Defect Projection Methodologies

S Venu Madhav Chitta,*A01964307*

***Abstract*—A project team always aspires to procreate a quality software product with zero or little defects. High riks components within the software project should be caught as soon as possible, in order to enhance software quality. Software defects always incur cost in terms of quality and time. Moreover, identifying and rectifying defects is one of the most time consuming and expensive software processes. It is not practically possible to eliminate each and every defect but reducing the magnitude of defects and their adverse effect on the projects is achievable. Allocating quality assurance resources wisely is a risky task. If some nondefective module is tested for months and months, this is a waste of resources. If a defective module is not tested enough, a defect might escape into the field, causing a failure and potential subsequent damage. If the defect trends are estimated prior to the release then it helps to reduce the defects produced in the products. Defect-occurrence projection is necessary for the development of methods to mitigate the risks of software defect occurrences. In this paper, various statistical and Machine Learning methods are applied on the software defect datasets and future defect trends are analyzed.**

***Keywords*—Defect Projection, Defect trends, Statistical Distriutions.**

I. Introduction

Defect projection is comparatively a novel research area of software quality engineering.Defects can be defined in a disparate ways but are generally defined as aberration from specifications or ardent expectations which might lead to failures in procedure.Defect data analysis can help us for providing better understanding of the software defect data at large.

The defect occurrence is a user-reported problem that requires developer intervention to correct. This is the observable event of interest for both maintenance and insurance purposes. Defect occurrences not only create problems for software consumers, but also cause problems in maintenance planning for software producers. The costly consequences of defect occurrences have increased interest in insuring software consumers against the associated risks.Defect-occurrence projection is crucial to the development of methods for managing the risks associated with defect occurrences. Accurate defect-occurrence projections can help software maintenance planners to better allocate resources and will be a major step towards novel risk-mitigation techniques for software consumers, such as software insurance.

We are interested the defect-occurrence pattern, which is the rate of defect occurrence as a function of time over the lifetime of a release. We define the lifetime of a release as the duration of time between when a release becomes generally available and when there are no defect occurrences reported to the software development organization for three consecutive time intervals.

Software defect curves describe the behavior of the estimate of the number of remaining software defects as software testing proceeds. The general defect curve trend would be: In the early testing phase, as more and more software defects are detected and removed, the estimate of the number of remaining defects tends to increase. In the middle testing phase, the estimate of the remaining defects remains steady. In the late testing phase, the estimate of the number of remaining defects tends to decrease.

We aim at answering the following research questions:
1. How are defect curves of the same product similar to each other?
2. Which statistical distribution would fit best for the defect curves of Software defect products?
3. What are the prediction accuracy of the defect volume (for each time unit and for the whole reporting time) when using i) only defect curves of prior releases as training data, and ii) only defect curves of other projects?

The organization of the paper is as follows: Section 2 explains about the methodology of the Defect Projection process in the paper. Section 2A discusses about the data preparation process and how the defect curves are related to each other. Section 2B discusses about statistical distributions (Weibull, Gamma and Rayleigh distributions) that would fit best for the defect curves of Software defect products. Section 2C comments about the prediction accuracy of the defect volume (for each time unit and for the whole reporting time) using only defect curves of prior releases as training data, and only defect curves of other projects Section 3 discusses about the results that are generated in the design. Section 4 concludes with the final discussion about the results.

TABLE I: Datasets used in this paper

Product Names	Versions
OpenBSD	V1 - V6
Eclipse	V1 - V6
TomCat	V1 - V4
Eclipse-PDE	V2,V2.1,V3
Eclipse-JDT	V2,V3,V3.1
Eclipse-CDT	V2,V3,V4

II. Methodology

A. Preparation of Datasets and Similarity between the defect curves of the products

To answer the research question Q1 ”How are defect curves of the same product similar to each other?, the Data sets

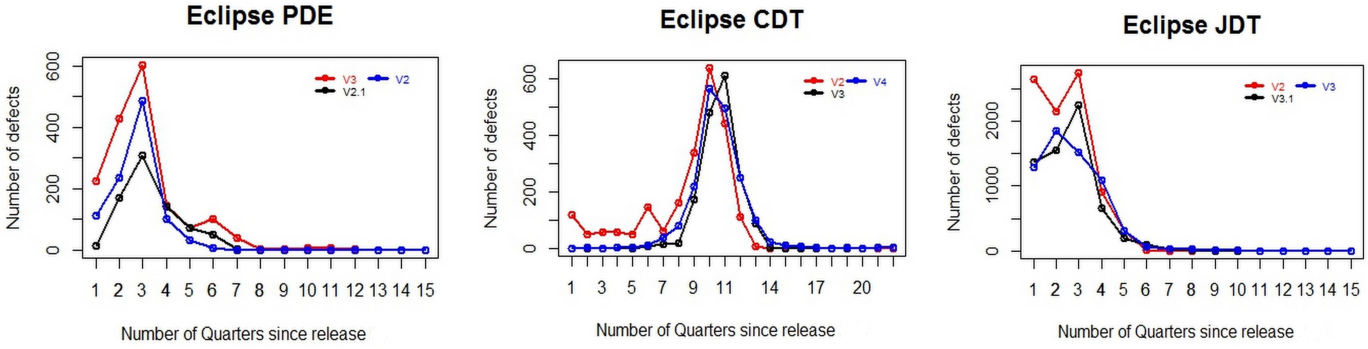

Fig. 1: Plots of versions in the three Eclipse projects

are prepared on the basis of defect curve, i.e. a sequence of numbers of post-release defects per quarter reported for the corresponding version (release) of those software products. The software products for which the datasets are prepared are shown in TABLE I.

