CHANGEABILITY EVALUATION MODEL FOR OBJECT ORIENTED SOFTWARE

Nidhi Goyal[1] and Reena Srivastava[2]

[1]Ph.D, Scholar , BBDU, Lucknow India
[2]Dean, School of Computer Applications, BBDU, Lucknow, India

ABSTRACT

Changeability has a direct relation to software maintainability and has a major role in providing high quality maintainable and trustworthy software. The concept of Changeability is a major factor when we design and develop software and its constituents. Developing programs and its constituent components with good changeability continually improves and simplifies test operations and maintenance during and after implementation. It encourages and supports improvement in software quality at design stage in the development of software. The research here highlights the importance of changeability broadly and also as an important aspect of software quality.

In this paper a correlation between the major attributes of object oriented design and changeability has been ascertained. A changeability evaluation model using multiple linear regression has been proposed for object oriented design. The validation of the proposed changeability evaluation model is made known by means of experimental tests and the results show that the model is highly significant.

KEYWORDS

Object Oriented Design, Changeability, Maintainability Evaluation, Software quality, Empirical validation.

1. INTRODUCTION

The nature and intricacy of software has changed significantly in the last two decades. The software industry has seen a lot of changes since its beginning. The omnipresent internet and ubiquity of network based software along with huge transformation in computers has made the software industry a force to reckon with and the progress has been brisk and tremendous. Creating software that is fault free is a major roadblock plaguing the software industry. Software industry has been witnessing exponential increase in price as well as performance but still there is little decline in numerous issues with software.

The excellence in talent in programming alone is insufficient for making bulky programs. The factors of price, maintenance, time resources and quality pose a serious problem in many software products. Software Industry has evolved into an area where its survival and growth are directly linked to the effective utilization and maintenance of its products within strict limits of resources, time and budgets [1, 2]. Delivering on these constraints requires the delivery of robust software that is resilient and adaptable.

The method of altering the software that has been already delivered is called software maintenance and the effortlessness with which this can be achieved is defined as software maintainability [3]. Software changeability is a significant part of maintainability, particularly in

DOI:10.5121/ijcsit.2017.9403

surroundings where software changes are numerous. High- level designs have an impact on maintainability which influences changeability as it is a sub factor of it as defined in ISO-9152. The importance of maintainability and changeability of software can no longer be ignored or underestimated.

The main concept of "changeability," is combination of three things namely change agents, the effects of change and change mechanisms. The making of changeable systems facilitates higher chances of maintaining value delivery over different phases of system lifecycle, providing value robustness to software [4]. The software has to be modified in accordance with the change in requirements of customers that may happen due to reasons as varied as a sudden change in technology, new hardware, changing socio-economic environments or features enhancements. Producing software(s) that do not handle change well can be very uneconomical and unviable. Software changeability is an arduous and expensive work but still its proper management most of the time is not given due consideration. One of the prime reasons for this is the inadequacy of established measures for software changeability. [5].

Changeability of the software has a correlation and influence on the parameters that directly or indirectly enhance software quality. Complexity in design may perhaps result in ineffective changeability, thus escalating unproductive software testing causing liability of severe penalties and other consequences like loss of credibility. Defects in design structure and metrics propagate a powerful cascading negative impact on quality aspect. Even then, proper structuring of a quality oriented design remains a poorly defined process [6]. Consequently, object oriented design should be built in such a way so as to make them effortlessly understandable, changeable, adaptable, and preferably stable.

The intricacy and the need to conform can complicate incorporating changes in software, if not thought over and incorporated early during design phase itself. Its early estimate lays down the foundation of making possible as well as easing out a software maintenance procedure. Consequently, it is a quality of the software that requires close up development cooperation with software maintenance [7].

