HASSE DIAGRAM TECHNIQUE BASED RANKING OF SOFTWARE RELIABILITY GROWTH MODELS

Dr. Archana Kumar[1], Abhinav Juneja[2],Sapna Bajaj[3]
Director[1], DITM,Gannaur
Ph.D.Scholar[2,3],AFSET, Faridabad

Abstract: The development today's technology loving society has done is commendable. There has been revolution in technological applications and it has shaped our lives in its own ways. What so ever activity we do we are using technology and we have faith on the products that we use because we know that the products are reliable. Reliability is one of the most important feature that enhances our trust at Science and Modern Technological Development.
In the current research work reliability has been the prime focus throughout. Reliability of Software has been explored in this research. As software plays a very critical role in the today's society, so through this current research efforts have been made in order to enhance the level of reliability in the developed software. There has been a lot of research for reliability and reliable products in the past, the current research tries to propose a technique where the available Software Reliability Growth Models which are used to forecast the software failure behavior are given a preference order called ranking based on certain comparison criteria. The Software Reliability Growth Models which we have chosen for this current research work are all of Non Homogeneous Poisson Process category.

Keywords: Software Reliability, Software Reliability Growth Models,Comparison Criteria, Hasse Diagram, Average Rank, Partial Order

1.Introduction

Reliability has received increasing importance in the past few years in manufacturing, the civilian organizations and government organizations. With recent issues about government spending, agencies are trying to purchase systems with higher reliability and lower cost of maintenance. As consumers, we are mainly broadly bothered about buying products that last longer and are cheaper to maintain, i.e., products with higher reliability. Reliability plays an important role in Software industry as well. Software Reliability has been defined as one of the most critical and demanded quality aspects mentioned in IEEE Standard 729 [1].

2.Software Reliability

Software reliability has been explored thoroughly in Pham [2]. Software Reliability is a very desired component of the availability of computer system. Software Reliability is the probability of failure-free operation of software for a specified period of time in a specific environment. Software Reliability is also an important factor affecting system reliability. It apparently differs from hardware reliability as in later it reflects the design perfection, rather than perfection in manufacturing.

3.Software Reliability Growth Models

A Software Reliability Growth Model is one of the major procedures to survey programming unwavering quality quantitatively. The Software Reliability Growth Model requires having a decent execution as far as integrity of-fit, consistency etc. Keeping in mind the end goal to assess and also to anticipate the unwavering quality of programming frameworks, migrating or deviating information should be legitimately measured by different means amid programming improvement and operational stages. As system failures are encountered, the corresponding underlying faults responsible for these failures are neutralized so that the inherent reliability of the software improves during the process of system testing and debugging[7-11]. To predict Software Reliability, the conceptual Reliability Growth Models must be translated into a mathematical model. Software Reliability Growth Modeling involves comparing measured reliability at a number of instances of time with some familiar functions that show probable variations in reliability. Littlewood and Musa [3-6] have pioneered in development and implementation of Software reliability estimation techniques and have written extensively on reliability growth models.
A variety of Software Reliability Growth Models have been proposed by researchers and statisticians in the long history of evolution of the modern day Software Reliability Techniques. There are more than 200 models proposed by the contributors. All these models have taken into account the various factors that capture the information on the latent errors in the software and try to project the future behavior of the software.
The various classified categories of models are [13-16]:1)Early prediction models2)Architectural based models3)Hybrid white box approach4)Hybrid black box approach5)Software Reliability growth models6)Input domain models.Software Reliability Growth Models can be used as a tool to make a decision on when to stop testing.Some of the popular Software Reliability Growth Models based on NHPP for Software Reliability prediction which have been used in the current research for the ranking of models are as follows[17-28] :

Goel-Okumoto Model

This model was first proposed by the famous researchers in this field Goel and Okumoto. It is one of the most applied NHPP model in the field of software reliability prediction. It is also referred to as exponential NHPP model. Assuming failure detection as a typical Non homogeneous Poisson process having an exponentially decaying rate function, the corresponding mean value function is depicted in this model is represented as
$m(t) = a(1 - e^{-bt})$

and the associated intensity function of this model is represented as

$$\lambda(t) = abe^{-bt}$$

a is the projected total number of faults to be uncovered in due course and *b* is the fault detection rate.

Gompertz Growth Curve Model

Gompertz Growth Curve Model has been used in the Fujitsu and Numazu work on the reliability. Several computer manufacturers and software companies of Japan have started using this model due to it being the simplest S-shaped software reliability growth models. Its mean value function is represented as

$m(t) = ake^{-bt}$

and intensity function is represented as

$\lambda(t) = ab\ln(k)k^{e^{-bt}}e^{-bt}$

a is the expected number of faults to be detected with the passage of time and, *b* and *k the* are parameters which are estimated using regression analysis.

Logistic Growth Curve Model

In general, software reliability tends to improve and it can be treated as a growth process during the testing phase. That is, the reliability growth occurs due to fixing faults. Therefore, under some conditions, the models developed to predict economic population growth could also be applied to predict software reliability growth. These models simply fit the cumulative number of detected faults at a given time with a function of known form. Logistic growth curve model is one of them and it has an S-shaped curve. Its mean value function is represented as

$$m(t) = \frac{a}{1 + k^{-bt}}$$

and intensity function is represented as

$$\lambda(t) = \frac{abke^{-bt}}{(1 + ke^{-bt})^2}$$

a is the expected faults to be detected with the passage of time, and k and *b* are parameters which are traced by fitting the data of failure.

Generalized Goel NHPP Model

Goel proposed a generic version of the Goel-Okumoto model with the inclusion of an additional parameter c. The intention of giving this generic model was to showcase the situation that software failure intensity increments to some extent at the beginning and then begins to decline. The mean value function is represented as

$$m(t) = a\left(1 - e^{-bt^c}\right)$$

and intensity function is represented as

$$\lambda(t) = abct^{c-1}e^{-bt^c}$$

a is the expected total number of faults to be eventually detected and *b* and *c* are parameters that represent the quality of testing.

Yamada Delayed S-Shaped Model

The Yamada Delayed S-Shaped model is a variation of the non homogeneous Poisson process to attain an S-shaped curve for the detected cumulative number of failures so that the rate of failure y increases in the initial stages and exponentially decays later. We may consider it as a generalized exponential model having its failure rate initially increasing and finally decreasing. The software error detection process summarized by such an S-shaped curve can be considered as a learning process. The mean value function is represented as

$$m(t) = a(1 - (1 + bt)e^{-bt})$$

and intensity function is represented as

$$\lambda(t) = ab^2te^{-bt}$$

a and *b* are the expected total number of faults to be eventually detected and the fault detection rate, respectively.

Inflected S-Shaped Model

This model overcomes a limitation technically associated with the Goel- Okumoto model. It was introduced by Ohba and its fundamental principle is that the observed software reliability growth acquires S-shaped curve if the inherent faults in a program are mutually dependent, i.e., some of the faults are not traceable before some of the others are removed. The mean value function is

$$m(t) = \frac{a(1 - e^{-bt})}{1 + \beta e^{-bt}}$$

and intensity function is represented as

$$\lambda(t) = \frac{ab(1+\beta)e^{-bt}}{(1+\beta e^{-bt})^2}$$

Modified Duane Model

Duane presented a research report that evidenced failure data of a number of systems during their developments during 1962 by analysis of the data. It was seen that the cumulative Mean-Time Between-Failures versus the cumulative operating time acquires a curve close to a straight line if it is plotted on log-log paper. After some time, Duane model was proposed after modification and it's hypothesized mean value function is represented as

$$m(t) = a\left[1 - \left(b \,/\, (b+t)\right)^{c}\right]$$

and its intensity function is represented as

$$\lambda(t) = acb^{c} + (b+t)^{1-c}$$

a is the expected total number of faults to be eventually detected.

Musa-Okumoto Model

Musa-Okumoto incorporates the fact that the reduction in the rate of failure resulting from repair action following early failures are generally greater as they tend to become the most frequently occurring once in the model. The mean value function is represented as

$$m(t) = a\, ln(1 + bt)$$

and intensity function of the model is represented as

$$\lambda(t) = ab \,/\, (1 + bt)$$

a is the expected total number of faults to be eventually detected and *b* is the fault detection rate.

Yamada Imperfect Debugging Model 1

This model is assumed to be unrealistic for software reliability modeling as it assumes that the faults traced during the process of software testing are perfectly neutralized without introduction of any new faults. Yamada introduced software reliability assessment model with imperfect debugging using the assumption that new faults are sometimes injected when the faults originally remaining in the software system are neutralized during the testing phase. The assumption is that the fault detection rate is directly proportional to the summation of the numbers of faults remaining initially in the system and further faults introduced by imperfect debugging. This model is elaborated by a non homogeneous Poisson process. The mean value function

$$m(t) = \frac{ab\left(e^{\alpha t} - e^{-bt}\right)}{a+b}$$

and intensity is represented as

$$\lambda_{(t)}\, \frac{ab(\alpha e^{\infty} + e^{-bt})}{a+b}$$

a is the expected total number of faults to be eventually detected and *b* is the fault detection rate. is constant fault introduction rate.

Yamada Rayleigh et.al model

This model typically attempts to take into account the testing effort which has been put in order to trace the errors. The mean value function is represented as

$$m(t) = a\left(1 - e^{-r\alpha\left(1 - e\left(-\beta t^{2}/2\right)\right)}\right)$$

Where [a>0,b>0,α>0,β>0]

and intensity function is represented as

$$\lambda(t) = ar\alpha\beta t e^{-r\alpha\left(1 - e^{\;(Bt^{2}/2)}\right) - \beta t^{2}/2}$$

Where [a>0,b>0,α>0,β>0]

a is the expected total number of faults to be eventually detected and is constant fault introduction rate. *r* and β are constants.

Yamada Imperfect Debugging Model 2

This model typically takes an assumption that there is a constant introduction rate and constant fault detection rate *b*. The mean value function is represented as

$$m(t) = a(1 - e^{-bt})\left(1 - \frac{a}{b}\right) + \alpha a t$$

and intensity function of model is represented as

$$\lambda(t) = abe^{-bt}\left(1 - \frac{a}{b}\right) + \alpha a$$

Yamada Exponential Model

This model also typically puts an attempt to take into account the testing effort put into the project for tracing and detecting faults. The mean value function is represented as

$$m(t) = a\left(1 - e^{-r\alpha\left(1 - e^{-\beta t}\right)}\right)$$

and intensity function given is represented as

$$\lambda(t) = ar\alpha\beta e^{-r\alpha\left(1 - e^{-\beta t}\right) - Bt}$$

a is the total expected faults to be eventually detected over the period of time and is constant fault introduction rate. *r* and β are constants.

Pham–Nordmann–Zhang (P–N–Z) model
It assumes that introduction rate is a linear function of testing time, and the fault detection rate function is non-decreasing with an inflexion S-shaped model. The mean value function of this model represented as

$$m(t) = \frac{a(1-e^{-bt})\left(1-\frac{a}{b}\right)+\alpha at}{l+\beta e^{-bt}}$$

And intensity function is represented as

$$\lambda(t) = \frac{abe^{-bt}\left(\frac{1-a}{b}\right)+\alpha}{1+\beta e^{-bt}} + \frac{ab\beta e^{bt}\left(1-\frac{\alpha}{B}\right)\left(1-e'^{bt}\right)+\alpha t}{(1+\beta e^{-bt})^2}$$

a is the total expected faults to be eventually detected over the period of time and is constant fault introduction rate. *β* is constant.

Pham–Zhang (P–Z) model
It assumes that introduction rate is exponential function of the testing time, and the fault detection rate is non decreasing with an inflexion S-shaped model. The mean value function is represented as

$$m(t) = \frac{1}{1+\beta e^{-bt}}\left((c+a)(1-e^{-bt}) - \frac{ab}{b-a}(e^{-\infty}-e^{-bt})\right)$$

and intensity function of this is represented as

$$\lambda(t) = \frac{b(C+a)(1+\beta)e^{-bt}-\left[be^{-bt}(1+Be^{-\alpha t})-\alpha e^{-\infty}(1+1^3e^{-bt})\right]}{(1+Be^{-bt})}$$

a is the total expected faults to be eventually detected over the period of time and is constant fault introduction rate. β and *c* are constants.

