An Effectual Estimation of Software Defect (EESD)

Mr.. Raghvendra Omprakash Singh
Computer Engineering Department
G.M Vedak Institute of Technology,Tala
Mumbai University . India

Mr.Yuvraj B. Hembade
Department of Computer Engineering
Dnyanganga College of Engineering and Research
Pune university. India

Guide Prof. Pradeep Kumar Sharma
Computer Engineering Department
Sobhasaria Group of Institute , Sikar
R.T.U Kota .India

Abstract
An correct prediction of range of defects during a product throughout system testing contributes not solely to the management of the system testing method however additionally to the estimation of product's maintenance. Here, a replacement approach, referred to as 'Functional Estimation package Defect is conferred that computes associate degree estimate of total range of defects in associate degree in progress testing method. The estimation of the whole range of defects at early stages of the testing method helps managers to create resourceallocation and point in time selections. the employment of nonbayesian approaches has well-tried to be correct however presents a definite latency to realize an inexpensive accuracy. Here we have a tendency to describe useful Estimation package Defect model
Keywords*: Estimation Model, Defect Estimation, Estimator, Approximator.*

I. Introduction

In this paper, the focus is on estimated number of defects in a software product. Availability of this estimate allows a test manager to improve planning, monitoring, and controlling activities to provide more efficient testing process. Also, since, in many companies, system testing is one of the last phases (if not the last), time to release can be better assessed; the estimated remaining defects can be used to predict the required level of customer support. Ideally, a defect estimation technique has several important characteristics. First, the technique should be accurate as decisions based on inaccurate estimates can be time consuming and costly to correct. However, most estimators can achieve high accuracy as more and more data becomes available and the process nears completion many researchers have addressed this important problem with varying end goals and have proposed estimation techniques to compute the total number of defects.

In present approach is to estimate the number of remaining defects in ongoing testing process is during inspection using actual inspection data. To predict which files will contain the most faults in the next release [1][2]. To use data collected from previous projects and estimate the number of defects in a new project. However, these data sets are not always available or, even if they are, may lead to inaccurate estimates. Another alternative that appears to produce very accurate estimates is based on the use of Bayesian Belief Networks (BBNs) [3] However, these techniques require the use of additional information, such as expert knowledge and empirical data.

Our goal in this paper is to implement various estimation methods which are used to develop these defect estimation techniques. We had implemented different method makes about the data model, probability distribution, and mean and variance of data and estimator. We will also discuss statistical efficiency of the estimators developed from these estimation methods.

The remainder of this paper is organized as follows: A description of the present approach in section 2. An overview of estimation theory is provided as background in Section 3. Proposed approach is presented in Section 4. Section 5. Presents a survey of the related work Section 6. Conclusions.

II. Present Approach

There are two broad groups of estimation methods called Classical Approach and Bayesian Approach as shown in Figure 1. If there is prior statistical knowledge about the parameter which we are interested to estimate, then an estimator based on Bayesian Approach can be found. Note that the prior knowledge can be in the form of prior mean and variance and probability distribution of the parameter. There are

ISSN: 2278 – 1323

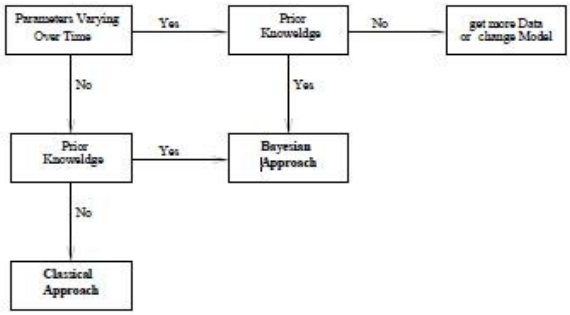

Fig.1.Classical Approach

An Estimation of Software Defects Model is a new approach proposed. The input is the defect data set (as date and defect description) and error correction rate. The error correction rate is various as per the experience of the tester. This approach is fully automated.