2. EXISTING LITERATURE ON SOFTWARE CHANGEABILITY

Researchers, practitioners and quality controllers emphasize on the need of having a systematic approach for changeability measurement. They argue that changeability can be measured at design phase by assessing the design level metrics of changeability. The contextual findings of related work on software changeability and the approaches available for its measurement may be summarized as follows:

In a study by H. Kabaili et al. (2001), the authors have discussed cohesion as a changeability indicator in SDLC phases. In this work, authors explored whether there exists a correlation between cohesion and changeability. Two cohesion metrics, LCC and LCOM were considered by author for estimating software changeability and a model of change impact was applied. To test the hypothesis that cohesion and changeability are correlated, researchers inspected the well-known cohesion metrics, LCC but due to deficit of resources, were unable to examine the complete list of proposed sixty six changes of their impact model for object oriented language C++. Rather, researchers limited themselves to a subset of six changes which they preferred according to a set of four chosen criteria. They could not come up with a strong correlation between cohesion and changeability. They could not prove with conformity that defined cohesion metrics were good indicator of changeability [8].

Study done by M. Ajmal Chaumun (2002) discussed Changeability in terms of measuring change impact while considering correlation coefficient among two variables and a WMC metric. After deleting the outliers cases, the correlation coefficients was found to be weak and of the order of 0.55. Using Anova test they were though able to support that WMC metric and change impact is related. They applied their study in application areas such as telecommunications. Using change Impact analysis for assessing changeability of software systems they were able to derive a limited model to implement successful system compilation after changes. A generalized model for performance of changeability was though not proposed by them. [9].

Study done by M. K. Abdi et al. (2009) proposed a probabilistic method having Bayesian networks as opposed to earlier non-probabilistic approaches to help analyze change impact in object oriented systems. They primarily used coupling measurements like coupling between objects including classes used by target, coupling with no ancestors and so on to verify their approach. They used three scenarios in which a correlation hypothesis amid different metrics of coupling and the change impact that had been previously established in former works. In these three scenarios the change impact was found to be weak, of the order of 0.46, 0.48, 0. 54. In the fourth scenario the results in relationship proposed between these metrics and change impact contradicted to earlier results [10], leading them to search a hypothesis explaining factors like complexity and system size. This work suggests methods for improving maintenance of software in object oriented systems and focuses on change impact analysis in generic SDLC phases [11].
Yirsaw Ayalew et al. (2013) used cases on open source software and tried to explore impact of coupling and complexity metric in changeability and assess modularity of the system. The authors used three coupling metrics as indicators of changeability on open source software and showed that coupling metrics may be good indicators of changeability [12]. In their work the authors provided theoretical approach for measuring changeability and extensibility of aspect oriented software. Moreover, no quantitative changeability measurement model has been provided in this work.

In the paper by Sun et al. (2012) an approach was developed to estimate a software system's changeability using two steps. The first method was using the formal theory analysis to do change impact analysis (CIA) that estimated the cascading effect of the proposed alteration .The author further proposed a new impact metric to demonstrate the capability in the system to absorb these changes. Study on three case application programs showed the usefulness of proposed approach of changeability evaluation. Depending on a questionnaires analysis, the study classified the change impact analysis (CIA) according to their impact on system changeability [13]. Further based on outcomes, author proposed guidelines for making design decisions, and provides theoretical guidelines to improve system changeability. In this study the quantitative measure for improvement of changeability was not given and theoretical guidelines are not clear about the cause effect relation between given patterns.

The authors Malhotra et al. (2013) proposed a change proneness prediction model which predicted classes that showed change proneness by means of object oriented design metrics. The model proposed was fully based on open source software data sets [14]. They analyzed and reused the produced estimate model of a chosen project and mapped it on a separate project thereby reducing to some extent dependency of training data on development of prediction model. Measurement of change prone classes of software was done using non liner data fitting bi square method with robust results by Ankita Urvashi et al. [15]. They would have extended their work by using probability density function to give better insight into the nature of mathematical relationship between the change-proneness and the factors/random variables that influence it. Moreover, this model was not empirically validated and not applicable in the context. These outcomes though have not had a wide acceptance and thus, have not been used in practical by the

practitioners. In addition, the model provided by the authors is not sufficient for both structural and behavioral architecture.

