(2002) | | √ | | | |
| [18]M Bruntik (2004) | | | | √ | |
| [19]S.Mouchawrab (2005) | √ | √ | | √ | |
| [20]I.Ahson et al.(2007) | √ | √ | √ | √ | |
| [10]Nazir et al.(2010) | √ | √ | √ | √ | |
| [21]Suhel et al.( (2012) | √ | √ | √ | √ | √ |
| [12]Khan et al. (2012) | √ | √ | √ | √ | |
| [13]Nikfard &Babak(2013) | | √ | √ | √ | |

Table 3. Object Oriented Design Constructs Contributing in Testability Measurement at Design Phase: A Critical Look

***D. Relationship between Testability Factors and Object Oriented Design Constructs***

International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.3, No.6, June 2014 DOI:10.15693/ijaist/2014.v3i6.81-84

Many experts tried to establish relationship between object oriented design construct and testability factors. A consolidated view of the same is given in Table 4. It was observed that each of these characteristics either have positive or negative impact on the factors that influence testability of object oriented design. After an in-depth evaluation of available literature on the topic [5] [6] [10] [11] [12] [15] [18], the relation between object oriented design characteristics and testability factors, shown in Figure 1 has been established.

| Author/Study | Testability Factors / Object Oriented Characteristics | Controllability | Observability | Modifiability | Understandability | Traceability | Built-in-test |
|---|---|---|---|---|---|---|---|
| [10]Nazir et al.(2010) | Encapsulation | √ | √ | | | √ | √ |
| [22]Suhel et al.(2012) | Encapsulation | √ | √ | | | √ | √ |
| [12]Khan et al.(2012) | Encapsulation | | | √ | | | |
| [28]agile | Encapsulation | | | √ | | | |
| | | | | | | | |
| [10]Nazir et al.(2010) | Inheritance | | √ | | √ | | |
| [22]Suhel et al.(2012) | Inheritance | | | √ | | | |
| [12]Khan et al.(2012) | Inheritance | | √ | | √ | | |
| | | | | | | | |
| [10]Nazir et al.(2010) | Coupling | √ | | | √ | √ | √ |
| [22]Suhel et al.(2012) | Coupling | | | √ | | | |
| [12]Khan et al.(2012) | Coupling | √ | | | √ | √ | √ |
| | | | | | | | |
| [10]Nazir et al.(2010) | Cohesion | √ | | | √ | | √ |
| [22]Khan et al.(2012) | Cohesion | √ | | | √ | | √ |

Table 4. Mapping between Object Oriented Design Constructs and Testability Factors

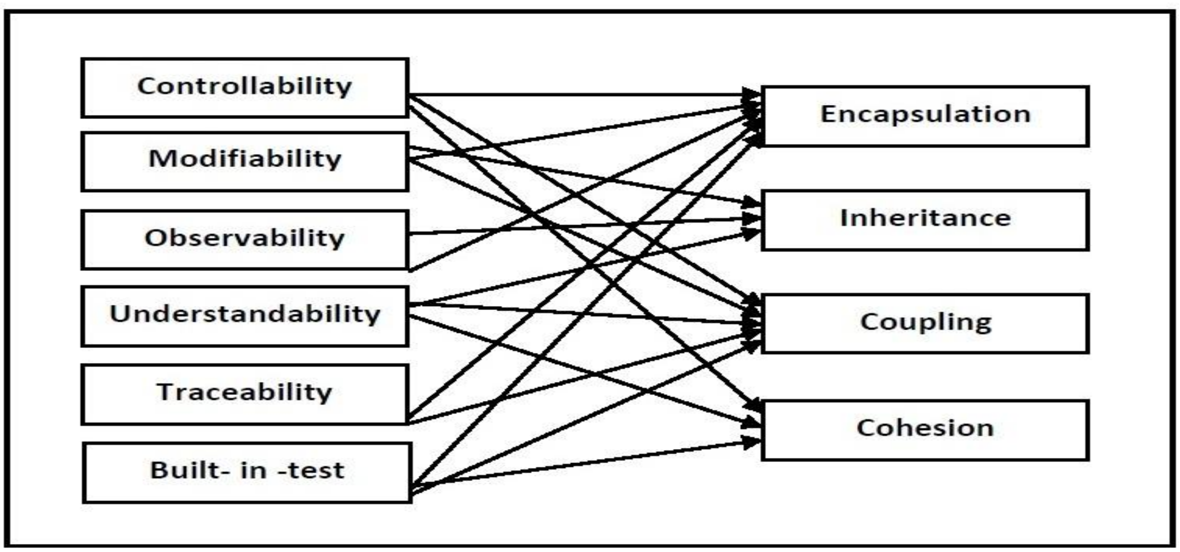


Figure 1. Relation between OOD Characteristics and Testability Factors

### *E. Relationship between Object Oriented Design Constructs and metrics*

Many researchers worked on establishing relationship between object oriented design constructs and metrics. The same has been summarized in Table 5.

| Author/Study ↓ | Object Oriented Characteristics ↓ | Suited Metrics ↓ |
|---|---|---|
| [22]Suhel et al.(2012) | **Cohesion** | **NA(Number of Attributes)** |
| | | |
| [23]Steven &W(1974) | **Coupling** | **NAsso** |
| [24]Jungmayr(2002) | | |
| [10]Nazir et al.(2010) | | |
| [9]Suhel et al.(2012) | | |
| | | |
| [5]M Bruntink & Van(2004) | **Encapsulation** | **NM(Number of Methods)** |
| [9]Khan et al.(2009) | | |
| [22]Suhel et al.(2012) | | |
| [25]Agile/CSE.ncse.edu/ | | |
| | | |
| [26]Chidember et al.(1991) | **Inheritance** | **Max DIT** |
| [27]Chidember et al.(1994) | | |
| [5]M Bruntink & Van(2004) | | |
| [10]Nazir et al.(2010) | | |
| [22]Suhel et al.(2012) | | |

Table 5. Relation between OOD Characteristics and Metrics

## III. OVERVIEW OF THE PROPOSED MODEL

### *A. Modifiability*

Modifiability is strongly related to testability and constantly plays a key role to deliver high class maintainable and trustworthy software within time and budget. It is one of the most important concepts in design and testing of software programs and components. It always supports for improved software design at early stage of software development life cycle that is to say at design phase that have positive impact on the overall testing cost and effort.

On the other hand, building programs and components with good testability and modifiability constantly improves and simplifies test operations, may really reduce overall test cost and rework during design phase and after implementation. It facilitates the creation of better quality software in time and resources. The ongoing discussion extracts the fact that modifiability is a key factor to testability for delivering high class software and control the time and cost to implement, test and deploy change.

### *B. Modifiability Quantification Model*

We have developed a modifiability quantification model that demonstrates the quantification process of software modifiability. The proposed model is shown in Figure 2. The model establishes a contextual impact relationship between modifiability and object oriented design constructs and the related metrics. The values of these metrics can be easily identified with the help of class diagram. The quantifiable evaluation of modifiability is very supportive to get testability index of software design for low cost testing and maintenance.

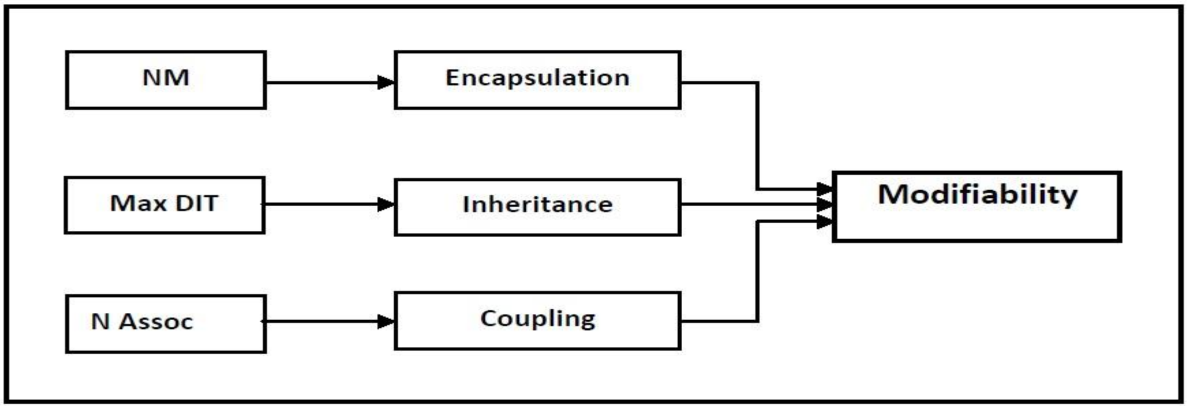


Figure 2. Modifiability Quantification Model

## IV. Model Development

### *A. Modifiability Quantification Model*

In order to set up a model for modifiability, multiple linear regression process has been used. Multivariate linear model is given below in *Eq* (1) which is as follows.

$$Y=a_0+a_1X_1+a_2X_2+a_3X_3+\text{-------}+a_nX_n \qquad Eq\ (1)$$

Where,

- **Y** is dependent variable.
- X1, X2, X3--------$X_n$ (be independent variables) associated to Y and are expected to explain the variance in Y.
- $a_1$, $a_2$, $a_3$--------$a_{n.,}$ are the coefficient of the particular independent variables.
- And $a_0$ is the intercept.

The data used for developing modifiability model is taken from [33] that have been collected through the controlled experiment. It includes a set of 28 class diagrams (denoted as D0 to D27) and the metrics value of every diagram. In addition, the mean value of the expert's rating of modifiability of these diagrams is also known and termed as 'Known Value' in this research paper.

The correlation among Testability Factors and Object Oriented Characteristics has been established as depicted in Figure 1. As per the mapping, Metrics 'NM, Max DIT, N Assoc' are elected from [33] as independent variable to develop the modifiability quantification model. Using SPSS, values of coefficient are calculated and modifiability model is formulated as given below.

**Modifiability = 1.107--.102*Encapsulation +1.810*Inheritance + .850*Coupling** *Eq* **(2)**

.

## V. Statistical Significance of the Model

The descriptive Table (Table 6) gives very valuable and considerable descriptive statistics. This Table includes mean, standard deviation etc. These records are very useful for further work.

**Table 6. Descriptive Statistics**

| | Mean | Std. Deviation | N |
|---|---|---|---|
| Modifiability | 3.0000 | 1.87083 | 5 |
| Encapsulation | 31.2000 | 32.45304 | 5 |
| Inheritance | 1.4000 | 1.14018 | 5 |
| Coupling | 3.0000 | 4.00000 | 5 |

Karl Pearson's coefficient of correlation (simple relation) method was used for measuring the degree of correlation between variables. The value of corelation 'r' lies between ±1, positive value of 'r' in Table 7 indicates positive corelation between the two variables. The value of 'r' near to +1 indicate high degree of corelation between the two variables in given Table below.

**Table 7. Correlations**

| | | Modifiability | Encapsulation | Inheritance | Coupling |
|---|---|---|---|---|---|
| Pearson Correlation | Modifiability | 1.000 | .914 | .820 | .935 |
| | Encapsulation | .914 | 1.000 | .889 | .940 |
| | Inheritance | .820 | .889 | 1.000 | .713 |
| | Coupling | .935 | .940 | .713 | 1.000 |

ANOVA Table (Table 8) highlights the result of the ANOVA examination. In this Table, I obtain F ratio of **18.862** with (**3, 1**) degree of freedom. Obtained value is larger than the critical value of F is **10.13** for the **0.05** significance level. Analyzing the significance (p-value) for the F-test in the final column of the Table, it can be concluded that the modifiability model (2) is statistically significant at a confidence level of more than **95%**.

**Table 8. ANOVA[b] for Modifiability Model**

| Model | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| 1 | Regression | 13.757 | 3 | 4.586 | 18.862 | .167[a] |
| | Residual | .243 | 1 | .243 | | |
| | Total | 14.000 | 4 | | | |

a. Predictors: (Constant), Coupling, Inheritance, Encapsulation

b. Dependent Variable: Modifiability

Furthermore the assessment of R2 (Coefficient of Determination) and adjusted R2 in the Table 9, is too very hopeful. As, it refers to the percentage or proportion of the entire variance in modifiability by all the three metrics (independent variables) participating in the model (2).

**Table 9. Model Summary**

| | | | | | Change Statistics | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | R | R Square | Adjusted R Square | Std. Error of the Estimate | R Square Change | F Change | df1 | df2 | Sig. F Change |
| 1 | .991[a] | .983 | .931 | .49306 | .983 | 18.862 | 3 | 1 | .167 |

a. Predictors: (Constant), Coupling, Inheritance, Encapsulation

## VI. Statistical Significance of Independent Variables

As long as statistical significance and importance of individual independent variables in the modifiability model (2) is concern. It can be noticed from the last column of Table 10, (p value for ‘t’ test) that all of the three metrics participating in the model is statistically significant at a significance level of 0.05 (equivalent to a confidence level of 95%).

