Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P) Vol.8 No.S3, 2019, pp. 114-116

DOI: https://doi.org/10.51983/ajcst-2019.8.S3.2051

An Efficient Predictive Paradigm for Software Reliability

Srivyshnavi Pagadala[1], Sony Bathala[2] and B. Uma[3]
[1]Assistant Professor (Senior), [2&3]Assistant Professor
[1,2&3]Departmentof Computer Science and Engineering, School of Engineering and Technology,
Sri Padmavati Mahila Visva Vidyalayam, Tirupati, Andhra Pradesh, India
E-Mail: vyshu.sri@gmail.com, sonybathala@gmail.com, umaranganatham@gmail.com
(Received 25 April 2019; Revised 6 May 2019; Accepted 22 May 2019; Available online 28 May 2019)

***Abstract* - Software Estimation gives solution for complex problems in the software industry which gives estimates for cost and schedule. Software Estimation provides a comprehensive set of tips and heuristics that Software Developers, Technical Leads, and Project Managers can apply to create more accurate estimates. It presents key estimation strategies and addresses particular estimation challenges. In the planning of a software development project, a major challenge faced by project managers is to predict the defects and effort. The Software defect plays critical role in software product development. The estimation of defects can be determined in the product development using many advanced statistical modelling techniques based on the empirical data obtained by the testing phases. The proposed estimation technique in this paper is a model which was developed using Rayleigh function for estimating effect of defects in Software Project Management. The present study offers to decide how many defects creep in to production and determine the effort spent in months. The estimation model was used on Software Testing Life Cycle (STLC) to complete product. The accuracy of the model explains the variation in spent efforts in months associated with number of defects. The model helps the senior management in estimating the defects, schedule, cost and effort.**
***Keywords:* Defect Prediction, Rayleigh Function, STLC**

I. INTRODUCTION

Software Testing Life Cycle (STLC) is one of the key divisions of Software Development Life Cycle (SDLC). Software defects influences cost of the software, quality, performance of the product. The recent research shows only 40% of the IT companies reported failed software schedule and budget estimation while only 14% reported good performance. And 62% of organizations practiced IT projects that failed to meet their time schedules, whereas 49% practiced budget overruns. Most of the enterprises, IT or non - IT are suffering big losses due to poor estimations [1].

Pham has created a prediction model based on lifecycle phases, which divide the complete development life cycle in different phases, such as requirements review, design, implementation, unit testing, integration, system testing and functional testing etc [4]. Earlier studies suggest many approaches to defect prediction: time-based and metrics-based. A period - based expectation models assesses the quantity of deformities from the quantity of imperfections officially found in various past phases of the life cycle. Metric based approach used metrics from historical data, applied to a prediction model [5].

Software defects are more costly if discovered and fixed in the later stages of the testing and development life cycle or during the production [6]. Software testing is one of the most critical and time consuming phase of the software development life cycle and accounts for 50% of the total cost of development. Imperfection predictions enhance the efficiency of the testing phase in addition to help developers for evaluating the quality and defect proneness of their software product [6].

II. MATERIALS AND METHODS

The aim of the this paper is to estimate defects passes to the customer using Rayleigh model We collect data from one of the complete projects all the testing phases' requirements, design, authoring, execution, user accepting testing and preproduction, with spent hours. The application we used for this defect estimation the complete product. We analyzed the testing phases with respective spent in months data, and verify is there any association with defects and hours. The estimation model one of the assumptions is strongly correlated in between independent and dependent parameters. We collect data from projects for same environment kind of Using Rayleigh function for estimating defect in STLC,

$$F(x) = (2/T)\,(T/C)^{2}\mathrm{Exp}^{-(T/C)^{2}} \quad (1)$$

Where, C= (Maximum of time)*√2, T=time (Actual effort in months).Substitute the testing phases values like Requirement, for example

F (estimated requirement phase)

$$= = (2/1.8)\,(1.8/7.071)^{2}\mathrm{Exp}^{-(1.8/7.071)2} \quad (2)$$

Here T=1.8 (Spent hours in requirement phase) and C=7.071(Maximum spent hours of all phases *2.4142), we will get estimated defects for requirement phase for the production in equation (1).Using the same approach to remaining phases we will get the corresponding estimated defects values. The complete model based on cause and effect relationship between defects and spent hours.

A. Assumptions on Model

a. Defects rate during development/testing is correlated with defect rate after release.

b. If defects are discovered and removed earlier in development/testing, fewer will remain in later stages.

B. Benefits of the Model

a. Defect estimation model can help your organization to estimate proper effort and schedules. It helps to defects count passed to the customer.

b. It helps to preventive stepladder to reducing the effort for future stages. It helps to estimate cost with respective defects for next release Defect prediction helps in estimating the quality before it is released.

C. Results and Discussion

TABLE I ESTIMATED AND CUMULATIVE DEFECTS ON PHASE WISE

Phase	Actual Effort (in months)	Actual Defects	Estimated Defects	Cumulative Defects	Estimated Cumulative Defects
Requirement	1.8	4	8	4	8
Design	2.8	14	14	18	23
Authoring	5	40	36	58	59
Execution	2.8	32	14	90	73
UAT	2.5	15	12	105	85
Pre -Production	2	4	9	109	95

Regularly, the quantity of imperfections acquired per process can fluctuate from delivery to delivery; items to items, contingent upon the quantity of processes executed their complexity [2]. For the STLC the first Requirement phase actual effort in months is 1.8 and number of defects are occurred 4 only, in the beginning phase fewer defects and followed by next Design phase little bit increased based on requirements it has 2.8 months and defects are 14. The Authoring phase very critical and more time spent in the particular phase and defects are 40 which is maximum defects compare all the software testing life cycle. This is normally doing on IT organizations.

