



Requirement Based Testability Estimation Model of Object Oriented Software

MOHAMMAD ZUNNUN KHAN^{1*}, M. A. KHANAM² and M. H. KHAN³

^{1,2}Department of Computer Science and Engineering, Integral University, Lucknow 226026, India.

³Department of Computer Science and Engineering, IET Campus, Lucknow 226026, India.

Abstract

To measure testability before the actual development starts will play a crucial role to the developer, designers and end users as well. Early measurement of testability, especially in early requirement stage to assist the developer for the further development process, and will also assures us to produce and deliver the high quality requirement that can surely reduce the overall cost and improves the quality of development process. Taking view of this fact, this paper identifies testability estimation factors namely understandability and modifiability and establishes the correlation among testability, understandability and modifiability. Further, a model is developed to quantify software testability in requirement phase and named as Requirement Testability Model of Object Oriented Software-RTMOOS. Furthermore, the correlation of Testability with these factors has been tested and justified with the help of statistical measures.



Article History

Received: 26 July 2017
Accepted: 16 October 2017

Keywords

Requirement Testability,
Understandability,
Modifiability,
Object Oriented
Software,
UML


Introduction

One of the major issues that are considered in the current era of software development is to provide quality software; the reason behind it can be defined as human life is depending upon the correct functioning of computer and to get competitive advancement. Object oriented technique is widely accepted and preferred way of development in large-scale software development^{8,32}. This new paradigm has made new difficulties to testing, which needs to

manage new issues presented by the most useful object oriented features, for example, inheritance, polymorphism, coupling, cohesion, encapsulation, and dynamic binding. Particularly managing instantiations of classes and their coordinated effort might be extremely troublesome when testing is performed⁶. Testability proposes the testing force, and gives the level of trouble which will be supported amid testing of a specific position to recognize a fault⁷.

CONTACT Mohammad Zunnun Khan  zunnun Khan@gmail.com  Department of Computer Science and Engineering, Integral University, Lucknow 226026, India.

© 2017 The Author(s). Published by Techno Research Publishers

This is an  Open Access article licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (<https://creativecommons.org/licenses/by-nc-sa/4.0/>), which permits unrestricted NonCommercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

To link to this article: <http://dx.doi.org/10.13005/ojcs/10.04.14>

Software testability information is a useful technique corresponding to testing. Higher test scope might be completed by making requirement more testable for a similar amount of efforts, which thus builds the certainty to the system. It is obvious from literature survey that there is no known existing and complete model or system available for assessing the testability of object oriented paradigm mulling over requirement stage^{8,12}. The model proposed in this paper tends to many issues raised by different specialists and experts of the domain. This model has low-level object oriented metrics all around characterized regarding plan qualities. The arrangements of exactly distinguished and weighted object oriented properties are utilized to evaluate the testability. Rest of the section in this paper are as follows: Part 2 and Part 3 describe software testability and testability at Requirement phase. Part 4 depicts testability factors and key contributors.

In Part 5, a Requirement Based Testability Estimation Model for Object Oriented Requirement: Quality Perspective has been proposed, requirement properties have been characterized and a concise depiction of distinguished metrics has been incorporated. The utility of this model in foreseeing requirement testability has been approved against a few certifiable undertakings in Part 6. It has been inferred that testability anticipated by this model shows high connection with evaluators assessments in Part 7 and a hypothesis is tested based on 2-sample t-test is being performed and confidence interval is being seen by the distinction of two standards mean in Part 8. At the end study finished up with some key discoveries in Part 9 and Part 10.

Testability

Study of Software testability has been a fundamental research heading since 1990s and turns out to be more inescapable when entering 21st century^{13,14}. As indicated by IEEE standard, the software testability alludes to how much a framework or part encourages the execution of tests and the foundation of test criteria to decide if those test criteria have been met¹⁵. In ISO 9126 quality model, testability holds an unmistakable place as a component of the modifiability for software quality¹⁶. With a specific end goal to limit the testing try outs, an endeavour can be made to foresee which class is more testable,

by taking a gander at two classes¹⁷. Testability, containing certain attributes of a software framework that makes it less demanding or harder to test and to examine the test outcomes, is an essential factor to accomplish a productive and successful test handle. Developing, confirming and measuring exceedingly testable software turns into an imperative and testing undertaking for software originators^{9,10}. A significant part of the examination work uncovers that greatest endeavours have been committed with the source code. The assurance of testability for an effectively composed code might be too exorbitant in light of the fact that in last the progressions are presented the more costly they are¹⁸. A broad review of writing uncovers that procedures, rules, and devices identified with software testability are missing¹⁹. Subsequently, there had all the earmarks of being requirement for assessing the testability of software in early phase of development life cycle without the accessibility of code. Accessibility of a reasonable and sufficient measuring model at the early phase of improvement empowers early expectation of framework testability, subsequently, upgrades the quality of overall development.