In the work done by Panjeta et al. (2014) authors highlighted changeability as one of the key characteristics of software maintainability [16]. They theoretically and graphically tried solving this important subject by proposing a structure that facilitates to evaluate level of changeability by means of clustering methods (Machine Learning). To quantify changeability, authors proposed theoretical and graphical approach; but quantifying changeability through this technique showed high complexity. In this study, the authors have not given the quantitative measure of software changeability. However, they have only discussed theoretical approach for measuring software changeability.

Work done by Sen-Tarng Lai (2014) proposed a model for improving process of plan change in software project and mitigating development risk. A model called (WBSPM) i.e. WBS–based Plan changeability was proposed to increase change capability plan in WBS-based method taking into consideration changeability factors.This approach did not propose a generic model for changeability at design phase. This study is largely concerned with recognizing and assessing the factors of changeability in object oriented software and metrics correlated to the factors, which are been backed by the case studies. The authors used the source code analysis for characterizing the software changeability. In this research, authors identify possible relevant metrics to predict the class changeability and analyzed the approach in theoretical manner only. Moreover, this approach has more emphasis on analysis phase; design phase has been considered only partially [17].

In the study done by Rongviriyapanish et al et al. (2016) java code changeability prediction model was proposed. Authors highlighted a value model for evaluating the levels of changeability in java program as significant in software development. The paper proposed a software changeability estimation model that took into account the metrics involving several appropriate object oriented attributes. The proposed method presented by using the multi layer perception, as a classifier arrangement and for training data of java classes from jEdit open source software project. An approximation of 74.07% was attained and the model could completely divide java classes with decent changeability level ranging from reduced or acceptable changeability levels [18]. The proposed java code changeability prediction model measured changeability at source code level only. Study argued this model improved maintenance, debugging and hence improves software quality.

After a systematic literature review it comes into observation that there are numerous approaches available for measuring object oriented software changeability at analysis and coding phase. However at analysis phase, we only have the requirements and at design phase only, the complete structure of the software comes into picture. Therefore, changeability assessment at design phase is much more relevant as compared to analysis phase and also cannot be compensated during subsequent development life cycle. Panjeta et al. (2014) proposed theoretical and graphical approach for changeability assessment at design phase but quantifying changeability through this technique is very complex. Hence, there is a potential to develop a systematic solution for changeability evaluation at design phase in SDLC. Therefore, a comprehensive outline and related model to evaluate changeability of object oriented software with the help of object oriented design properties at design phase seems highly needed and significant.
At analysis phase, we have the functionality requirement and only at design phase, the complete structure of the software comes into picture. The lack of software changeability at design phase may be difficult to overcome throughout ensuing system development life cycle. Hence, there is a potential to develop a systematic solution for changeability evaluation which is implemented at design phase of different stages of SDLC. Therefore, a comprehensive outline and related model

to evaluate changeability of object oriented software with the help of object- oriented design properties at design phase seems highly needed and significant.

3. OBJECT ORIENTED DESIGN

A lot of study by experts has given insights to how object oriented design properties correlate and influence the attributes of quality in design. They propose a robust relationship between these attributes of quality Changeability and properties of design. The work of these researchers on establishing relationship between object oriented design properties and changeability can be summarized as:

M. Ajmal Chaumun, quality expert, in [9] established a relationship among design property coupling and changeability. They observed that coupling either has a direct or inverse impact on changeability. The characteristics of changeable software like coupling make it easier for reviewers to understand the software artifacts under review.

In 2009, M. K. Abdi et al. [10] identified the design properties inheritance, coupling, polymorphism that influence software changeability in object oriented design and development. They also formulated a set of rules in object oriented design and development to increase the software quality by increasing their changeability.

Panjeta et al. in [16] identified design properties that influence changeability. They described encapsulation, inheritance and coupling as having direct impact on software changeability. They strongly proposed that these are the characteristics of program leading to changeable software.
Sen-Tarng in [17] also recommended that encapsulation, inheritance and polymorphism affect changeability from the beginning, especially at design phase though there main emphasis was on analysis phase.