We used Rayleigh model for estimation defects to passed customer. The present study explains how many defects passed to the customer with respect to next release. Typically, there is no proper method or model for estimating defects passed to the customer. Rayleigh model one of the appropriate models for estimating the defects. The project data follows normal curve which is mentioned Reyleigh curve.

For the project point of view collect all the testing phases' data effort spent their defects data. We can calculate maximum number of time spent in all phase's data. Here, we observe the maximum spent in months the authoring phases which are 5 months. We can determine T following formulae

T is the Maximum of time spent of 1.424, i.e. above the project data the maximum of spent in months is Authoring is 5 months, and multiplied by 1.4142, we will get the values is 7 which is Time value for all phases. Substituting the numerical values, we can determine the expected defects data for all phases. For validating of estimation model, we used statistical tools for accuracy of the model as well as model decision. For model decision we used coefficient of determination (R2) which gives percentage of variation in defects is associated with spent hours in months. How strongly relationship between defects and hours spent. The present projects data of coefficient of determination is 0.77 i.e. 77% of variation in defects is explained with spent hours in months. We conclude that strongly associated between defects and spent hours. For model accuracy, we used Theils statistical too for validating accuracy of the estimation model. Of the model gives 80.47% accuracy it means the actual defects are close to estimated defects data. Model accuracy plays important role in to validate the performance of the any statistical predicting or estimation model. Of the project 95 defects occurred with 95% of Confidence limits. Probably we estimate 20 defects passes to customer,

D. Estimated Values

1. Estimated defects passed to the Customer 20
2. Total estimate defects (95% Confidence Limits) 95
3. Model Decision (R-Square value) 0.77
4. Model Accuracy (%) 80.47%

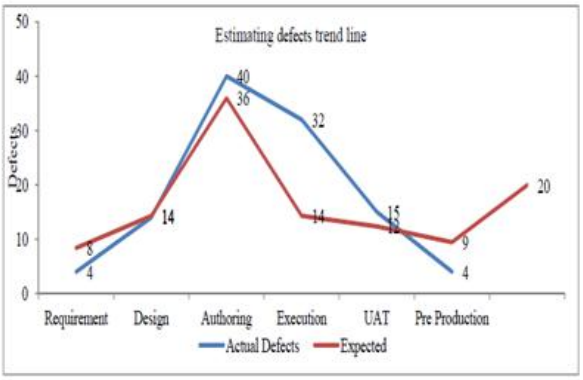

Fig. 1 Trend line on actual and estimated defects

III. CONCLUSION

This present paper uses the existing STLC based model as beginning state in developing a new scientific approach to estimation the defects of software product. We have

considered two parameters in Rayleigh model and defects calculated phase only. The most important portion of the model was to have a good estimation of the number of defects passes to the customer, represented in the function of maximize spent hours in months of complete phases. It was good 80.47%accuracy. It is most suitable for agile kind of projects.

REFERENCES

[1] Stephen H. Kan, *Metrics and Models in Software Quality Engineering*, 1st Edition, Pearson, 2003.

[2] Ana Maria Vladu, "Software Reliability Prediction Model using Rayleigh Function", *UPB Sci. Bull., Series C*, Vol. 73, No.4, 2011.

[3] K. Naik, and P. Tripathy, *Software Testing and Quality Assurance. Theory and Practice*, John Wiley and Sons Inc., New Jersey, pp. 471-479, 2008.

[4] E. Georgiadou, "Software Process and Product Improvement: A historical Perspective", *International Journal of Cybernetics*, No.1, pp. 172 - 197, 2003.

[5] Graham Clark, William H. Sanders, "Implementing a Stochastic Process Algebra within the Möbius Modeling Framework", *Proceedings of the Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modelling and Verification*, pp. 200-216, September 12-14, 2001.

[6] Marta Z. Kwiatkowska, Gethin Norman, and David Parker, "Probabilistic Symbolic Model Checking with PRISM: A Hybrid Approach", *Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp.52-66, April 08-12, 2002.

[7] S. Jasmine, R. Vasantha, "DRE- a quality metric for Component based Software Products", *Proceeding of world Academy of Science, Engineering and Technology*, Vol. 23, 2007.

[8] Lionel C. Briand, Khaled El Emam, Bernd G. Freimut,Oliver Laitenberger, "A Comprehensive Evaluation Of Capture-Recapture Models For Estimating Software Defect Content", *IEEE Transactions On Software Engineering*, Vol. 26, No. 6, June 2000.

[9] R. Abreu, P. Zoeteweij, R. Golsteijn, A.J. Van Gemund, "A practical evaluation of spectrum-based fault localization", *J. Syst. Softw.*,Vol. 82, No. 11, pp. 1780-1792, November 2009.

[10] Sandeep Kumar Nayak, Raees Ahmad Khan and Md. Rizwan Beg., "Requirement Defect identification An Early Stage Perspective", *IJCSI International Journal of Computer Science Issues*, Vol. 9, No. 1, September 2012.

[11] Sakthi Kumaresh and R. Baskaran., "Defect analysis and prevention for software process quality improvement", *International Journal of Computer Application*, Vol. 8, No. 7, Oct 2010.

[12] D. Abramson, I. Foster, J. Michalakes, and R. Sosic, "Relative debugging and its application to the development of large numerical models", *8th Int. Conf. High Perform. Comput. Netw. Storage Anal.*, Vol. 51, December 1995.