Software Testability at Early Stage

A requirement is a procedure that begins from an investigation of a domain issue prompting some formal documentation. Software requirement, from multiple points of view, is an unusual craftsmanship²⁰. At the principal occurrence, it might bring about a model of the domain issue by formally catching and representing to the client's necessities and henceforth, making ready for a reasonable connection. Amid the requirement phases of software, it is spoken to as far as requirement specification, building and point by point requirement diagram^{10,11}. these portrayals catch the structure and conduct of the software before it is executed. The portrayals are then changed into the genuine software implementation.

The challenge is to examine how these portrayals affect the last execution of the software, with the point of distinguishing qualities as well as examples in the portrayals that may upgrade or maybe testability. Recognizing such attributes or potentially examples would empower one to make portrayals of software that develop into better testable usage and in this way, enhance the time and exertion proficiency amid software testing¹² Enhancing software testability

is plainly a key target to lessen the quantity of deformities that outcome from ineffectively outlined software^{23,21}. Testable requirement is more particular then great plan since it is unequivocally proposed to coordinate a specific test context^{28,25}. One proactive system that associations can embrace is to plan their software product with testability as one of the key criteria. Parts of testability like requirement understandability and modifiability conduct are the essential concentration of good requirement and require uncommon treatment²⁴. Undoubtedly, it is a key to the successful development of quality software. It is additionally the progression that will decide the general structure, nature, and approach of the resulting software.

Testability Factors and Major Contributors

A lot of work has been completed in depicting the need and significance of consolidating software testability since mid 90s. Various strategies for measuring testability have been proposed. Sadly, huge accomplishments made by the specialists in the territory have not been broadly acknowledged and are not received practically speaking by industry²⁶. It has been discovered that there is a contention in considering the variables while assessing software testability as a rule and at configuration level solely. It has been gathered from the writing overview on testability investigation that there is an overwhelming need of distinguishing a regularly acknowledged arrangement of the components influencing software testability^{11,12,15,21}. A critical exertion has been set up to gather an arrangement of software testability factors to be specific, modifiability, understandability, simplicity, reusability, self descriptiveness, complexity, traceability and modularity² that can influence software testability at requirement time being developed life cycle^{3,5}. Out of these testability factors, some of them have their immediate effect in assessing testability of question arranged software, while different elements have less or immaterial effect. An attempt has been made to distinguish the testability factors that precisely influence software testability estimation at configuration stage. "Understandability"³ and "Modifiability"⁴ are recognized the key testability factors that precisely influence software testability Estimation and satisfy the quality criteria, especially understandability quality criteria is traceability²⁷, self descriptiveness and modifiability quality criteria is

complexity, simplicity¹. Along these lines, it comes into see sensible to incorporate understandability and modifiability for testability Estimation at requirement stage.

Requirement Based Software Testability Model Development

To Develop a Requirement Based Testability Model of Object Oriented Software-RTM^{OOs}, Quality Model proposed by Dromey^{31,22} has been considered. Figure 1, demonstrates the relationship foundation among Testability, Testability factors, Object Oriented Metrics, and depicts the estimation procedure of Requirement testability display. This includes ensuing strides:

- Recognizable proof of Testability Factors that impacts testability of programming.
- Recognizable proof of Object Oriented Metrics.
- A methods for connecting them.

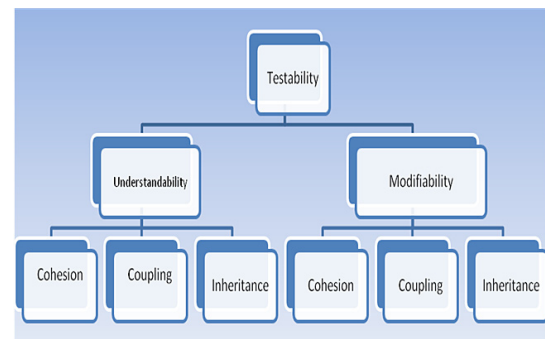


Fig. 1: Correlation among Testability, Testability Factors and Object Oriented Metrics