Kabaili in [8] has defined changeability as a product of coupling, inheritance and polymorphism.
In view of this fact, a relation figure is proposed between the major properties of object oriented design Changeability as shown in Fig. 1. The mapping puts in place a contextual impact relationship among Changeability and object oriented design properties and the related design metrics.

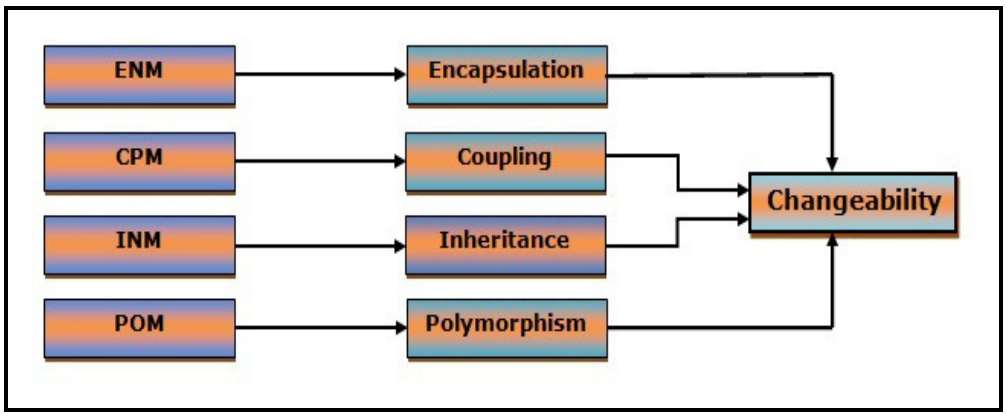

Fig. 1: Mapping between Design Properties and Changeability

4. DEVELOPMENT OF MODEL

Here we have implemented the concept of multiple linear regression (MLR) to develop a model for Changeability. This method is represented by Equation (1) below.

$$Z=\beta_0+\beta_1 Y_1+\beta_2 Y_2+\beta_3 Y_3+\text{-------}+\beta_n Y_n \quad \text{….. } (1)$$

Here,

- **Z** (Dependent Variable).
- $\mathbf{Y_1, Y_2, Y_3}$**--------**$\mathbf{Y_n}$ (are independent variables) related to Z and are an indication to the variance in Z.
- $\beta_1, \beta_2, \beta_3$--------$\beta_{n.,}$ these are coefficient of the each independent variables.
- The intercept is given by β_0

The datasets for developing and validating Changeability model is acquired from [19] that have been collected through the class diagrams. It includes a set of twenty (20) class diagrams (indicated from Project1 up to Project20) along with the value of metrics of each of these. Along with this, we have the actual mean values of different ratings by experts of Software Changeability for these projects. These are called 'Known Value' here in this research paper.
Equation (2) represents the relation between Maintainability Factor i.e changeability and the Object Oriented design properties as calculated by us. The coefficient values are calculated in SPSS and a Changeability model is framed. The data is taken from 10 projects viz. Pr8, Pr9, Pr10, Pr11, Pr12, Pr13, Pr14, Pr15, Pr16 and Pr17.

Changeability = 8.477 - 0.367× Encapsulation -1.53 × Coupling -1.945 × Inheritance + 1.923 ×

Polymorphism Eq. (2)

Table 1 shows the coefficients for Changeability evaluation model. We use the values we get from the unstandardized coefficients component of the table 1 to help develop the regression equation (2). The results of this trial experiment in assessment of changeability meet expectations and are very promising to attain maintainability index of object oriented design for small cost Software maintenance.

Table 1: Coefficients for Changeability Evaluation Model

Changeability Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Standard Error	Beta	t	Sig.
1	Constants	8.477	2.906		2.917	.033
	Encapsulation	-.367	.452	-.140	-.813	.453
	Coupling	-1.530	.481	-.522	-3.182	.024
	Inheritance	-1.945	1.288	-.250	-1.510	.191
	Polymorphism	1.923	.512	.643	3.759	.013

The results of summarized model table 2 are useful when calculating multiple regression. The coefficient determinant (R) exhibits the strong relation between the independent variables and the

respective dependent variable. The value of this coefficient when squared i.e R(square) from the table depicts the coefficient of determination.

Table 2: Changeability Evaluation Model Summary

Model(Summarized)				
Model	R	R Square	R Square (Adjusted)	Standard Error of the Estimate
1	.934[a]	.872	.770	.60873
a. Predictors: (Constant), Polymorphism, Coupling, Inheritance, Encapsulation				

5. VALIDATION OF ABOVE MODEL

This part of study focuses on the way the model proposed above is able to evaluate the Changeability calculated in object oriented software(s) at SDLC design stage. This experimental validation exists as a crucial step of proposed research to estimate Changeability evaluation model for better and high level adaptability. Therefore with this objective a validation of the Changeability evaluation model and it is done using experimental tests.

In order to validate the developed Changeability evaluation Model the projects viz. Pr1, Pr2, Pr3, Pr4, Pr5, Pr6, Pr7, Pr18, Pr19 and Pr20 were taken. The known Changeability rank of the provided projects class diagram is shown in Table 3.

Table 3: Known Changeability Value

Pr1	**Pr2**	**Pr3**	**Pr4**	**Pr5**	**Pr6**	**Pr7**	**Pr18**	**Pr19**	**Pr20**
7.8	6.9	8.1	7.4	8.5	7.2	7	9.1	8.9	9.3

Table 4: Known Changeability Rank

Pr1	**Pr2**	**Pr3**	**Pr4**	**Pr5**	**Pr6**	**Pr7**	**Pr18**	**Pr19**	**Pr20**
5	1	6	4	7	3	2	9	8	10

Using the similar set of data for the given projects class diagram Changeability was calculated using proposed Changeability evaluation model and the results are shown in Table 5.

Table 5: Calculated Changeability Value Using Proposed Model CEM^{OOD}

Pr1	**Pr2**	**Pr3**	**Pr4**	**Pr5**	**Pr6**	**Pr7**	**Pr18**	**Pr19**	**Pr20**
2.5	2.2	3.8	3.4	3.7	1.4	0.4	4.4	6.7	5.8

Table 6: Calculated Changeability Rank Using Proposed Model CEM^{OOD}

Pr1	**Pr2**	**Pr3**	**Pr4**	**Pr5**	**Pr6**	**Pr7**	**Pr18**	**Pr19**	**Pr20**

4	3	7	5	6	2	1	8	10	9

Charles Speraman's rank relation r_s was used to test the significance of correlations calculated amidst Ranks of Changeability via proposed model and the ranks Known in it. The 'r_s' was calculated using the formula given as under:

$$r_s = 1 - \frac{6\Sigma d^2}{n(n^2-1)} \quad -1.0 \leq rs \leq +1.0 \qquad \text{Eq. (3)}$$

'd' = difference that exists in Calculated Rank and Known Rank of Changeability.
'n' = total quantity of Projects taken in the experimentation.

Table 7: Computed Rank, Actual Rank and their Relation

Project(s) Changeability	Pr1	Pr2	Pr3	Pr4	Pr5	Pr6	Pr7	Pr18	Pr19	Pr20
Computed Ranks	4	3	7	5	6	2	1	8	10	9
Known Ranks	5	1	6	4	7	3	2	9	8	10
d^2	1	4	1	1	1	1	1	1	4	1
$\sum d^2$	16									
$r_{s\ Calculated}$	0.90303									
$r_s > \pm.781$	✓									

The correlation value among calculated Changeability ranks using proposed model CEM^{OOD} and known ranks are shown in Table 7 above. Correlation value r_s undoubtedly show that the Changeability model is significant. The correlation meets the expectations standard showing high confidence, i.e. of 95%.

6. CONCLUSIONS

This paper shows the importance of Changeability and its correlation with object oriented design properties viz. encapsulation, coupling, inheritance and polymorphism. Using multiple linear regression formula on these attributes of object oriented design CEM^{OOD}, a changeability model is developed. The results obtained statically confirm the significance and acceptability of the proposed model. Changeability evaluation model has been validated empirically via experimental test. The real-world validation of the Changeability model accomplishes that developed model is highly dependable, acceptable and significant.

