Predicting Reliability of Software Using Thresholds of CK Metrics

Johny Antony P
Asst. Professor, Dept. of MCA, SHIMT, Sitapur, U.P-261001, India

Abstract: Predicting Reliability is one of the key function of a software system. Many of the software fail due to unreliability. The demand of reliable software is increasing day-by-day. In industry, information on reliability is available too late in the software development process. Hence any corrective action becomes unaffordable. A step towards the remedy to this problem is the ability to provide a threshold for the reliability of a software product. Object oriented metrics are most beneficial and reliable for the estimation of the threshold for reliability. In this paper, we use the Chidamber and Kemerer Metrics to assess the threshold values for reliability. A tool is designed and developed called 'Java Class Analyzer' which extracts the values of the metric parameters from the source code. These values are evaluated against the threshold values of the metrics from the literature. It provides a threshold for the software reliability. The result provides a standard against which the software reliability can be evaluated and necessary corrective actions can be implemented.

Key Words: CK Metrics, Object Oriented Metrics, Reliability, Software, Threshold

1. Introduction

Today quality is critical for survival and success. Several years back, software was considered to be a technical business where functionality was the key factor of success. Today functionality alone is not sufficient, but various quality factors are evaluated against the product. So quality is a major issue in the software development. As Nina S Goodbole points out, "Quality goals must be clearly defined, effectively monitored and rigorously enforced" [16]. Quality must be defined and measured if excellence is to be achieved, business is to be successful. These thoughts bring to our mind that how we assess the quality of something intangible like software quality. To assess quality, the quality attributes must be taken into consideration and measured in the planning and design of the software [26].

The measure of reliability ensures whether software under development has been implemented correctly. Unlike other engineering disciplines, absolute measurements like mass, velocity are uncommon in software engineering. Metrics are used to evaluate the process and the product in its various stages against standard and norm. Metrics can provide the information we need to control resources and processes used to produce the software. Metrics are the continuous application of measurement based techniques to the software development process and its products to supply meaningful information together with the use of techniques to improve the process and its products [7].

Metrics are indicators used to denote a representation of metric data that provides insight into an ongoing software system development project. They provide measurement for software development – source and object code, requirement documents, programs and tests. Introducing metrics and making use of it, we can control and improve the reliability of software

Threshold is a point beyond which there is a change in the manner a program executes; in particular, an error rate above which the operating system shuts down the computer system on the assumption that a hardware failure has occurred. There are threshold values defined by the researches and vendors for the metrics. Based on these thresholds values, threshold for software reliability can be estimated. This will help the designers and producers to check the product against the threshold of reliability, if it doesn't fall within the range, then the decision for redesign has to be made in order to meet the specifications.

2. Literature Review

2.1 Different Views of Software Reliability

Over 200 models have been developed since the early 1970, but now to quantify software reliability still remains largely unsolved. Challenges and open questions still exist. Number of guidelines are available in the literature that suggests various do's and don'ts to produce a reliable system [1,2,6,36]. There is always increasing demand for reliable software. Software reliability has emerged as people try to understand the characteristics of how and why software fails [40].

The IEEE defines software reliability as "The ability of a system or component to perform its required functions under stated conditions for a specified period of time [25]. The user oriented reliability of a program is defined as the probability that the program will give the correct output with a typical set of input data from the user environment [36]. Software reliability is the probability of failure free software operation which affects the system reliability and it differs from hardware reliability in that it reflects the design perfection rather than manufacturing perfection [30]. Reliability of software is a function that combines number of faults and probability of these faults to occur i.e. to produce a failure [39]. Quyoum noted that "Reliability is a probabilistic measure that assumes that the occurrence of failure of software is a random phenomenon" [44]. Randomness means that the failure can't be predicted accurately. The high complexity of the software is a contributing factor towards the reliability problems. Good engineering methods can largely improve software reliability. Software reliability is a part of software quality. It relates to many areas where software quality is concerned. Hence measuring software reliability remains a difficult problem as we don't have a good understanding of the nature of software. Reliability is measured as the probability that a system will not fail to perform its intended functions over a specified time interval. Customers are critically conscious of the reliability of software; they are likely to be largely unconcerned with the degree of the reusability of the components making up the source code. Amrit noted that "software reliability is a useful measure in planning and controlling resources during the development process so that high quality software can be developed" [2]. Obtaining reliability estimates early in the development process can help determine if the software system is on track to meet its reliability goals and therefore increase management effectiveness.