The relative essentialness of individual recognized testability factors that have significant effect on testability estimation at requirement stage is weighted relatively. The estimations of these protest arranged metrics can be recognized by requirement graph metrics. Keeping in mind the end goal to set up a Requirement Based Model for Testability estimation, a various regression procedure has been utilized to get the coefficients of regression factors and regression intercept, appeared in Table 1. Distinguished testability components will participate in the part of autonomous factors while testability will be taken as needy variable. Estimation of testability is exceptionally useful to get testability list of

programming outline for top notch item. Multivariate regression condition is given in Equation (1) which is as per the following:

$$M = \alpha^0 \pm \alpha^1 N^1 \pm \alpha^2 N^2 \pm \alpha^3 N^3 \pm \dots \pm \alpha^n N^n \quad \dots(1)$$

Where

M is a variable (dependent),
 N¹, N², N³, and Nⁿ are variables (independent),
 α¹, α², α³ and αⁿ works as regression coefficient of independent variables respectively,
 α⁰ is the intercept of the equation.

It has been extensively checked on and examined in Section 4; Understandability and Modifiability are the main considerations influencing programming testability estimation at requirement stage²⁹. Hence, these recognized significant testability factors were tended to well ahead of time while consolidating testability at requirement organizes. By applying the regression technique, think about officially created Understandability Model^{3,32} and Modifiability Model⁵ that is given in Equations (2) and (3) individually and finally (4). The model of Understandability and Modifiability frames the solid reason for advancement of Requirement based Testability Estimation Model.

$$\text{Understandability} = 1.50 + 0.0070 * CA + 0.35 * MFA + 0.121 * CAM \quad \dots(2)$$

$$\text{Modifiability} = 3.27 - 3.73 * CAM + 0.219 * CA + 3.00 * MFA \quad \dots(3)$$

It was watched that each Object Oriented metrics influence quality factor. Object oriented metrics to be specific Inheritance (MFA: Measure of Functional Abstraction), Cohesion (CAM: Cohesion among Methods of Class) and Coupling (CA: Afferent Coupling CA)²⁶ are utilized to address the key testability factors specifically Understandability and Modifiability. These two recognized components are additionally used to measure testability index of object oriented software at requirement phase being developed life cycle. Figure 1, gives a diagram of the requirement. With a specific end goal to set up a model for software testability estimation, a numerous relapse strategy examined in Equation (1) has been connected. Therefore considering,

the effect of object oriented metrics specifically Inheritance, Coupling and Cohesion on testability supporters "Understandability and Modifiability", following various relapse display has been defined that can be utilized to create testability model for object oriented software.

$$\text{Testability} = \alpha^0 \pm \beta^1 * \text{Understandability} \pm \beta^2 * \text{Modifiability} \quad \dots(4)$$

For creating requirement based software testability model, the information has been taken from²³, which comprise of six business software ventures with around 22 numbers of projects. The estimations of object oriented metrics to be specific, Inheritance Metrics (MFA), Coupling Metrics (CA) and Cohesion Metrics (CAM) and the estimations of "Understandability and Modifiability " have been utilized. Utilizing SPSS, math work software relationship coefficients are calculated and model of testability Estimation is in this manner detailed as given in Equation (5).

$$\text{Testability} = 290 - 153 * \text{Understandability} + 0.95 * \text{Modifiability} \quad \dots(5)$$

Table 1: Testability Computation Table

Project	Understand-ability	Modifi-ability	Known Testability Index
P ₁	1.58228	2.89271	5.8787
P ₂	1.53945	1.75900	7.7159
P ₃	1.50540	2.07302	9.1878
P ₄	1.50540	00.51550	9.5653
P ₅	1.53651	2.07302	9.1742

Testability Equation Computation Table

Table 2: Statistical Significance of Testability Model- RTM^{oos}

	Descriptive Statistics		
	Mean	Std. Deviation	N
Cal. Testability	55.851311	3.0291272	22
Understandability	1.806394	.0915236	22
Modifiability	4.441628	1.1061617	22

Analysis of Testability Quantification Model Statistical Significance of Model

The descriptive table (Table II) is very important for further research work. It gives the valuable record

of descriptive statistics that are mean, standard deviation and number of samples selected for model validation.

Table 3: Correlation between Independent Variables

Correlations		Cal. Testability	Understandability	Modifiability
Pearson Correlation	Cal. Testability	1.000	.253	.464
	Understandability	.253	1.000	.974
	Modifiability	.464	.974	1.000

Pearson’s coefficient of correlation technique was used for estimating the degree of correlation among variables. The value of ‘r’, i.e. correlation lies between

+1 to -1, positive value of ‘r’ in Table III is a sign of represents the positive correlation between the two variables.