REFERENCES

[1] Malhotra & Anuradha Chug ,.(2012) "Software Maintainability Prediction using Machine Learning Algorithms.", *Software Engineering: An International Journal (SEIJ), Vol. 2, No. 2*

[2] ISO/IEC 9126-4:2004, "Software Engg. Product Quality-Quality in Use Metrics", *ISO/IEC 2004*

[3] McCall, J.A., Richards, P.K., and Walters, G.F., (1977) "Factors in Software Quality", *RADC TR-77-369, Vols I, II, III, US Rome Air Development Centre Reports.*

[4] Adam M. Ross1 , Donna H. Rhodes2 , and Daniel E. Hastings3,(2008) "Defining changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value", *Systems Engineering © 2008 Wiley Periodicals, Inc*

[5] Sneed, H., Mercy, A. (1985), *Automated Software Quality Assurance. IEEE Trans. Software Eng., 11Bi, 9: 909-916.*

[7] Laxmi Shanker Maurya & Gauri Shankar,*(2012),"* Maintainability assessment of web based application.'', *Journal of Global Research in Computer Science,* Vol 3, No. 7

[8] Hind Kabaili, Rudolf K. Keller and François Lustman,(2001) "Cohesion as Changeability Indicator in Object-Oriented Systems", *Software Maintenance and Reengineering, 2001- Fifth European Conference on Software Maintenance and Reengineering, IEEE Xplore: 07* August 2002

[9] M. Ajmal Chaumun, Hind Kabaili, Rudolf K. Keller and François Lustman, (2002)"A Change Impact Model for Changeability Assessment in Object-Oriented Software Systems", *Science of Computer Programming,, Elsevier, Volume 45, Issues 2–3, November–December 2002, Pages 155-174*

[10] M.K Abdi, H. Lounis, H. Sahraoui,(2006) "Analyzing Change Impact in Object-Oriented Systems " *In proceedings of the 32nd EUROMICRO Software Engineering and Advanced Applications Conference, Cavtat/Dubrovnik (Croatia), August 29- September 1, 2006.*

[11] M.K. Abdi, H. Lounis, H. Sahraoui,(2009)" A Probabilistic Approach for Change Impact Prediction in Object-Oriented Systems", *AIAI-2009 Workshops Proceedings.*

[12] Yirsaw Ayalew & Kagiso Mguni, (2013) "An Assessment of Changeability of Open Source Software", *Computer and Information Science; Vol. 6, No. 3*

[13] Xiaobing Sun, Bixin Li & Qiandong Zhang,(2012) "Change Proposal Driven Approach for Changeability Assessment Using FCA-Based Impact Analysis", *Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual*

[14] Dr. Ruchika Malhotra, Megha Khanna,(2013)" Inter Project Validation for Change Proneness Prediction using Object Oriented Metrics", *An International Journal (SEIJ), Vol. 3, No. 1*

[15] Ankita Urvashi , Anamika Chhabra,(2014)" Predicting Changeability in Software Components Using Bisquare Method for Optimization", *International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 9*

[16] AnuradhaPanjeta,Prof.AjayKumar,(2014) " Performance Analysis to Changeability of Measuring Software Components", *International Journal Of Engineering Research & Management Technology Volume 1, Issue-4.*

[17] Sen-Tarng Lai Shih, (2014) "A WBS-Based Plan Changeability Measurement Model for Reducing Software Project Change Risk ",*Lecture Notes on Software Engineering,Vol. 2, No. 1,*

[18] Songsakdi Rongviriyapanish, Thanapol Wisuttikul & Boonchai Charoendouysil,(2016) "Changeability prediction model for java class based on multiple layer perceptron neural network", *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2016 13th International Conference*, IEEE Xplore: September 2016

[19] MoboDexter Software India Pvt. Ltd., Novel Tech Park, Third Floor, #43/4, GB playa, Hosur Road Bangalore

Authors

Nidhi Goyal is a Ph.D. Scholar at BBDU and works as Asst. Professor at Amity University, Lucknow.

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, and Software Engineering.