III. Estimation Theory Overview

In defect estimation we need to extract information from observations that have been corrupted by noise. In formation of interest to extract from the observations is total number of defects in software. The concepts of the data, noise, and information to extract are represented mathematically; the observations are an N-point data set represented as vector x, or x[0] x[1] . . . x[N-1] and information to extract from the data set is the scalar parameter θ. The noise is represented as the vector w, or w [0], w[1], . . . , w[N -1]. The parameter θ is linearly related to the observation, i.e., each observation is the sum of the parameter and noise.

$$X[n] = \theta + w[n] \quad (1)$$

The noise components w [0], w [1], . . . w[N -1] are observed at different instants of time; they are assumed to be identically independently distributed (iid) Gaussian random variables. The iid assumption is an established, simplifying assumption, which means the Cross Correlation of noise components is 0. Hence, there is no interference among noise components.

A more general form of the linear model is presented in

The noise components w [0], w [1], . . . w[N -1] are observed at different instants of time; they are assumed to be identically independently distributed (iid) Gaussian random variables. The iid assumption is an established, simplifying assumption, which means the Cross Correlation of noise components is 0. Hence, there is no interference among noise components.

A more general form of the linear model is presented in

$$X = h\,\theta + w \quad (2)$$

Where x N × 1 is the data vector, h N ×1 is the observation matrix, h can contain information such as number of testers, failure intensity rate, number of rediscovered Given a data set of observations x, the problem is to estimate θ. There are several steps needed to define an estimator. The first step is to select the model relating the observations, parameter θ and noise. A Normal, or Gaussian, random variable with mean m and variance σ 2, N[m, σ 2] is often assumed. The equation for the probability density function (PDF) of the Normal distribution is used.

$$f(x)=[1/(2\sigma^2\pi)^{(1/2)}]-[1/2\,\sigma^2(x-m^2)] \quad (3)$$

In estimation problem, a PDF is a function of both data and a value of θ. The result of substituting for x and mean m in (5) with x[n] and θ, respectively, is shown in

$$f(x) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} e^{\left[-\frac{1}{2\sigma^2}\left(x[n]-\theta\right)^2\right]} \quad (4)$$

If probability distribution of data is known, finding an estimator θ ^_is simply finding a function of data which gives estimates of the parameter. Many defect estimators [4, 5, 6, 11, 12, 12, 14] assume probability distribution for data. The variability of estimates determines efficiency of estimator. The higher variance of estimator, less effective (or reliable) is estimates. Hence various estimators can be found for data but one with the lowest variance is the best estimator. Another important characteristic of the estimator is that it must be unbiased

(1) There are various methods available to determine the Lower bound on the variance of the estimators, e.g. [8, 10],

But Lower Bound (CRLB) [7] is easier to determine

ISSN: 2278 – 1323

$$E\left[\frac{\partial lnp(\mathbf{x};\theta)}{\partial\theta}\right]=0, \quad \forall\theta, \tag{5}$$

Where expectation is taken with respect to p(x; θ) then, variance of any unbiased estimator θ must satisfy

$$VAR(\hat{\theta}) \geq \frac{1}{-E\left[\frac{\partial^2 lnp(\mathbf{x};\theta)}{\partial\theta^2}\right]}. \tag{6}$$

An estimator which is unbiased, satisfies the CRLB theorem, and is based on a linear model is called an efficient. Minimum Variance Unbiased (MVU) estimator [7]. In other words it efficiently uses data to find estimates. An estimator whose variance is always minimum when compared to other estimators but is not less than CRLB is simply called MVU estimator. For a linear data model (Eq. 2) it is easier to find the efficient MVU estimator as given by Eq. 5. The efficient MVU estimator and its variance bounded by CRLB are given by Equation 6.

IV. Proposed Approach

The new estimation model proposed in this work. This model is illustrated in Fig. 2. The only input to this model is set of data observations; no additional a priori knowledge is required. The data contains number of defects removed from the software under test; data may be sampled using any given fixed period of time (e.g., daily, weekly, bi-weekly, etc.).

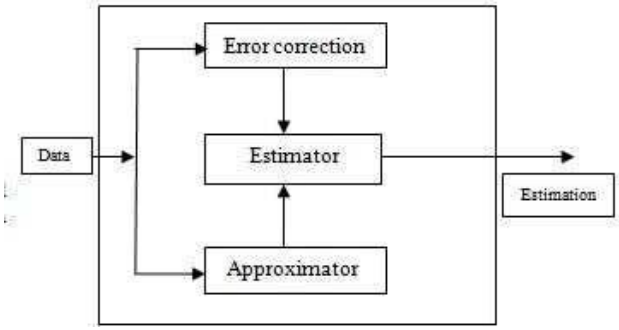

Fig.2. Estimation Model of an Estimator, Approximator and Error Correction.

The output of proposed model is an estimate of total number of defects in software. As additional data become available, the estimate may be recalculated. This model has three main components: the Estimator, Approximator (λ1 λ2), and Error Correction block (refer to Fig. 2). An overview of these components is presented below; Estimator, λ1λ2 Approximator, and Error Correction components. The Estimator component is responsible for calculating an estimate of total number of defects in product. It requires data observations and initial values for two constants, λ1 and λ2, as inputs. The two constants are generated by λ1 λ2Approximator component is responsible for finding two constants, λ1 and λ2 these are rate and scale parameters for defect curve over time. The inputs are data observations. Two existing techniques are applied in this component: exponential peeling and nonlinear regression. The approximation approach is also based on this model for software test process. This component calculates and outputs values for λ1 and λ2.

The Error Correction component is responsible for improving the estimate. The input is estimate ^Rinit calculated by Estimator component. The algorithm in this component calculates a mean growth factor using a history of previous estimates, where each growth factor value is the ratio of one estimate over a previous estimate. The mean growth factor is used to correct current estimate; the corrected value is the output, ^Rinitc .

$$R(t) = R_{init}\left(\frac{\lambda_2}{\lambda_2-\lambda_1}e^{-\lambda_1 t} - \frac{\lambda_1}{\lambda_2-\lambda_1}e^{-\lambda_2 t}\right) \tag{7}$$

where R(t) is number of remaining errors at time t. Rinit is initial number of defects present in software at time t =0. Equation (4) has been rewritten, where R (n) is number of remaining defects at nth time unit

$$R(n) = R_{init}\left(\frac{\lambda_2}{\lambda_2-\lambda_1}e^{-\lambda_1 n} - \frac{\lambda_1}{\lambda_2-\lambda_1}e^{-\lambda_2 n}\right) \tag{8}$$

R (n) is number of remaining defects at day (or any other fixed time interval) n after removing defects at day n - 1. Rinit is the total or initial number of defects present in software product. The values of λ1 and λ2 are calculated by the λ1 λ2Approximator.

To computing the estimator using three unknowns, Rinit, λ1 and λ2, leads to a very complex solution. To avoid this situation, the λ1, λ2 Approximator approximates values of λ1 and λ2. The λ1, λ2 Approximator uses two steps. The first step is to find the initial values λ1 and λ2 using a technique.

Error Correction:

The accuracy of an estimator is heavily dependent on the data set provided. For example, with few initial points in a data set, it is very difficult to produce highly accurate estimates.

ISSN: 2278 – 1323

Survey and Comparative Study

Over the years, many defect prediction studies have been conducted. The studies consider problem using a variety of mathematical models (e.g., Bayesian Networks,

Probability distributions, reliability growth models, etc.) And characteristics of project, such as module size, file structure, etc. A useful survey and critique of these techniques is available in [9].

The classical method is used BBNs to predict number of defects in software. The results shown are possible; the authors also explain causes of results from this model. However, accuracy has been achieved at cost of requiring expert knowledge of Project Manager and historical data (information besides defect data) from past projects. Currently, such information is not always collected in industry. Also, expert knowledge is highly subjective and can be biased.

28.7.2011	Disabled filtered html from configuration/texdt format
29.7.2011	Rich Text Editor Not Visible
01.8.2011	After Clicking Next in one Pager, all pagers getting Changed
03.8.2011	Theme is not working Properly
03.8.2011	Subscribe Option not Working
03.8.2011	Voting Option Not Working
06.8.2011	Unique Visitor and Page Visit Numbers are different than in Database
10.08.2011	XML Sitemap with all nodes excluded as default
13.08.2011	Favicon not working or not showing in all browsers
13.08.2011	Source code Format is Not Appropriate
14.08.2011	Facebook login not working
15.08.2011	Facebook stram breaking the site if not logged in through facebook stream
16.08.2011	Spammers Registration Successful even with CAPTCHA
17.08.2011	SMTP Setting change to send email from another account is breaking the Site
23.08.2011	Image insertion not successful in Inline posts
23.08.2011	Even with 20Mb Dedicated RAM,more than 500Kb Data is not getting uploaded.
24.08.2011	Download Successfull even without registration
26.08.2011	Multiple Registration from Same Email ID is Possible
27.08.2011	Google DOC viewer not opening with the document attachments
28.08.2011	Google AdSense Breaking the View Slide Show
28.08.2011	Registered users are unable to insert image

Fig4.1.Defect Dataset

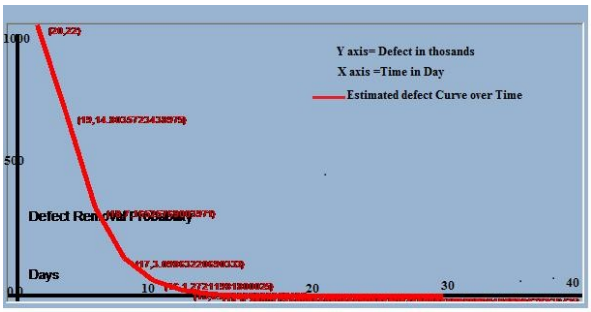

Fig.4.2 Predication using new method

These factors may limit application of such models to a few companies that can cope with these requirements. This has been a key motivating factor in developing this approach. The only information needs is defect data from the ongoing testing process; this is collected by almost all companies fig.4.1. Shows defect data set which contains date of defect occurred and defect type. Fig.4. Shows that estimation of data set(refer fig4.1 data set as input) of 22 defects as input and error correction rate as input is given as 0.5 per day.

Fig4.3. It is similarities of the defects by using bigram similarities. Fig4.4. shows that noise calculation (finding similar defects in dataset i.e. zero) the graph show estimation curve of 22 defects will be removed 14 day.

4.22847939
3.765682
5.17534447
5.59309626
5.23350239
4.94751024
5.06063461
3.86476445
5.95976
4.36723757
5.00445
5.828061
4.81128073
6.266669
5.225881
5.48880434

Fig4.3.Similarities in defect

0
0
0.222222224
0.222222224
0
0
0
0.125
0
0
0.25
0.3529412
0
0.125
0.3529412
0.235294119

Fig.4.4. Noise Calculation

However, they point out that the use of BBN is not always, Possible and an alternative method, Defect Profile Modeling (DPM), is proposed. Although DPM does not demand as much on calibration as BBN[16], it does rely on data from past projects, such as the defect identifier, release sourced, phase sourced, release found, phase found, etc. Both approaches produce accurate estimates with similar convergence rates. Although the results of new approach are very good, there are a number of inherent issues in the approach that restrict its application. New approach not requires more information in addition to the history of number of defects found in the current project.

In summary, the comparison between the application of BBN and the new approach for the data set available in [7] indicates both approaches produce accurate estimates with similar convergence rates. However, as pointed out before, the major advantage of the new approach is that it relies only on defect data, while BBN approach requires much more information to compute the estimates. Such information is not available in many organizations

V. Conclusion

An correct prediction of total variety of software effects helps in analysis of the standing of testing method. however the accuracy of the calculator owes to the estimation technique that is employed to develop the calculator. we\'ve got tried to supply a general framework of obtainable estimation ways.This approach has following characteristics: initial, it uses defect count, associate degree virtually omnipresent input, as solely information needed to reason estimates (historical information don\'t seem to be required). Most firms, if not all, developing package contains a thanks to report defects which may then be simply counted. Second, the user isn\'t needed to supply any initial values for internal parameters or knowledgeable knowledge; it\'s a completely machine-controlled approach. during this paper we have a tendency to mentioned theory behind defect prediction as a product quality part. we have a tendency to conferred some style ideas and meant options for our prediction model.

ISSN: 2278 – 1323

VI. References

[1] Basili and B. Perricone, "Software Errors and Complexity: An Empirical Investigation," Comm. ACM, vol. 27, no. 1, pp. 42-52, 1984.

[2] B.T. Compton and C. Withrow, "Prediction and Control of ADA Software Defects," J. Systems and Software, vol. 12, pp. 199-207, July 1990.

[3] A.E. Ferdinand, "A Theory of System Complexity," Int'l J. General Systems, vol. 1, pp. 19-33, 1974.

[4] L.C. Briand, K.E. Emam, B.G. Freimut, and O. Laitenberger, "A Comprehensive Evaluation of Capture-Recapture Models for Estimating Software Defect Content," IEEE Trans. Software Eng., vol. 26, pp. 518-540, June 2000.

[5] L. C. Briand, K. E. Emam, B. G. Freimut, and O. Laitenberger. A comprehensive evaluation of capture-recapture models for estimating software defect content. IEEE Transactions on Software Engineering, 26(6):518 – 540, June 2000.

[6] E. W. Dijkstra. The humble programmer. Communications of the ACM, 15(10):859– 866,Oct 1972.

[7] S. W. Haider and J. W. Cangussu. Estimating defects based on defect decay model

[8] W. S. Jewell. Bayesian extensions to a basic model of software reliability. IEEE Transactions on Software Engineering, SE-11:1465–1471, 1985.

[9] K. O. John D. Musa. A logarithmic poisson execution time model for software reliability measurement. In Proceedings of the 7th international conference on Software engineering, pages 230–238. IEEE,ACM, IEEE Press, March 1984.

[10] S. M. Kay. Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice Hall PTR, 1993.

[11] S.M. Kendall and A. Stuart. The Advanced Theory of Statistics, volume 2. Macmillan, New York, 1979.

[12] B. Littlewood and J. Verrall. A bayesian reliability growth model for computer software. Journal of the Royal Statistical Society, 22(3):332–336, 1973.

[13] R. J. McaAulay and E. M. Hofstetter. Barankin bounds on parameter estimation. IEEE

Guide. PROF. MR. PRADEEP KUMAR SHARMA , recived his Master of Technology from Sobhasaria Engineering College Sikar , Rajasthan under Rajasthan Technical University,Kota , He is currently working as a Professor in the Department of Computer Engineering, Sobhasaria Group of Institution , Sikar- Rajasthan under Rajasthan Technical University-Kota.

MR. RAGHVENDRA OMPRAKASH SINGH, received his bachelor's degree in Computer Science & Engineering from Solapur University. He is currently Persuing Mtech in Computer Engineering from Rajasthan Technical University and also working as an Assistant Professor with the Department of Computer Engineering, G.M Vedak Institute of technology-Tala ,Mumbai University India. His research interests mainly focused on Software Testing ,Defect Decay Models, and Quality.

MR. YUVRAJ B. HEMBADE, received his bachelor's degree in Computer Science & Engineering from Solapur University. He is currently Persuing Mtech in Computer Engineering from Rajasthan Technical University and also working as an Assistant Professor with the Department of Computer Engineering, Dnyanganga College of Engineering and Research (DCOER)Pune university. His research interests mainly focused on Software Testing

ISSN: 2278 – 1323