Table 4: Model Summary for Testability Model

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.990a	.980	.978	.4454136

a. Predictors: (Constant), M, U

Summary Table IV for Testability Quantification Model proves that all the four selected metrics are statistically significant at confidence level of 95%.

justified for acceptance of model; an experimental way of validation of the proposed Requirement Testability quantification model has been carried out using various samples.

Empirical Validation Of Requirement Based Testability Model

This section is used to prove that proposed study is how much significant, where metrics and model are used to estimate the Requirement Testability index of object oriented software specially at requirement time. The empirical validation plays an important role in the research to evaluate the proposed Requirement based Testability quality model for high level acceptability and appropriate execution. Empirical validation is one of the finest approach and best practice for claiming the model acceptable or not^{19, 30}. The claimed approach is

Data Set for 22 Projects

An analysis was conducted on collected warehouse with more the 90 versions of around 37 proprietary, academic and open-source projects were used^{32,31}. Keeping in mind that an experimental validation of the proposed model for Requirement based Testability evaluation has been carried out using illustration tryouts. In order to validate proposed requirement based Testability quantification model, the value of metrics are available by using³² data set for following 22 projects in table V.

Table 5: Standard and Calculated Testability Index Values for 22 Projects

Project	Understandability	Modifiability	Std. Testability	Cal. Testability
P1	1.7855	3.9164	54.3405	52.4896
P2	1.7822	4.5761	54.8879	60.6428
P3	1.8659	5.3738	55.1397	59.1720
P4	1.8299	4.1910,u	54.3544	49.9753
P5	1.6157	1.8783	53.4912	50.5555
P6	1.8406	4.5608	54.6025	52.9231
P7	1.8659	5.3738	55.1397	59.1720
P8	1.8131	4.5920	54.7560	56.8314
P9	1.8365	4.8924	54.8890	55.3325
P10	1.9485	6.1617	55.3889	57.7356
P11	1.8241	4.9968	55.0312	60.1577
P12	1.7814	4.1728	54.5663	56.0194
P13	1.8405	4.9966	54.9544	58.0385
P14	1.8487	4.6695	54.6525	53.1554
P15	1.6297	2.3163	53.7788	53.8741
P16	1.8161	4.6173	54.7626	56.7467
P17	1.8195	4.8967	54.9717	59.5718
P18	1.8743	5.2416	54.9938	56.5401
P19	1.8321	4.6271	54.6956	54.7922
P20	1.5615	1.8105	53.6897	56.7558
P21	1.8884	5.2930	54.9694	55.3240
P22	1.8406	4.5608	54.6025	52.9231

It is compulsory to test the validity of proposed model for acceptance. A 2 sample t test applies for check the significance between standard Testability and

calculated Testability. 2t-test is handy hypothesis tests in statistics when compare means.

Table 6: 2 t- tests between Standard Testability and Calculated Testability

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	Std. Testability	54.666286	22	.4850048	.1034034
	Cal. Testability	55.851311	22	3.0291272	.6458121

Null hypothesis (H₀)

There is no significant difference between Standard Testability and Calculated Testability; H₀: μ₁-μ₂ = 0

Alternate hypothesis (H_A)

There is significant difference between Standard Testability and Calculated Testability; H_A: μ₁-μ₂ ≠ 0

In the above hypothesis μ₁ and μ₂ are treated as sample means of population. Mean value and Standard Deviation value have been calculated for specified two samples and represented in table 6. Correlation comes out to be 0.933, that shows the standard Testability and calculated Testability is highly correlated. The hypothesis is tested with zero level of significance and 95% confidence level. The

p value is 0.056. Therefore alternate hypothesis directly discards and the null hypothesis is accepted. The developed equation used for requirement based Testability estimation is accepted.

Critical Findings

This Study developed 'Requirement Based Testability Estimation Model for Object Oriented software.

The Model is also validated using the same set of try-out data. An empirical validation is also preformed on the developed model using different try-out data sets. Few of the major findings are mentioned below:

- Software testability has been recognized as major quality factor for software development, addressed in requirement phase of object oriented software to produce quality software.
- Low level method of each of the testability factors may be obtained.
- Software requirement constructs are most suitable and influential for controlling software quality factors especially in requirement phase.
- There is a possibility of establishing correlation among testability and other quality factors in near future, in order to address them in requirement phase.
- Understandability and Modifiability are

identified as two most important factors affecting software testability in requirement phase.