Table 1 – Reliability attributes in Literature

Reliability Attributes ↓

Reliability Models →

	Boehm [9]	McCall [36]	Pabitra [41]	Roger [47]	Goel [17]	Ramani [46]	IEEE [25]
Accuracy	X	X		X		X	X
Consistency	X	X					
Completeness	X						
Error Tolerance		X			X	X	
Simplicity		X					
Defects Free			X		X	X	X
Usability				X			
Correctness					X		X
User confidence					X		

Table 2- Object Oriented Metrics in Literature

Name	Source	Metrics
MOOSE/CK	Chidamber et.al. [13]	WMC, DIT, NOC, CBO, RFC, LCOM
MOOD	Abrreu et.al. [1]	MIF, AIF, MHF, AHF, POF, COF
LK	Lorenz et.al.[33]	CS, NOO, NOA, SI, OS, OC, NP
QMOOD	Bansiya [5]	DSC,NOH,NSI,NMI, NNC,NAC,NLC,ADI,AWI,ANA,MFM,
LiW	Li et.al. [31]	NAC, NLM,CMC,NDC,CTA,CTM
SATC	Rosenberg et.al. [48]	CC, LOC,WMC,RFC,LCOM,DIT,NOC
STREW-J	Nagappan et.al. [38]	NTC/SLC,NTC/NR, TLC/SLC,NA/SLC,NTC/NSC,NC,NLC/NC
TANG	Tang et.al. [51]	AMC, CBM, IC
MARTIN	Martin [35]	Afferent Coupling, Efferent Coupling
HENDERSON	Henderson [21]	LCOM1, LCOM2, LCOM3

2.2. Threshold for Object Oriented Metrics

Today wide varieties of software metrics are proposed and broad range of tools are available to measure them. However, effective use of software metrics is hindered due to lack of meaningful thresholds. Threshold of software metrics can be used as indicators to identify possible anomalies in software. The designers should make use of the threshold limit of the metric values for confirming the project is on the right track. There are a few research works done to effectively measure the threshold of metrics. Mago Jagmohan and Kaur Parwinder made a study using Fuzzy Logic to estimate the threshold of CK metrics and proposed rule to predict the quality of the software [34].

Table 3 – Threshold Values for CK Metrics in the Literature

Works on
Thresholds ↓ CK Metric Threshold Values →

	WMC	RFC	DIT	LCOM	CBO	NOC
Camarzo [11]	Low	Low	Trade off	Low	Low	Trade off
Goel [17]	2	5	2	1	1	2
Benlarbi [8]	100	100			5	
Herbold [22]	100	100			5	
Rosenberg [48]	25-40	<50	2-5		<5	
S-D Metric [39]		3-365	0-3		0-31	
Together Soft [39]	100		4		30	
OEE [39]	307		0-4		1-4	1-4
Zhou & Lenug [56]	0-15	0-35	0-6	0-1	0-8	0-6
NASA [39]	20-100					
SEM [24]	Trade off	Low	Trade off	Low	Low	Trade off
Mago [34]	Low, < 11	Low, <12	Low, <4	Low, 0	Low, < 3	Low, < 3
Edith Linda [15]	0-15	0-35	0-6	0-1	0-8	0-6
Kaur [28]	14	31	1		7	
SATC [23]	Low	Low	Low (Trade off)	Low	Low	Low(Trade off)

2.4. Object Oriented Metrics and Software Reliability

Numerous studies have empirically validated the association between OO metrics and quality of software. The selected literature includes OO metrics based prediction models and estimation models that focus on validating the effectiveness of OO metrics for either predicting or estimating fault-prone classes or reliability of the system.

Sherry A.M., et.al. made study on object oriented software reliability models and proposed a new model stating the number of initial parameters serves as an important parameter of reliability model. He established a relationship between the number of initial faults present in an object and some metrics (CK) of OOPS [50]. Rosenberg Linda et.al. discussed how NASA projects in conjunction with SATC (Software Assurance Technology Centre) are applying software metrics to improve the quality and reliability of software products. Reliability is a by-product of quality and can be measured. Metrics used early can aid in detection and correction of requirement faults and guarantee reliability of the product [48].

Hitz Martin and Montazeri Behzad measured product attributes of object-oriented system using object oriented metrics based on their effects on product attributes [23]. Chillar Usha and Bhasin Sucheta established a relationship between complexity of software and object oriented metrics. Complexity affects quality attributes like reliability, testability etc. [12]. Sharma Aman Kumar et.al identified a few object oriented metrics suitable for measuring the software quality and provided thresholds that could be used to judge the metrics collected from designs [51]. Pandey Asheesh and Ahlawat Anil proposed application of neural networks for providing software reliability using object oriented metrics. He made use of complexity measures, cohesion and coupling measures as the independent variable. The validation has shown several well-known metrics can be profitably employed for the estimation of reliability [40]. Several other studies Helle [20], Varun Gupta [54], Dekkers [14], Klasky [30], Kaur [28], Raed [45], Arti [4], Khan [29], Subramnayam .R. [52], Micheal [37], Yu [56], Gyimothy [18] made use of object oriented metrics to make quality assessment of the software product. They provide useful feedback to the management to keep the software process and product more reliable.

3. Research Objective

The main objective of this study is to find the threshold of software reliability of a software project and validate. Threshold for reliability is estimated using the already established relationship between CK Metrics with the reliability at the class level from our previous work [3]. Threshold values of CK metrics are proposed based on the researchers and vendors in the literature. This threshold of reliability will be an indicator to the developer to verify the project is on the right track and if not to make necessary changes in the design.

4. Methodology

- First of all, CK metric suite is selected for estimating the threshold for Reliability of the software. This is due to (i) they are simple and intuitive to use, (ii) they are able to use at any stage of the development cycle, (iii) they can be supplemented with some other object oriented metrics, (iv) they are predominantly referenced researchers in the literature.

- Table -2 pools the threshold values of CK metrics from the literature. Based on the experience and the principle that if the metric values are too low may represent poor utilization of the advantages of object-oriented technology and too high values may represent too much complexity and overkill the OO technology. We must make of use of the great advantages of Object Oriented technology without paying the price in complexity, a new threshold is proposed for the CK Metrics.

- CK metrics values are assigned weighted values

- Threshold for the reliability (R^T) is calculated using relationship established between Reliability and CK metrics in our previous work [4]:-

  Reliability **α** 1/WMC
  Reliability **α** 1/RFC
  Reliability **α** 1/DIT
  Reliability **α** 1/LCOM
  Reliability **α** 1/CBO

- CK metric values are extracted/collected from the applications/projects on class level using a specially developed tool, viz., Java Class Analyzer and its reliability calculated is checked whether to lie within the thresholds.

- Projects are analyzed to test whether they are in the proposed threshold.

4. Research Hypothesis

A project whose R-Value (Reliability value) lies within the thresholds will have less number of defects and high reliability.

Mathematically: **If** R^T**(Min) < R-Value <** R^T**(Max), then P = Defect(Min) & Reliability(Max)**
(Where R^T = Threshold of Reliability, P = Project).

5. Experiment and Analysis

The first step towards the experiment is to propose threshold values for all CK metrics based on the table - 3, keeping it minimum and calculate the threshold for reliability

Proposed threshold for the CK Metrics

Table - 4

	WMC	RFC	DIT	LCOM	CBO	NOC
Threshold	6-30	6-36	1-6	1-3	3-9	1-3

Assigning weighted values to the Metrics

Rule -1
If Value of Metric lies between the lower limit and (mean of lower limit and upper limit) of the threshold, then the Weightage given to Metric is 1

Mathematically: If (Lower Value of Threshold ≤ Value of Metric ≤ Mean of Threshold) , then Weightage (Metric) = 1

Rule – 2
If Value of Metric lies between the (mean of lower limit and upper limit) and upper limit of the threshold, then Weightage given to Metric is 2

Mathematically: If (Mean of Threshold ≤ Value of Metric ≤ Upper Limit of Threshold), then Weightage (Metric) = 2

Rule-3
If Value of Metric lies outside the Threshold, then the Weightage given to Metric is 7.

Rule-4

In the case of NOC, $(\log(\text{upper threshold}))^2$ is considered for R^T**(Max)** and $(\log(\text{lower threshold}))^2$ is considered for R^T**(Min).** If any of the CK metric value is outside the thresholds, then this metric is neglected.

Calculating threshold of Reliability Using Rule 1 to Rule 4

R^T**(Max) =** $k*(1/(wt(WMC)+wt(DIT)+wt(RFC)+wt(LOCM)+wt(CBO)) + (\log(U\text{-}Lt(NOC)))^2$

R^T**(Min) =** $k*(1/(wt(WMC)+wt(DIT)+wt(RFC)+wt(LOCM)+wt(CBO)) + (\log(L\text{-}Lt(NOC)))^2$

Accordingly
Let us assume k = unity = 1

R^T**(Max) =1**$*(1/(1+1+1+1+1)) + (\log(3))^2$ = 0 .4276
R^T**(Min) = 1**$*(1/(2+2+2+2+2)) + (\log(1))^2$ = 0.1000

Therefore we state the Threshold for Reliability of software based on the relationship of Reliability and CK Metrics lies between 0.6777 and .10000

$$\mathbf{0.1000 < R^T < 0.4276}$$

Extracting CK Metric Values from the Projects

The CK metrics values are collected from the application using a specially developed Java Class Analyzer. For each class, ck metrics were collected.
Procedure to extract values of CK metric parameters from Java Projects

1. Import necessary headers and packages
2. Load the project
3. Use the appropriate methods and procedures to retrieve the metric parameters from the project

Data Extracted from the Projects which are considered less fault prone and reliable

Table – 5

Metrics	P1	P2	P3	P4	P5	P6	P7	P8
WMC	19	12	28	26	24	12	28	26
RFC	16	15	32	20	16	15	36	12
DIT	2	2	5	2	2	2	5	2
LCOM	2	2	3	2	2	2	3	2
CBO	4	2	5	1	1	2	5	1
NOC	3	3	2	2	2	1	2	2
Reliability	0.3704	0.4276	0.2017	0.2334	0.2334	0.2	0.2156	0.2572

Data Extracted from the Projects which are considered more fault prone and unreliable
Table – 6

Metrics	P9	P10	P11	P12	P13	P14	P15	P16
WMC	45	43	30	26	64	24	28	26
RFC	67	32	42	20	32	30	36	12
DIT	6	6	5	6	6	6	5	2
LCOM	4	3	3	3	3	4	3	2
CBO	4	6	5	5	9	10	5	1
NOC	12	9	2	7	4	0	1	0
Reliability	0.05	0.06	0.06	0.08	0.06	0.05	0.1	0.1428

6. Results and Discussions

The results obtained from the analysis of data supports the research hypothesis that a project whose R-Value (Reliability value) lies within the thresholds will have less number of defects and high reliability [8,12,15,26,49].

In the above experiment, 16 projects were analyzed out of which 8 are working properly and 8 are more of error prone. Analysis of data of Table-5 shows that the projects P1 to P8 whose reliability lies within the thresholds of reliability and they are working properly. As per the object oriented design wise the project P1 to P8 are correct and as per the norms.

Analysis of Table -6, only P15 and P16 whose reliability comes within the threshold, but at the very lower limit of the threshold. Therefore the projects P9 to P16 whose design is not properly as per the object oriented design and has to be redesigned for achieving higher reliability.

Hence the Research Hypothesis is validated.

The study proves that by keeping the threshold values of WMC, DIT, CBO, LCOM, RFC and NOC, the designers can improve the reliability of the software and as a whole quality of the system.

8. Conclusion

Highly reliable software is becoming an essential ingredient in many systems. This study made an assessment of the relationship between CK metrics and the reliability of objected oriented software system. We have selected entire CK metrics suite to estimate the threshold of reliability. The study proved empirically that by keeping WMC, DIT, CBO, LCOM, RFC and NOC within the threshold, the designers can attain high reliability of the system. Therefore we can say that CK metric parameters are useful indicators for predicting the reliability and thus quality of the system. The size of the data set is small, the result is of limited capability. Validation of the estimated reliability value of the projects using other metric suites suite like MOOD, QMOOD etc and other reliability estimation will be of future work with larger data sets.

References

1. Abreu F.B : “The MOOD Metrics Set”, Proc. ECOOP’95, Workshop on Metrics.
2. Amrit L. Goel (1985): “Software Reliability Models: Assumptions, Limitations, and Applicability”, IEET Transactions on Software Engineering, Vol SE-II, 12 December 1985 pages 1411-1423.
3. Antony P Johny and Dev Harsh: “Estimating Reliability Of Software System Using Object-Oriented Metrics”, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) ISSN 2249-6831 Vol. 3, Issue 2, Jun 2013, 283-294
4. Arti Chhikara., R. S. Chhilla and Sujata: “ Prediction Of The Quality Of Software Product Using Object Oriented Metrics”, International Journal Of Computer Engineering And Software Technology 2(1) Jan-June 2011; Pp. 1-6
5. Bansiya J. and C. G. Davis : “A Hierarchical Model for Object-Oriented Design Quality Assessment”, IEEE Transactions on Software Engineering, 28, (1), (2002), 4–17.
6. Barbacci Mario., Klein and Mark H., Longstaff, Thomas A., and Weinstock, Charles B: “Quality Attributes*”,* Technical Report of Software Engineering Institute Carnegie Mellon University Pittsburgh, Pennsylvania.
7. Basili V.R., L.C. Briand and W.L. Melo: “A Validation of Object oriented design Metrics as Quality Indicators”, IEEE Transctions on software engneering, 22(10), 1996, pp 751-761.
8. Benlarbi Saida., El-Emam Khaled., Goel Nishith and Rai N. Shesh: “Thresholds for Objected Oriented Measures”, In 11th International Symposium on Software Reliability Engineering, 2000. http://dl.acm.org/citation.cfm?id=851024.856210.
9. Boehm B. W., Brown, J. R., Kaspar H., Lipow, M., McLeod G., and Merritt M: “Characteristics of Software Quality”, North Holland, 1978.
10. Briand L., Daly, W. and Wust J: “Unified Framework for Cohesion Measurement in Object Oriented Systems”, Empirical Software Engineering, 65-117, 1998.
11. Camrazo Cruz Ana Erika: “Chidamber & Kemerer suite of Metrics”, Thesis submitted to Japan Advanced Institute of Science and Technology, 2008.
12. Chhillar, Usha and Bhasin, Sucheta: “Establishing relationship between complexity and faults for objected oriented software system”, IJCSI, Vol 8, Issue 5, No. 2 September 2011.
13. Chidamber, Shyam and Kemerer, Chris: “A Metric Suite for Object Oriented Design”, IEEE Transactions of Software Engineering, June 1994.
14. Dekkers, C.S: “The Secrets of Highly Successful Measurement Programs”, Cutter IT Journal, vol 12 no. 4, pp. 29-35
15. Edith Linda P and E. Chandra: “Class Break Point Determination Using CK Metrics Thresholds”, Global Journal of Computer Science and Technology, 73 Vol. 10 Issue 14 (Ver. 1.0), November 2010
16. Godbole, S. Nina: “Software Quality Assurance Principles and Practice” Narosa Publishing House
17. Goel B.Mohan and Bhatia Pradeep Kumar; “Analysis of Reusability of Object-Oriented System using CK Metrics”, International Journal of Computer Applications (0975 – 8887) Volume 60– No.10, December 2012
18. Gyimothy, Tibor., Rudolf, Ferenc and Istvan, Siket: “Empirical validation of object oriented metrics on open source software for fault prediction”, IEEE Trans. Softw.Eng., 31(10):897{910, 2005.
19. Harrison, R., S.J.Counsell and R.V.Nithi: “An evaluation of the MOOD set of OOSM”, IEEE Transaction on Software Engineering, vol.24 no.6, pp.491-496, June 1998. JürgenWüst, “SD METRICS TOOL”, in der Lache 17, 67308
20. Helle, Damborg Frederiksen**: “**Using Metrics in Managing and Improving Software Development Processes”, Doctoral Consortium at ECIS 2001’, June 2001, Department of Computer Science, Aalborg University, Denmark.
21. HENDERSON-SELLERS B., *Object-Oriented Metrics, measures of Complexity*. Prentice Hall, 1996.

22. Herbold Steffen; Grabowski Jens and Waack Stephan: "Calculation and Optimization of Thresholds for Sets of Software Metrics" , http://www.swe.informatik.unigoettingen.de/sites/default/files/publications/paper_tr.pdf
23. Hitz, Martin and Monazeri, Behzad: "Measuring Product Attributes of Object OrientedSystem", http://citeseerx.ist.psu.edu/viewdoc/summary? doi=10.1.1.52.4579.
24. http://books.google.co.in/books?id=WSLzZc1ANqEC&pg=PA5&lpg=PA5&dq=definitions+of+software+reliability&source=bl&ots=YTwcy22wY3&sig=CySnHdqYCqtJK6XEMVTrQUL93po&hl=en&sa=X&ei=n91wUYq8JYiIrQfy6YCoBg&ved=0CEsQ6AEwAw#v=onepage&q=definitions%20of%20software%20reliability&f=false
25. IEEE Standard (1991) 610.12-1990 Glossary of Software Engineering Terminology.
26. Kan, H. Stephen: "Metrics and Models in Software Quality Engineering", Prentice Hall, 2003
27. Kaur, Kuljit: "Process and Product Metrics to Assess Quality of Software Evolution", Conference Proceedings, International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT), 2012
28. Kaur Sarabjit; Singh Satwinder and Kaur Harshpreet: " A Quantitative Investigation of Software Metrics Thresholds Values at Acceptable Risk Level, IJERT, Vol. 2, Issue3, March – 2013.
29. Khan, R.A., K. Mustafa and S. I. Ahson**: "**An Empirical Validation of Object Oriented Design Quality Metrics", J. King Saud Univ., Vol. 19, Comp. & Info. *Sci*., pp. 1-16, Riyadh (1427H./2007).
30. Klasky, B. Hilda: "A Study of Software Metrics", A thesis submitted to the Graduate School-New Brunswick Rutgers, The State University of New Jersey, 2003.
31. Li, W and Sallie, Henry: "Metrics for Object-Oriented system", Transactions on Software Engineering, 1995
32. Lionel, C. Briand and JÄurgen, WÄust: "Empirical Studies of Quality Models in Object-Oriented Systems", In Advances in Computers, volume 56, September 2002.
33. Lorenz, M and J. Kidd: "Object Oriented Software Metrics", Prentice Hall, NJ, (1994).
34. Mago Jagmohan and Kaur Parwinder, 2012: "Analysis of Quality of the Design of the Object Oriented Software using Fuzzy Logic", iRAFIT, p 21- 25.
35. MARTIN R: *OO Design Quality Metrics - An Analysis of Dependencies.* Proc. of Workshop Pragmatic and Theoretical Directions in Object-Oriented Software Metrics, OOPSLA'94, 1994
36. McCall, J. A., Richards, P. K and Walters, G. F: "Factors in Software Quality", National l Technical Information Service, Springfield, VA, 1977
37. Michael, English. Chris, Exton., Irene, Rigon, and Brendan Cleary: "Fault detection and prediction in an open-source software project", In PROMISE '09: Proceedings of the 5th International Conference on Predictor Models in Software Engineering, pages 1{11, New York, NY, USA, 2009. ACM.
38. Nagappan, Nachiappan., Williams, Laurie, and Vouk, Mladen: "Towards a metric suite for early software reliability assessment". http://research.microsoft.com/en-us/people/nachin/publications.aspx.
39. Pan, Jiantao: " Software Reliability", http://www.ece.cmu.edu/~koopman/des_s99/sw_reliability/
40. Pandey, Asheesh and Ahlawat, Anil: "Reliability and Maintainability of Software using object oriented metrics by artificial neural network", IJMET, Vol.1, Issue 1, March 2013.
41. Panda Pavitra Kumar: "Software Reliability", 2010, http://www.slideshare.net/AnandKumar87/software-reliability-11841804
42. Parvinder, Singh Sandhu and Dr. Hardeep, Singh: " A Critical Suggestive Evaluation of CK Metric" (www.pacis-net.org/file/2005/158).
43. Ping, Yu., Tarja Systa and Hausi, Muller: "Predicting Fault-Proneness using OO Metrics: An Industrial Case Study", In Sixth European Conference on Software Maintenance and Reengineering *(CSMR 2002),* pages 99{107, March 2002.
44. Quyoum, Aasia., Din, Dar and Mehar, Ud Quadri S.M.K: "Improving software reliability using software engineering approach – A Review", International Journal of Computer Application, Vol. 10, No. 5, November 2010.
45. Raded Shatnawi: "An Investigation of CK Metrics Thresholds". ISSRE 2006 Supplementary conference proceedings (www.computer.org/portal/web)

46. Ramni Srinivasaan, Swapna S., Gokhale and Koshore S. Trivedi: “SREPT: Software Reliability Estimation and Prdiction Tool”, Centre for advanced computing and communication, Duke University, USA.
47. Roger C. Chenung, A user oriented software reliability model, IEET Transactions on Software Engineering, Vol SE-6, March 1980 pages 118-129.
48. Rosenberg Linda, Hammer Ted, and Shaw Jac, Software metrics and reliability http://www.google.co.in/#hl=en&rlz=1R2ASUM_enIN517&sclient=psy-ab&q=software+metrics+and+reliability+linda+rosenberg&rlz=1R2ASUM_enIN517&oq=sofware+metrics+and+reliability+lind&gs_l=hp.1.0.0i22i30.1716.13541.0.15429.50.38.6.0.0.2.982.13805.2-1j24j1j6j2.34.0...0.0...1c.1.9.psy-ab.VtRQal9pyYY&pbx=1&bav=on.2,or.r_qf.&bvm=bv.45512109,d.bmk&fp=3833cd2675a87390&biw=1440&bih=600
49. Rosenberg Linda, Stapko Ruth, Gallo Albert, “ Risk-based Object Oriented Testing”,http://www.google.co.in/#hl=en&gs_rn=12&gs_ri=psy-ab&cp=48&gs_id=42&xhr=t&q=Risk+-+based+object+oriented+metrics+ +Rosenberg& es_nrs= true &pf=p& output=search&sclient=psyab&rlz=1R2ASUM_enIN517 &oq=Risk+-+based+object+oriented+metrics++rosenberg&gs_l=&pbx=1 & bav=on .2, or.r_qf &bvm=bv.45960087, d.bmk&fp= 80178076eeb76333& biw=982& bih=452
50. Sherry,A.M., A.K. Malviya and R.C. Tripathi: “A study of object oriented software reliability models” http://www.bvicam.ac.in/news/ INDIACom%202009%20Proceedings/pdfs/papers/INDIACom09_262_Paper.pdf
51. Shmara Manik; Singh Gurudev; Arora Anish and Kaur Parneet: “A Comparative Study of Static Object Oriened Metrics”, International Journal of Advancements in Technology, Vol 3, No.1 (January 2012).
52. Subramnayam R. and M.S. Krishnan: “Empirical Analysis of CK Metrics for Object Oriented Design Complexity: Implications for software Defects”, IEEE Trans. Software Eng, 2003
53. TANG M-H., KAO M-H. and CHEN M-H: “An Empirical Study on Object-Oriented Metrics”. Proc. of the Software Metrics Symposium, 1999, 242-249.
54. Varun Gupta: “Object Oriented Static and Dynamic Software Metrics for Design and Complexity”, Ph.D Thesis submitted to Department of Computer Engineering and NIT, Kusukshetra, 2010.
55. Yadava Amitabha and Khan R.A., (2012), “Reliability Estimation Framework – Complexity Perspective” retrieved from http://airccj.org/CSCP/vol2/csit2509.pdf
56. Yu Ping, Tarja Syst, and Hausi A. Muller. Predicting fault-proneness using oo metrics: An industrial case study. In CSMR '02: Proceedings of the 6th European Conference on Software Maintenance and Reengineering, pages 99 107, Washington, DC, USA, 2002. IEEE Computer Society.
57. Zhou Yuming and Hareton Leung: “Empirical analysis of object-oriented design metrics for predicting high and low severity faults”, IEEE Trans. Softw. Eng.,32(10):771-789, 2006