Pham Zhang IFD model
It assumes constant initial fault content function and imperfect fault detection rate combining the fault introduction phenomenon. The mean value of function is represented as

$$m(t) = a - ae^{-bt}(1+(b+d)t+b\,dt^2)$$

and intensity function of this model represented as

$$\lambda(t) = ae^{-bt}[bt(b-d)+d(b^2t^2-1)]$$

a is the total expected faults to be eventually detected over the period of time and is constant fault introduction rate. *d* is constant.

Zhang-Teng-Pham model
This model has been designed on the assumption that fault introduction rate, and the corresponding fault detection rate functions are non decreasing having a corresponding inflexion S shape model. The mean value function is represented as

$$m(t) = \frac{a}{p-\beta}\left[\left(1-\frac{(1+\alpha)e^{-bt}}{1+\alpha e^{-bt}}\right)^{\frac{c}{b}(p-\beta)}\right]$$

and intensity function of this model is represented as

$$\lambda(t) = \frac{ac}{1+\alpha e^{-bt}}\left[\left(\frac{(1+\alpha)e^{-bt}}{\frac{-b^{\Sigma}}{1+\alpha e}}\right)\frac{c}{b}(P-\beta)\right]$$

a is the total expected faults to be eventually detected over the period of time and is constant fault introduction rate. *c*, *p* and β are constants.

4.Ranking the Models and Selecting the Best Candidate
There has been a long research in this dimension of selecting the best candidate model. This is an area where we have to consider multiple comparison criteria for the choice of the best model. Due to the multiple dimension decision criteria, it becomes difficult to prioritize the models. There are 8 identified criteria's for the ranking of the models namely MSE, MAE, MEOP, AE, Noise, RMSPE, SSE, TS [29-33]which have been used here in this paper for the comparison.

5. Proposed Technique for Ranking the Models
Researchers have proposed few selection criteria's in literature based on different techniques.
Yamada et.al.[34] gave Maximum Likelihood Classification Method,Sharma et.al.[35] used Distance Based Approach, Mohd. Anjum et. al. [36] used Weighted Criteria method, Liang Fuh Ong [37] used Partical Swarm Optimization Based Ranking for the selection of models.In the current work 16 Non-Homogeneous Poisson Processes Models [12, 29-33] have been used for ranking and comparison using the above mentioned comparison criteria.
We have used Hasse Diagram ranking methodology for ranking the models. Hasse diagram technique is a trusted multi-criteria evaluation method from the partial order set mathematical theory. The main feature of this technique is the visualization of partial ordered sets by so-called Hasse diagrams. For the purpose of making decisions, the Hasse diagram and the partial order set respectively, can be additionally transformed to total order[38-42]. The Total rank may be obtained from formula
Rk(av) = (S+1)*(N+1)/(N+1-U)

where S is the total number of successors of the object x, N is the number of objects (of the remaining quotient set), and U is the total number of objects incomparable with x.

The Sample dataset collected for this research in table 1 has been taken from open literature Kapil Sharma et. al.[35],which is software failure data for Tandem computers over several weeks. From this data parameters have been estimated using maximum likelihood method[35]. The values of the different comparison criteria may be calculated from this using their standard formulas [29-33,36] as calculated in table 2.

Weeks	**No. of CPU Hours**	**Detected Defects**	**Weeks**	**No. of CPU Hours**	**Detected Defects**
1	519	16	11	6539	81
2	968	24	12	7083	86
3	1430	27	13	7487	90
4	1893	33	14	7846	93
5	2490	41	15	8205	96
6	3058	49	16	8564	98
7	3625	54	17	8923	99
8	4422	58	18	9282	100
9	5218	69	19	9641	100
10	5823	75	20	10000	100

Table 1 Tandem Computer Software Failure Data

S.No	**Model/ Criteria**	**MAE**	**MSE**	**Noise**	**MEOP**	**AE**	**RMSPE**	**SSE**	**TS**
1	**Musa Okumoto**	0.351071	0.924321	0.112795	1.307126	0.028442	0.570933	18.97398	1.008592
2	**Yamada Model**	3.543599	179.2539	0.14698	17.75146	0.305942	8.454833	1597.26	11.28644
3	**Yamada Rayleigh**	3.558081	17.31985	5.479	19.0736	0.744818	2.447927	352.0871	5.067527
4	**Delayed S-shape**	1.693175	5.794858	2.769102	5.807915	0.018245	3.077485	116.9649	2.791747
5	**Yamada Model**	1.825164	21.93011	2.236911	9.628311	0.145899	4.583359	445.0041	5.743455
6	**Yamada Exp**	47.654	1919.854	0.019004	2.395486	0.748399	94.054	38457.07	58.409
7	**P-N-Z Model**	0.911859	3.61805	0.206977	3.108742	0.061985	1.464794	76.73437	2.214798
8	**P-Z Model**	0.383722	0.357683	0.16753	0.788449	0.464667	0.330636	9.158949	0.659803
9	**Pham Zhang IFD**	2.06796	6.540207	2.834834	6.347738	0.020288	3.321157	134.2953	3.010679
10	**Zhang-Teng-Pham**	0.383908	0.358311	0.003751	0.789582	0.4992	0.331376	9.173131	0.66047
11	**Generalized Goel**	8.371275	85.55917	0.385555	24.502	0.979	13.28857	1723.841	11.74652
12	**Goel-Okumoto**	0.242577	0.452238	0.166812	0.799432	0.017961	0.335463	9.320362	0.666706
13	**Gompertz**	0.734919	3.348342	0.215442	3.856035	0.048316	1.762726	69.46258	2.096149
14	**Inflection S-shape**	0.27435	0.398804	0.169332	0.766704	0.017401	0.320838	8.830393	0.64588
15	**Logistic Growth**	0	0	1.178951	0	0	0	0	0
16	**Modified Duane**	0.31556	0.534152	0	0.93777	0.314024	0.398268	11.67395	0.761004

Table 2 Estimated Value of Comparison Criteria for Various Sofware Reliability Growth Models

S.No.	Model Name	Model Notation in Pro Rank	Average Rank[Rk(av)]	Overall Ranking
1	Goel-Okumoto	E12	1.888889	1
2	Inflection S-shape	E14	1.888889	1
3	Musa okumoto	E1	2.125	2
4	Logistic Growth	E15	2.833333	3
5	Zhang-Teng-Pham	E10	3.4	4
6	Modified Duane	E16	3.4	4
7	P-Z Model	E8	4.25	5
8	P-N-Z Model	E7	8.5	6
9	Gompert	E13	8.5	6
10	Yamada Model	E2	11.33333	7
11	Delayed S-shape	E4	11.33333	7
12	Yamada Exp	E6	12.75	8
13	Pham Zhang IFD	E9	14.16667	9
14	Yamada Model	E5	14.875	10
15	Generalized Goel	E11	15.3	11
16	Yamada Rayleigh	E3	15.45455	12

Table 3 Overall and average rank obtained using Hasse Diagram Technique

Figure 1 Hasse Diagram for Data Set of Tandem Computer Failure Data

6.Results

We may now use the Hasse Diagram Ranking Technique for ranking these models using the multiple attributes(Comparison Criteria) and giving due recognition to all the attributes (Comparison Criteria).

Figure 1 shows the Hasse diagram for ranking obtained through the various comparison criteria. And Table 3 shows the overall ranks of various models. We may see in table 4 that there are 4 such models that share common ranks with other 4 models. In the large there are only 12 ranks as this ranking technique provides equal opportunity to all the comparison criteria to be given due consideration for being given weightage while ranking and selection.

7.Conclusion

The current work provides a new approach for ranking and selection of the best model form the available options of Software Reliability Growth Models. If the correct models are chosen for a particular development environment then it is possible to take a decision on the latent faults in the software and decide on its future release date. Through this technique it is possible to identify which particular models have equal utility in prediction of software faults, This may give the developers a choice of several models during testing.

8.References

[1] Glossary of Software Engineering Terminology, IEEE Standard 729,1983.
[2] H. Pham," System Software Reliability.",Springer London, 2006.pp.81-84
[3] Littlewood B, Stigini I , The Risks of Software, Science Am., November 1992,page 38-43.
[4] B.Littlewood,J.L.Verall,"A Bayesian Reliability Growth model for computer software,"Journal Royal Stat. Society,1973,pp.332-346
[5] J.D.Musa,"The Measurement and Management of Software Reliability,"Proceeding of IEEE,68(9), 1980,pp. 1131-1143
[6] J.D.Musa,"Software Reliability Measurement,"Journal of Systems and Software, Volume 1,1979-1980,pages 223- 241
[7] M. Xie, G. Y. Hong, and C. Wolin, "Software reliability prediction incorporating information from a similar project," J. Syst. Softw., vol. 49,pp. 43–48, 1999
[8] M. Xie, Software Reliability Modelling. :World Scientific Publishing Co. Ltd., 1991
[9] A. Wood, "Predicting software reliability," IEEE Computer, vol. 11,pp. 69–77, 1996
[10] C. Huang, S. Kuo, M. R. Lyu, and J. Lo, "Quantitative software reliability modeling from testing to operation," in Proc. 11th Int. Symp. Software Reliability Engineering, 2000, pp. 72–82
[11] D. R. Jeske and X. Zhang, "Accounting for realities when estimating the field failure rate of software," in Proc. 12th Int. Symp. Software Reliability Engineering, 2001, pp. 332–339
[12] Hossain, S., & Dahiya, R. C. (1993, December). Estimating the parameters of a non-homogeneous Poisson process model for software reliability. IEEE Transactions on Reliability, 42, 604–612
[13] Smidts, R. W. Stoddard, and M. Stutzke, "Software reliability models: An approach to early reliability prediction," IEEE Trans.Reliability, vol. 47, no. 3, pp. 268–278, 1998.
[14] A. Wood, "Software reliability growth models," Tandem Computers,vol. TR 96.1, pt. Part No. 130056, September 1996.
[15] S. S. Gokhale, P. N. Marinos, and K. S. Trivedi, "Important milestones in software reliability modeling," in Proc. of Softw. Engineering and Knowledge Engineering (SEKE '96), Lake Tahoe, NV, 1996, pp.345–352.
[16] S. Gokhale, W. E.Wong, K. S. Trivedi, and J. R. Horgan, "An analytical approach to architecture-based software reliability prediction," in IEEE International Computer Performance and Dependability Symposium, September 1998, pp. 13–22
[17] C. Y. Huang and S. Y. Kuo, "Analysis of incorporating logistic testing effort function into software reliability modeling," IEEE Trans. Reliability,vol. 51, no. 3, pp. 261–270, 2002
[18] K. Pillai and V. S. S. Nair, "A model for software development effort and cost estimation," IEEE Trans. Softw. Engineering, vol. 23, no. 8,pp. 485–497, 1997
[19] S. Hwang and H. Pham, "Quasi-renewal time-delay fault-removal consideration in software reliability modelling," IEEE Trans. Systems, Manand Cybernetics-Part A: Systems and Humans, vol. 39, no. 1, January 2009
[20] H. Pham and C. Deng, "Predictive-ratio risk criterion for selecting software reliability models," in Proc. Ninth International Conf. On Reliability and Quality in Design, August 2003
[21] X. Zhang, X. Teng, and H. Pham, "Considering fault removal efficiency in software reliability assessment," IEEE Trans. Systems, Man, and Cybernetics—Part A, vol. 33, no. 1,2003, pp. 114–120
[22] X. Cai, M. R. Lyu, "Software Reliability Modeling with Test Coverage Experimentation and Measurement with a Fault-Tolerant Software Project". ISSRE, 2007,17-26
[23] C. Y. Huang, S. Y. Kuo, M. R. Lyu, "An assessment of testingeffort dependent software reliability growth models". IEEE Transactions on Reliability, 2007, 56(2): 198-211
[24] Lyu, M. R, Nikora, A. "Applying Reliability Models More Effective". IEEE Software, 1992, 9(4): 43-52 [7] Y. S. Su, C. Y. Huang, "Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models". The Journal of Systems and Software, 2007, 80:606-615
[25] C. J. Hsu, C. Y. Huang, "Reliability analysis using weighted combinational models for web-based software". 2009, 1131-1132
[26] Eduardo Oliveira Costa, Silvia R. Vergilio, Aurora Pozo, Gustavo Souza, "Modeling software reliability growth with Genetic Programming", ISSRE, 2005, pp.1-10
[27] Eduardo Oliveira Costa, Gustavo Souza, Aurora Pozo, Silvia R. Vergilio, "Exploring Genetic Programming and Boosting Techniques to Model Software Reliability". IEEE Transactions on Reliability, 2007, 56(3): 422-434
[28] S.Brocklehurst, P. Y. Chan, B. Littlewood, and J. Snell, "Recalibrating Software Reliability Models," IEEE Trans. Software Engineering, vol. SE-16, no. 4, pp. 458–470, April 1990
[29] Sultan Aljahdali and Alaa F. Sheta, ―Predicting the Reliability of Software Systems Using Fuzzy Logic, 2011 Eighth International Conference on Information Technology: New Generations
[30] Ralf H. Reussner, Heinz W. Schmidt and Iman H.Poernomo,"Reliability prediction for component-based software architecture", The Journal of Systems and Software 66,2003,241-25
[31] Norman F. Schneidewind, ―Software Reliability Model with Optimal Selection of Failure Data, IEEE Transactions On Software Engineering, Vol. 19, No. 11, Page 1095-1105, November 1993
[32] Dr. Ajay Gupta, Dr. Digvijay Choudhary and Dr. Suneet Saxena, ―Software Reliability Estimation Analysis and Ranking of Software Reliability Models Based on Weighted Criteria Value using Yamada Delayed S Shaped Model under Imperfect Debugging and Time Lag",International Journal of Computer Applications (0975-8887) Volume 23– No.7, June 2011

[33] Shaik.Mohammad Rafi et. al., ―Software Reliability Growth Model with Logistic-Exponential Test-Effort Function and Analysis of Software Release Policy, (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 02, 2010, 387-399

[34] S.Yamada,S.Osaki,"Software Reliability Growth Modeling: Models and Applications,"IEEE Transactions on Software Engineering, vol.SE-11, no.12, pp. 1431- 1437, Dec.,1985

[35] K.Sharma,R.Garg,C.K.Nagpal,R.K.Garg, "Selection of Optimal Software Reliability Growth Models Using a Distance Based Approach,"IEEE Transactions on Reliability ,vol.59, no.2, pp.266-276, June 2010

[36] Mohd. Anjum, Md. Asraful Haque, Nesar Ahmad ,"Analysis and Ranking of Software Reliability Models Based on Weighted Criteria Value", I.J. Information Technology and Computer Science, 2013, 02, 1-14, Published Online January 2013 in MECS (http://www.mecs-press.org/)DOI: 10.5815/ijitcs.2013.02.01

[37] Liang Fuh Ong, Mohd Adham Isa, Dayang N.A.Jawawi, Shahliza Abdul Halim, Improving Software Reliability Growth Model Selection Ranking Using Particle Swarm Optimization, Journal of Theoretical and Applied Information Technology,15th January 2017. Vol.95. No.1,ISSN: 1992-8645,www.jatit.org E-ISSN: 1817-3195155

[38] Halfon E & Reggiani MG (1986). On Ranking Chemicals for Environmental Hazard. Environmental Science and Technology 20, 1173-1179

[39] Pavan M & Todeschini R (2008). Scientific Data Ranking Methods: Theory and Applications. 27;1st Edition; chapter 9, 193 – 207

[40] U.Simon,R.Bruggemann,S.Pudenz,"Comparable Assessment of Water Management Strategies by Hasse Diagram Technique – An example City of Berlin"H-P.Luhr(Edition),Order Theoritical Tools in Environmental Science and Decision Systems,Proceeding of 3rd Workshop,Leibniz Institute of Freswater Ecology and Inland Fisheries, Berlin,6th -7th Nov,2006

[41] Brueggemann et. al.,"Estimation of Averaged Ranks by a Local Partial Order Model", Match Journal of Mathematics and Computer Chem. ,54, 2005, 489-518

[42] K.Voigt, R.Bruggemann, S.Pudenz,"A multi criteria evaluation of environmental databases using Hasse Diagram Technique (Pro Rank) Software",Enviromental Modelling and Software,2006,21,1587-1597