- Testability indexing (TI) is possible using the model "Requirement Based Testability Estimation Model for Object Oriented Software" for Industry project ranking.
- The models possibly will be generalized and used by other researcher for making testability levelling of projects undertaken.

Conclusion

Requirement based Software testability key factors namely Understandability and Modifiability are identified and their significance on testability Estimation at requirement phase has been experienced and justified. Requirement based Testability Estimation model for object oriented software has been developed and the statistical inferences are validated for high level model acceptability. The developed model to evaluate testability of object oriented software is extremely consistent and correlated with object oriented software artifacts. Requirement based Testability Estimation model has been validated hypothetically as well as empirically using experimental test. That validation study on this research work proves that developed testability estimation model is highly acceptable, more practical in nature and helps the software industry in project ranking.

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 Mohammad Zunnun Khan, M A Khanam, M. H. Khan. "Requirement Understandability Quantification Model of Object Oriented Software." *International Journal of Advanced Research in Computer Science* 8.7 (2017).
- 4 Khan, M. Z., Khanam, M. A., & Khan, M. H. (2017). Requirement Modifiability Quantification Model of Object Oriented Software. *Global Journal of Pure and Applied Mathematics*, 13(3), 2017.
- 5 Y. Wang, "Design for Test and Software Testability", University of Calgary, 2003. <http://www.ucalgary.ca/~ageras/wshop/abstracts/2003/design-foretestability.htm>
- 6 Jungmayr, Stefan. "Testability during Design, Software Technik- Trends." *Proceedings of the GI Working Group Test, Analysis and Verification of Software*, Potsdam (2002): 10-11.
- 7 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.

- 8 D. Esposito, "Design Your Classes for Testability", 2008. <http://dotnetslackers.com/articles/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 Davis, A.; Overmyer, S.; Jordan, K.; Caruso, J.; Dandashi, F.; Dinh, A.; Kincaid, G.; Ledebor, G.; Reynolds, P.; Sitaram, P.; Ta, A.; Theofanos, M., "Identifying and measuring quality in a software requirements specification," Software Metrics Symposium, 1993. Proceedings, First International, vol., no., pp.141,152, 21-22 May 1993
- 11 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.
- 12 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.
- 13 Shahid Iqbal and Naeem Ahmed Khan M. Yet another Set of Requirement Metrics for Software Projects. *International Journal of Software Engineering and Its Applications*. 2012; 6.1:19-28.
- 14 Ali Mohammed Javeed. Metrics for Requirements Engineering. 2006. Available from: www.cs.umu.se/education/examination/Rapporter/JaveedAli.pdf
- 15 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 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 Gallin, Software Quality Assurance from Theory to Implementation. Edinburgh Gate: Pearson Education, 2004.
- 21 J. Ramos, Ricardo; Piveta, Eduardo K; Castro, Jaelson; Moreira, Ana; Guerreiro, Pedro; Pimenta, Marcelo S; Price, R. Tom; Araujo, "Improving the Quality of Requirements with Refactoring," in Simposio Brasileiro de Qualidade de Software, 2009.
- 22 Firesmith, "Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them," *J. OBJECT Technol.*, Vol 6, No. 1, pp. 17-33, 2007.
- 23 Bokhari Mohammad Ubaidullah and Shams Tabrez Ubaidullah Siddiqui. Metrics for Requirements Engineering and Automated Requirements Tools. Proceedings of the 5th National Conference, INDIACom-2011.
- 24 Jungmayr, Stefan. "Testability measurement and software dependencies." Proceedings of the 12th International Workshop on Software Measurement. 2002.
- 25 Khan, M. Z., Khanam, M. A., & Khan, M. H. (2016). Software Testability in Requirement Phase: A Review. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(4), 1031-1035. DOI 10.17148/IJARCCCE.2016.54252
- 26 Chidamber, S. R. and Kemerer, C. F., "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, vol.20, 1994.
- 27 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.

- 28 ISO. International standard ISO/IEC 9126. information technology: Software product evaluation: quality characteristics and guidelines for their use, 1991.
- 29 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.
- 30 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.
- 31 Dromey, R.G.: A Model for Software Product Quality. IEEE Transaction on Software Engineering 21(2), 146–162 (1995)
- 32 Anshul Mishra, Devendra Agarwal, M.H. Khan, "Integrity Estimation Model: Fault Perspective", May 17 Volume 5 Issue 5 , International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC)