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

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 3, March 2015*

# Maintainability Estimation of Object Oriented Software: Design Phase Perspective

**Nupur Soni[1], Dr. Mazhar Khaliq[2]**

Assistant Professor, School of Computer Application, BBDU, Lucknow, India[1]

Assistant Professor, Department of Computer Science, K.M.C.A.F.U., Lucknow, India[2]

**Abstract**: Maintainability is an essential software quality factor that is useless if it is not available at an initial stage in the software development life cycle. It becomes more important in the case of object oriented design. Estimating maintainability of object oriented design near the beginning in the development cycle, mainly at design phase; significantly reduce the development cost and rework, and as well as assists the software designers and developers for delivering high quality maintainable software within time and budget. This paper illustrates the need and significance of maintainability at design phase and build up a multivariate linear Maintainability Estimation Model for Object-Oriented Design. Developed model estimates the maintainability of class diagrams in respect of their Extendibility, Reusability. Lastly the developed models have been validated using experimental tryout.

**Keywords**: Maintainability, Estimation, Software Metrics, Object Oriented Design.

## I. INTRODUCTION

Software is going away to be changed several times for different reasons while being developed and particularly after it has been delivered. Commonly the term maintenance is used when referring to those changes made to software products after they have been delivered Depending on the reasons for alteration and the wider organizational perspective, a variety of approaches to maintenance such as corrective or adaptive maintenance are or relatively should be applied. Despite the truth that software maintenance is a costly and difficult task; it is not correctly managed and often unnoticed. One cause for this poor management is the lack of established measures for software maintainability [1]. As an outcome, there is an imperative demand to put into practice software engineering concepts, strategy, practices to avoid deviation, and to improve the software development process in order to deliver good quality maintainable software in time and within account.

## II. SOFTWARE MAINTAINABILITY

The key word of "maintainability" appeared in the categorization of maintenance. It is also programmed as the first key attribute of good designed software [2].Maintainability is one of the most significant characteristics of software quality. The mass of software companies splurge 60 to 70 percent of resources for correcting, adopting and maintaining the existing software [3, 4, 7]. The majority of companies pay out over 70 percent budget on testing, maintenance of the software to manage the quality [5].Maintainability Estimation helps to examine the maintenance effort and easiness of software at design level [14]. The maintainability definition according to IEEE glossary of Software Engineering is *"the ease with which a software system or component can be modified to correct faults, get better performance or other attributes, or adapt to a change environment"[6,23]*. Software maintenance required for extra effort than any other software engineering activity [5]. The maintainability of software is not possible directly, but with the help of their internal characteristics Estimation [6]. Noticeably a definition of maintainability wishes to be strongly link to the term maintenance. Maintainability is the easiness or simplicity with which a software system can be maintained (using the definition of software maintenance above) and is a key characteristic of software [16, 18].

## III. OBJECT ORIENTED PROPERTIES

Object oriented design is the most popular concept in today's software development environment. Object oriented system consider object as the primary agent involved in a computation process. It requires more significant effort at the early phase in the software development life cycle to recognized objects, classes, and the relationships among them. Object oriented programming is a basic knowledge that supports quality objectives [13, 15].

The necessity to deal with the maintainability of software design is the essential issue that influenced the overall development cost and quality. A good object oriented design needs design procedures and practices that must be used in development cycle [17]. Their violation will ultimately have a strong impact on the quality attributes. Object oriented principles direct the designers what to hold up and what to keep away from. A number of measures have been defined so far to measure object oriented design. There are several important themes of object orientation that are known to be the basis of internal quality of object oriented design and support in the perspective of measurement [18,19]. These themes significantly include inheritance, encapsulation, cohesion and coupling.

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

IJARCCE

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 3, March 2015*

## IV. OBJECT ORIENTED DESIGN METRICS

The most central aim of metric selection is to pick such metrics which are statistically important and must be applicable. Studies have been conducted and found that there exists powerful relation among Object Oriented software metrics and its maintainability. Software metrics offer an effortless and inexpensive way to identify and correct probable reasons for low software quality according to the maintainability sub -factor as this will be supposed by the programmers.

Set up Estimation programs and design metric standards will support in preventing failures before the maintenance process and decrease the essential effort during that phase. Internal metrics are extremely associated with the programmers' view of maintainability [9-12]. However, unhappiness with internal quality standards may not necessarily outcome in low rank of maintainability although it is generally expected. In that case, it is likeable that, regardless of what internal Estimations designate, the concluding judge for the maintainability of the delivered software is the programmer [19, 21, 22].

## V. MODELS DEVELOPMENT

The generic quality model [4, 11, 23] has been considered as a foundation to develop the Maintainability Measurements model for object oriented software design.

Estimation of class diagram's Extendibility and Reusability is prerequisite for the accurate maintainability Estimation. For this reason prior to developing MEM$^{OOD}$, the study has developed two models for Extendibility and Reusability. In order to set up all the two models subsequent multivariate linear model (1) has selected.

$$Y = \mu + \beta_1 * X_1 + \beta_2 * X_2 + \ldots\ldots + \beta_n * X_n + \varepsilon \quad (1)$$

Where

- Y is dependent variables.
- X1, X2… Xn are independent variables.
- ß1, ß 2… ß n are the coefficients.
- € is error term
- µ is the intercept.

## VI. EXTENDIBILITY ESTIMATION MODEL

In order to set up an Extendibility estimation model of object oriented class diagram, metrics listed in [8] will play the role of independent variables while Extendibility will be taken as dependent variable. The data used for developing Extendibility model is taken from [8]. The correlation among Maintainability Factors and Object Oriented Characteristics has been established as depicted in equation2.

Using SPSS, values of coefficient are calculated and Extendibility model is originated as below.

**Extendibility=2.970 - 5.276 × Coupling + 14.054 × Inheritance + 1.517 × Polymorphism** **(2)**

Table1:Coefficients[a]

| Model | | Unstandardized Coefficients | | Standardized Coefficients | t | Sig. |
|---|---|---|---|---|---|---|
| | | B | Std. Error | Beta | | |
| 1 | (Constant) | 2.970 | 3.090 | | .961 | .438 |
| | Coupling | -5.276 | 2.686 | -.384 | -1.964 | .188 |
| | Inheritance | 14.054 | 4.535 | .623 | 3.099 | .090 |
| | Polymorphism | 1.517 | .886 | .285 | 1.711 | .229 |

a. Dependent Variable: Extendibility

Table 2:**Descriptive Statistics**

| | Mean | Std. Deviation |
|---|---|---|
| Extendibility | 8.0000 | 6.78970 |
| Coupling | .7617 | .49358 |
| Inheritance | .4333 | .30111 |
| Polymorphism | 1.9500 | 1.27554 |

Table 3:**Model Summary**

| Model | R | R Square | Adjusted R Square | Std. Error of the Estimate | Change Statistics | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | R Square Change | F Change | df1 | df2 | Sig. F Change |
| 1 | .983[a] | .966 | .915 | 1.97920 | .966 | 18.948 | 3 | 2 | .051 |

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

Table 4:ANOVA[b]

| Model | | Sum of Squares | df | Mean Square | F | Sig. |
|---|---|---|---|---|---|---|
| 1 | Regression | 222.666 | 3 | 74.222 | 18.948 | .051[a] |
| | Residual | 7.834 | 2 | 3.917 | | |
| | Total | 230.500 | 5 | | | |

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

b. Dependent Variable: Extendibility

ANOVA Table emphasizes the result of the ANOVA examination. In this Table, we obtain F ratio of 18.948 with (**3, 2**) degree of freedom. Obtained value is larger than the critical value of F is **9.55** for the **0.05** significance level

## VII. REUSABILITY ESTIMATION MODEL

The data used for developing Reusability model is taken from [33]. The correlation among Reusability Factors and Object Oriented Characteristics has been established as depicted in equation 3.

Applying the same technique of stepwise backward multiple regressions on the available data resulted into the following Reusability model (3).

**Reusability= 4.093 - 4.022 × Coupling + .328 × Cohesion + .370 × Design size** **(3)**


DOI 10.17148/IJARCCE.2015.43130

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

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 3, March 2015*

| Table 5:Coefficients[a] | | | | | | |
|---|---|---|---|---|---|---|
| Model | | Unstandardized Coefficients | | Standardized Coefficients | t | Sig. |
| | | B | Std. Error | Beta | | |
| 1 | (Constant) | 4.093 | 5.354 | | .764 | .525 |
| | Coupling | -4.022 | 1.676 | -.530 | -2.400 | .138 |
| | Cohesion | .328 | 8.095 | .009 | .041 | .971 |
| | Design size | .370 | .091 | .772 | 4.060 | .056 |
| a. Dependent Variable: Reusability | | | | | | |

| Table 6:**Descriptive Statistics** | | |
|---|---|---|
| | Mean | Std. Deviation |
| Reuseability | 6.5167 | 3.74882 |
| Coupling | .7617 | .49358 |
| Cohesion | .5550 | .10271 |
| Design size | 14.3333 | 7.81452 |

| Table 7:**Model Summary** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | R | R Square | Adjusted R Square | Std. Error of the Estimate | Change Statistics | | | | |
| | | | | | R Square Change | F Change | df1 | df2 | Sig. F Change |
| 1 | .964[a] | .929 | .822 | 1.58258 | .929 | 8.685 | 3 | 2 | .105 |
| a. Predictors: (Constant), Design size, Coupling, Cohesion | | | | | | | | | |

| Table 8:**ANOVA**[b] | | | | | | |
|---|---|---|---|---|---|---|
| Model | | Sum of Squares | df | Mean Square | F | Sig. |
| 1 | Regression | 65.259 | 3 | 21.753 | 21.685 | .105[a] |
| | Residual | 5.009 | 2 | 2.505 | | |
| | Total | 70.268 | 5 | | | |
| a. Predictors: (Constant), Design size, Coupling, Cohesion | | | | | | |
| b. Dependent Variable: Reuseability | | | | | | |

ANOVA Table emphasizes the result of the ANOVA examination. In this Table, we obtain F ratio of 18.948 with (**3, 2**) degree of freedom. Obtained value is larger than the critical value of F is **9.55** for the **0.05** significance level

## V. MAINTAINABILITY ESTIMATION MODEL

Before developing the model for maintainability, it is important to make sure the appropriate association among testability, extendibility and reuseability of class diagrams. Table 9, shows the relationship values among them. From the correlation values it is clear that both extendibility and reuseability are strongly correlated with maintainability.

| Table 9:**Correlations** | | | | |
|---|---|---|---|---|
| | | Maintainability | Extendibility | Reuseability |
| Pearson Correlation | Maintainability | 1.000 | .820 | .790 |
| | Extendibility | .820 | 1.000 | .995 |
| | Reuseability | .790 | .995 | 1.000 |

**Maintainability= -1.861 -1.137 × Extendibility +.960 × Reuseability (4)**

| Table 10:**Coefficients**[a] | | | | | | |
|---|---|---|---|---|---|---|
| Model | | Unstandardized Coefficients | | Standardized Coefficients | t | Sig. |
| | | B | Std. Error | Beta | | |
| 1 | (Constant) | -1.861 | 6.558 | | -.284 | .824 |
| | Extendibility | -1.137 | 1.783 | -3.234 | -.638 | .639 |
| | Reuseability | .960 | 2.005 | 2.427 | .479 | .716 |
| a. Dependent Variable: Maintainability | | | | | | |

| Table 11:**Descriptive Statistics** | | |
|---|---|---|
| | Mean | Std. Deviation |
| Maintainability | 1.2833 | .68838 |
| Extendibility | -.0305 | 1.95844 |
| Reuseability | 3.2395 | 1.74080 |

| Table 12:**Model Summary** | | | | |
|---|---|---|---|---|
| Model | R | R Square | Adjusted R Square | Std. Error of the Estimate |
| 1 | .856[a] | .733 | .199 | .61595 |
| a. Predictors: (Constant), Reuseability, Extendibility | | | | |

## VI. EMPIRICAL VALIDATION

Empirical validation is a vital phase of proposed research. Empirical validation is the standard approach to justify the model approval. Taking view of this truth, practical validation of the maintainability model has been performed using sample tryouts. In order to validate developed maintainability model the data has been taken from [8].

| Table13: Computed Ranking, Actual Ranking and their Relation | | | | | |
|---|---|---|---|---|---|
| Projects ↓ | Maintainability Ranking | | $\sum d^2$ | $r_s$ | $r_s$> ±.781 |
| | Computed Rank | Known Rank | | | |
| p1 | 9 | 10 | 1 | 0.99 | ✓ |
| p2 | 7 | 9 | 4 | 0.98 | ✓ |
| p3 | 8 | 2 | 36 | 0.78 | ✓ |
| p4 | 10 | 7 | 9 | 0.95 | ✓ |
| p5 | 6 | 3 | 9 | 0.95 | ✓ |
| p6 | 1 | 4 | 9 | 0.95 | ✓ |
| p7 | 3 | 5 | 4 | 0.98 | ✓ |

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

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 3, March 2015*

| | | | | | |
|---|---|---|---|---|---|
| p8 | 5 | 1 | 16 | 0.90 | ✓ |
| p9 | 4 | 8 | 16 | 0.90 | ✓ |
| p10 | 2 | 6 | 16 | 0.90 | ✓ |

Speraman's Coefficient of Correlation $r_s$ was used to check the significance of correlation among calculated values of maintainability using model and it's 'Known Values'. The '$r_s$' was estimated using the method given as under: Speraman's Coefficient of Correlation

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

'd' = difference between 'Calculated ranking' and 'Known ranking' of testability.
n = number of projects used in the experiment.

The correlation values between maintainability through model and known ranking are shown in table (13) above. Pairs of these values with correlation values $r_s$ above [±.781] are checked in table. The correlations are up to standard with high degree of confidence, i.e. up to 99%. Therefore we can conclude without any loss of generality that maintainability Estimation model measures are really reliable and significant and applicable.

## VII. CONCLUSION

The study has developed three models to compute extendibility, reusability and maintainability of the class diagrams. Maintainability model measures the maintainability of class diagrams in terms of their extendibility and reusability. All the three models have been developed using the method of multiple linear regressions. The study moreover validates the quantifying ability of maintainability model. The applied validation on the maintainability model concludes that proposed model is highly consistent, acceptable and considerable. The values of extendibility, reusability and maintainability are of instant use in the software development process. These values help software designers to review the design and take proper corrective measures, early in the development cycle, in order to control or at least reduce future maintenance cost. The maintenance team may also utilize this information to know, on what module to center during maintenance.

## REFERENCES

[1] M. Dagpinar and J. Jahnke, "Predicting Maintainability with Object- Oriented Metrics – an Empirical Comparison," Proc. 10th Working Conference on Reverse Engineering (WCRE'03), 13 - 17 Nov. 2003, pp. 155 -164, 2003.
[2] 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.
[3] P. Antonellis, D. Antoniou, Y. Kanellopoulos, C. Makris, E. Theodoridis,C. Tjortjis, and N. Tsirakis, "A Data Mining Methodology for Evaluating Maintainability According to ISO/IEC-9126 Software Engineering Product Quality Standard," Proc. 11th IEEE Conference onSoftware Maintenance and Reengineering (CSMR2007), 21 – 23 Mar.2007, Amsterdam, Netherlands, 2007.
[4] M. Genero, E. Manso, A. Visaggio, and M. Piattini, "Building Measure-Based Prediction Models for UML Class Diagram Maintainability," Journal of Empirical Software Engineering, vol. 12, no. 5, pp. 517 -549, 2007.
[5] J.H. Hayes and L Zhao, "Maintainability Prediction: a Regression Analysis of Measures of Evolving Systems," Proc. 21st IEEE International Conference on Software Maintenance, 26 - 29 Sept. 2005, pp. 601 -604, 2005.
[6] Abdullah, Dr, M. H. Khan, and Reena Srivastava. "Testability Measurement Model for Object Oriented Design ($TMM^{OOD}$)." International Journal of Computer Science & Information Technology (IJCSIT) Vol. 7, No 1, February 2015, DOI: 10.5121/ijcsit.2015.7115.
[7] M.O. Elish and K.O. Elish, "Application of TreeNet in Predicting Object-Oriented Software Maintainability: a Comparative Study," Proc. of European Conference on Software Maintenance and Reengineering (CSMR'09), 24 - 27 Mar., 2009, pp. 69 - 78, 2009.
[8] IEEE Press, "IEEE Standard Glossary of Software Engineering Technology,"ANSI/IEEE Standard 610.12-1990, 1990.
[9] M. Polo, M. Piattini, and F. Ruiz, "Using Code Metrics to Predict Maintenance of Legacy Programs: a Case Study," Proc. of International Conference on Software Maintenance, ICSM 2001, pp. 202-208, IEEE Computer Society, Florence Italy, 2001.
[10] 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.
[11] Jagdish Bansiya, "A Hierarchical Model for Object Oriented Design Quality Assessment", IEEE Transaction of Software Engineering, Volume 28, No. 1, January 2002, and pp: 4-17
[12] J.H. Hayes, S.C. Patel, and L. Zhao, "A Metrics-Based Software Maintenance Effort Model," Proc. 8th European Conference on Software Maintenance and Reengineering (CSMR'04), 24 – 26 Mar. 2004, pp. 254 – 258, IEEE Computer Society, 2004.
[13] W. Li and S. Henry, "Object-Oriented Metrics that Predict Maintainability," Journal of Systems and Software, vol. 23, no. 2, pp. 111 - 122, 1993.
[14] 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.
[15] M. Perepletchikov, C. Ryan, K. Frampton, and Z. Tari, "Coupling Metrics for Predicting Maintainability in Service-Oriented Designs," Proc. of Australian Software Engineering Conference (ASWEC'07), 10 – 13 April, 2007, pp. 329 – 340, Melbourne, Australia, 2007.
[16] Mahfuzul Huda, Dr.Y.D.S.Arya, and Dr. M. H. Khan. "Evaluating Effectiveness Factor of Object Oriented Design: A Testability Perspective." International Journal of Software Engineering & Applications (IJSEA), Vol.6, No.1, January 2015, DOI:10.5121/ijsea.2015.6104.
[17] ISO, "International standard ISO/IEC 9126. Information technology: Software product evaluation: Quality characteristics and guidelines for their use." 1991.
[18] S. Muthanna, K. Kontogiannis, K. Ponnambalam, and B. Stacey, "A Maintainability Model for Industrial Software Systems Using Design Level Metrics," Proc. 7th Working Conference on Reverse Engineering (WCRE'00), 23 - 25 Nov., 2000, pp. 248 – 256, Brisbane, Australia, 2000.
[19] 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.
[20] M. Kiewkanya, N. Jindasawat, and P. Muenchaisri, "A Methodology for Constructing Maintainability Model of Object-Oriented Design," Proc. 4th International Conference on Quality Software, 8 - 9 Sept., 2004, pp. 206 - 213. IEEE Computer Society, 2004.
[21] Mahfuzul Huda, Dr.Y.D.S.Arya, and Dr. M. H. Khan. "Testability Quantification Framework of Object Oriented Software: A New Perspective." International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 1, Jan, 2015, DOI: 10.17148/IJARCCE.2015.4168.
[22] S.S. Dahiya, J.K. Chabra, and S. Kumar, "Use of Genetic Algorithm for Software Maintainability Metrics Conditioning,"

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

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 4, Issue 3, March 2015*

Proc. 15th International Conference on Advanced Computing and Communications, pp. 87 -92, 200.

[23] Mahfuzul Huda, Dr.Y.D.S.Arya, and Dr.M. H. Khan. “Measuring Testability of Object Oriented Design: A Systematic Review.” International Journal of Scientific Engineering and Technology, Vol. 3, Issue 10, pp: 1313-1319 Oct, 2014.

[24] M. Perepletchikov, C. Ryan, K. Frampton, and Z. Tari, “Coupling Metrics for Predicting Maintainability in Service-Oriented Designs,” Proc. of Australian Software Engineering Conference (ASWEC'07), 10 - 13April, 2007, pp. 329 – 340, Melbourne, Australia, 2007.

[25] d H.M. Olague, L.H. Etzkorn, S.L. Messimer, and H.S. Delugach, “An Empirical Validation of Object-Oriented Class Complexity Metrics and their Ability to Predict Error-prone Classes in Highly Iterative, or Agile Software: a Case Study,” Journal of Software Maintenance, vol. 20, no. 3, pp. 171 - 197, 2008.

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