**Table 10. Coefficients and Statistical Significance of Independent Variables**

| | | Unstandardized Coefficients | | | |
|---|---|---|---|---|---|
| Model | | B | Std. Error | t | Sig. |
| 1 | (Constant) | 1.107 | .389 | 2.849 | .215 |

| | | | | | |
|---|---|---|---|---|---|
| | Encapsulation | -.102 | .055 | -1.852 | .315 |
| | Inheritance | 1.810 | .765 | 2.365 | .255 |
| | Coupling | .850 | .292 | 2.907 | .211 |

## VII. Empirical Validation

This part of study paying concentration how the above planned model is capable to conclude the modifiability of object oriented design at design phase. The empirical validation is an essential stage of planned research to estimate modifiability quantification model for high and improved level acceptability. Empirical validation is the correct approach and practice to say the model acceptance. Keeping view of this fact, practical validation of the modifiability quantification model has been performed using sample tryouts.

In order to validate proposed modifiability quantification model the value of metrics is available (Genero et al., 2001), data sets for given ten projects shown in Table 11 taken from [33].

Table 11.Data sets for the Projects

| Project No | Encapsulation | Inheritance | Coupling |
|---|---|---|---|
| P1 | 12 | 0 | 1 |
| P2 | 8 | 0 | 1 |
| P3 | 12 | 0 | 3 |
| P4 | 38 | 4 | 0 |
| P5 | 76 | 2 | 10 |
| P6 | 98 | 4 | 12 |
| P7 | 69 | 5 | 1 |
| P8 | 73 | 1 | 9 |
| P9 | 20 | 1 | 2 |
| P10 | 47 | 2 | 6 |

The known modifiability rating for the given 10 Projects (P1-P10) is shown in Table 12.

Table 12. Known Modifiability Value

| P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 |
|---|---|---|---|---|---|---|---|---|---|
| 2.0 | 1.0 | 3.0 | 4.0 | 6.0 | 7.0 | 5.0 | 5.0 | 3.0 | 4.0 |

Table 13.Known Modifiability Rating

| P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 3 | 4 | 6 | 7 | 5 | 5 | 3 | 4 |

Using the similar set of data for the given projects (P1-P10) modifiability was calculated using proposed modifiability quantification model and the results are shown in Table 14.

Table 14. Calculated Modifiability Value Using Model.

| **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** | **P9** | **P10** |
|---|---|---|---|---|---|---|---|---|---|
| **.733** | **1.141** | **2.433** | **4.471** | **5.475** | **8.551** | **3.969** | **3.121** | **2.577** | **5.033** |

Table 15. Calculated Modifiability Rating Using Model

| P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 7 | 9 | 10 | 6 | 5 | 4 | 8 |

**Table 16. Computed Ranking, Actual Ranking and their Relation**

| **Projects →** **Modifiability Ranking ↓** | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Computed Ranking | 1 | 2 | 3 | 7 | 9 | 10 | 6 | 5 | 4 | 8 |
| Known Ranking | 2 | 1 | 3 | 4 | 6 | 7 | 5 | 5 | 3 | 4 |
| $\sum \mathbf{d}^2$ | 1 | 1 | 0 | 9 | 9 | 9 | 1 | 0 | 1 | 16 |
| $r_s$ | **.993939** | **.993939** | **1** | **.945455** | **.945455** | **.945455** | **.993939** | **1** | **.993939** | **.90303** |
| $\mathbf{r_s}$ **>.781** | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ |

Charles Speraman's Coefficient of Correlation (rank relation) $r_s$ was used to check the significance of correlation between calculated values of modifiability using model and it's 'Known Values'. Rank corelation is the process of

determining the degree of corelation between two variables. The ‘ $r_s$ ’ was computed using the formula given as under:
Speraman’s Coefficient of Correlation ($r_s$) =

$$r_s = 1 - \frac{6\sum d^2}{n(n^2-1)} \qquad -1.0 \le r_s \le +1.0$$

‘d’ = difference between ‘Calculated Values’ and ‘Known Values’ of modifiability.
n = number of UML diagrams (n=10) used in the experiment.

The correlation values between modifiability via model and known ranking are publicized in Table above. Pairs of these values with correlation values $r_s$ above [±.781] are checked in Table 16.The correlation are up to standard with high degree of confidence, i.e.at the 99%. Therefore we can conclude without any loss of generality that modifiability quantification model estimates are extremely trustworthy, important and applicable in the perspective.Howeve, the study needs to be standardized with a large experiment tryout for better acceptability and utility.

## VIII. Conclusion

This paper shows the significance of modifiability in general and as a key factor to software testability for producing high class software within time and budget. Modifiability is evidently highly appropriate and significant in the perspective of software testability. Modifiability model is developed with the help of multiple linear regression process on object oriented design characteristics. Statistical examination shows that this model is statistically greatly significance and acceptable. Modifiability quantification model has been validated theoretically as well as empirically using experimental try-out. The applied validation on the modifiability model concludes that proposed model is highly consistent, acceptable and considerable.

## REFERENCES

[1] Binder, Robert V. "Design for testability in object-oriented systems." *Communications of the ACM* 37.9 (1994): 87-101.
[2] Bach, James."Heuristics of Software Testability" (1999).
[3]*Testability during Design*. Softwaretechnik-Trends, Proceedings of the GI Working Group Test, Analysis and Verification of Software, Potsdam, June 20th - 21th, 2002, pp. 10-11.
[4]Y. Wang, “Design for Test and Software Testability”, University of Calgary, 2003.
http://www.ucalgary.ca/~ageras/wshop/abstracts/2003/design-for-estability.htm
[5] Jungmayr, Stefan. "Testability during Design, Software Technik-Trends." *Proceedings of the GI Working Group Test, Analysis and Verification of Software, Potsdam* (2002): 10-11.
[6]J.Gao and M. C. Shih, “A component testability model for verification and measurement”, In Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC ’05), pages 211–218.IEEE Computer Society, 2005.
[7]E. Mulo, “Design for Testability in Software Systems”, Master’s Thesis, 2007.
[8]D. Esposito, “Design Your Classes for Testability”, 2008. http://dotnetslackers.com/articles/n net/Design-Your-Classes-for-Testability.aspx
[9]Nazir M & Khan R A (2009): Software Design Testability Factors: A New Perspective, Proceedings, 3rd National Conference: INDIACom-2009, Bharti Vidya Peeth Institute of Computer Application and Management, New Delhi, Feb 26-27, pp.323-328.
[10]M. Nazir, Khan R A & Mustafa K. (2010): A Metrics Based Model for Understandability Quantification, Journal of Computing, Vol. 2, Issue 4, April 2010, pp.90-94.
[11] M. Nazir, Khan R A & Mustafa K. (2010): Testability Estimation Framework, International Journal of Computer Application, Vol. 2, No. 5, pp.9-14. *June 2010.*
[12]Nazir, Mohd, and Raees A. Khan. "Testability Estimation Model (TEMOOD)."*Advances in Computer Science and Information Technology.Computer Science and Information Technology*. Springer Berlin Heidelberg, 2012. 178-187.
[13]http://ieeexplore.ieee.org/stamp/stamp.jsp
[14]McGregor, John D., and Satyaprasad Srinivas. "A measure of testing effort."*Proceedings of the 2nd conference on USENIX Conference on Object-Oriented Technologies (COOTS)-Volume 2*. USENIX Association, 1996.
[15]B. W. N. Lo and H. Shi, “A preliminary testability model for object-oriented software,” In Proc. Int. Conf. on Software Engineering, Education, Practice, pages 330-337. IEEE. 1998.
[16]Pettichord, Bret. "Design for testability."*Pacific Northwest Software Quality Conference*. 2002.
[17]B. Baudry, Y. Le Traon, and G. Sunyé, “Testability Analysis of a UML Class diagram”, Proceedings of the Eighth IEEE Symposium on Software Metrics [METRICS.02], IEEE 2002
[18]Bruntink, Magiel, and Arie Van Deursen. "Predicting class testability using object-oriented metrics." *Source Code Analysis and Manipulation, 2004. Fourth IEEE International Workshop on*. IEEE, 2004.
[19]Mouchawrab, Samar, Lionel C. Briand, and Yvan Labiche. "A measurement framework for object-oriented software testability." *Information and software technology* 47.15 (2005): 979-997.
[20]Khan, R. A., K. Mustafa, and S. I. Ahson. "An empirical validation of object oriented design quality metrics." *Journal of King Saud University-Computer and Information Sciences* 19 (2007): 1-16.
[21]Khan, Suhel Ahmad, and Raees Ahmad Khan. "Analyzability Quantification Model of Object Oriented Design." *Procedia Technology* 4 (2012): 536-542.
[22] Suhel Ahmad Khan, Raees Ahmad Khan." Addressing Analyzability in Terms of Object Oriented Design Complexity." Proceedings of the 48th Annual Convention of Computer Society of

India- Vol II Advances in Intelligent Systems and Computing Volume 249, 2014, pp 371-378

[23]Stevens, Wayne P., Glenford J. Myers, and Larry L. Constantine. "Structured design." *IBM Systems Journal* 13.2 (1974): 115-139.

[24]Jungmayr, Stefan. "Testability measurement and software dependencies." *Proceedings of the 12th International Workshop on Software Measurement*. 2002.

[25] Agile/CSE.ncse.edu/

[26] Abdullah, Dr. Reena Srivastava,Dr. M. H. Khan "Testability Estimation of Object Oriented Design: A Revisit", in IJARCCE, Volume 2, Issue 8, Pages 3086-3090, August 2013.

[27] Chidamber, S. R. and Kemerer, C. F., "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, vol. 20, 1994.

[28] http://www.agilemodeling.com/essays/simpleTools.htm

[29] J Voas and Miller , "Improving the software development process using testability research", Proceedings of the 3rd international symposium on software Reliability Engineering , p. 114--121, October, 1992, RTP, NC, Publisher: IEEE Computer Society.

[30]ISO.International standard ISO/IEC 9126.information technology: Software product evaluation: quality characteristics and guidelines for their use, 1991

[31]Testability during Design. Softwaretechnik-Trends, Proceedings of the GI Working Group Test, Analysis and Verification of Software, Potsdam, June 20th - 21th, 2002, pp. 10-11.

[32]Kolb, Ronny, and Dirk Muthig. "Making testing product lines more efficient by improving the testability of product line architectures." *Proceedings of the ISSTA 2006 workshop on Role of software architecture for testing and analysis*. ACM, 2006.

[33]M. Genero, J. Olivas, M. Piattini, and F. Romero, "A Controlled Experiment for Corroborating the Usefulness of Class Diagram Metrics at the Early Phases of Object-Oriented Developments," *Proc. of the ADIS 2001, Workshop on Decision Support in Software Engineering*, vol. 84. Spain, 2001.

[34] Dromey, R.G.: A Model for Software Product Quality. IEEE Transaction on Software Engineering 21(2), 146–162 (1995)

[35] Behkamal, Behshid, Mohsen Kahani, and Mohammad Kazem Akbari. "Customizing ISO 9126 quality model for evaluation of B2B applications." *Information and software technology* 51.3 (2009): 599-609.

[36]Mao, Chengying, Yansheng Lu, and Jinlong Zhang. "Regression testing for component-based software via built-in test design." *Proceedings of the 2007 ACM symposium on Applied computing*. ACM, 2007.

### Authors Profile

**Abdullah** received the MCA degree from Uttar Pradesh Technical University, Lucknow, in 2006. He is currently pursuing Ph.D. in the field of software engineering from BBD University Lucknow UP. His research interests Include Software testability, Software Quality Estimation. He has written various books and study materials for Bharati Vidyapeeth University ('A' Grade by Ministry of HRD, Government of India .Accreditation & Re-Accreditation with 'A' Grade by NAAC, India) Pune, North Orissa University Orissa, Suresh Gyan Vihar University, Jaipur, Rajasthan.

**Dr. Reena Srivastava** is currently working as Dean, School of Computer Applications at BBD University. She received her Ph.D. degree from MNNIT Allahabad, India. Her research area includes Multi-Relational Classification, Privacy Preserving Data Mining and Software Engineering.

**Dr. M. H. Khan** Associate Professor, Department of Computer Science and Engineering at IET Lucknow UP.
Obtained his MCA degree from Aligarh Muslim University (Central University) in 1991 .Later he did his PhD from Lucknow University. He has around 24 years rich teaching experience at UG and PG level. His area of research is Software Engineering. Dr. Khan published numerous articles, several papers in the National and International Journals and conference proceedings.