We have computed the number of post release defects reported per quarter for each version of the products in TABLE I. The raw data consists of the productID, version, Date the bug reported. So for each quarter from the date the software relased we would compute the total number of bugs. Most of the times the defect curves would be same for each version in a product. The various versions of a product are plotted in Fig.1.

The shape would be multiple-trapezoidal-like curve most of the times. It can be justified from the fact that under certain circumstances, the estimate of the initial number of software defects tends to increase as software testing proceeds.

B. Distributions that are best fit for the defect curves

To answer the research question Q2 " Which statistical distribution would fit best for the defect curves of Software defect products?", the datasets are analyzed for the best fit between three statistical distributions (Weibull distribution, Gamma Distribution and Reyleigh Distributions).

We are interested in the defect-occurrence pattern. Therefore, we are interested in models that model the number of defect occurrences in each time period over the lifetime of a release. We consider the Reileigh model, the Gamma model, and the Weibull model. These models are promising because prior research in software reliability engineering has shown each model to be effective at modeling defect-occurrence patterns at a software development organization [1][2][3].The distribution fits the curve to the shape and scale parameters.

The different distribution models and Various parameters are: *Weibull distribution:* The probability density function of a Weibull random variable is

$$f(x;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0, \\ 0 & x < 0 \end{cases}$$

where k >0 is the shape parameter and lambda >0 is the scale parameter of the distribution.

Rayleigh Distribution: A Rayleigh distribution is often observed when the overall magnitude of a vector is related to its directional components. The probability density function of the Rayleigh distribution is

$$f(x;\sigma) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)}, \qquad x \geq 0,$$

$$where\, \sigma > 0\, is\, the\, scale\, parameter\, of\, the\, distribution$$

AIC score: The Akaike information criterion (AIC) is a measure of the relative quality of a statistical model, for a given set of dataset.AIC deals with the trade-off between the goodness of fit of the model and the complexity of the model. It is founded on information entropy: it offers a relative estimate of the information lost when a given model is used to represent the process that generates the data. AIC does not provide a test of a model in the sense of testing a null hypothesis; i.e. AIC can tell nothing about the quality of the model in an absolute sense. If all the candidate models fit poorly, AIC will not give any warning of that. In the general case, the AIC is:

$$AIC = 2k - 2ln(L)$$

where k is the number of parameters in the statistical model, and L is the maximized value of the likelihood function for the estimated model.

R^2 *score:* (R^2) indicates how well data points fit a statistical model sometimes simply a line or curve. It is a statistic used in the context of statistical models whose main purpose is either the prediction of future outcomes or the testing of hypotheses, on the basis of other related information. It provides a measure of how well observed outcomes are replicated by the model, as the proportion of total variation of outcomes explained by the model.

The Weibull model can account for different increasing and decreasing trends, which reflect initial increases and eventual decreases in defect occurrences. We generally expect early defect occurrences to show a primarily increasing trend as users migrate to the release, exercise the software, and report

defects, and we expect later defect occurrences to show a primarily decreasing trend as the rate of adoption declines and the release becomes more reliable due to defect removal.

We used Non-Linear Least Squares method to find out the best fit of the defect curve to the distribution. The fitness is given by R^2 and AIC values. R^2 is the similarity between the two curves. So higher the value of R^2 higher the fit and lower the AIC values highest is the fit.

C. Prediction of Defect curves

To answer the research question Q3 "the prediction accuracy of the defect volume (for each time unit and for the whole reporting time) when using only defect curves of prior releases as training data, and only defect curves of other projects", the defect curves of the future versions are predicted from the present versions.

To predict the future versions from the present versions the shape and scale parametersof the distribution that fits best to the curve from previous section is taken. This parameters are then applied to the latter versions by applying the shape and scale parameters of the previous versions to the versions that are to be predicted and then prediction accuracy is then determined. The conclusions would be drawn from all the observations.

To predict the future versions from the other projects we would use the shape and scale parameters from other projects. Regression models like Linear Regression is used to make the estimates. The prediction accuracy of these results is determined by the Mean Absolute Error values and various conclusions are drawn.

In the next section the various results are analysed and discussed.

III. Results and Discussion

Similarity between the Defect Curves: In Fig. 1 we observe that the curves are similar to each other. The shape of the curves for different versions in the product are the same. This property would be helping us to predict the defect curves of the future versions thereby helping any software organization to reduce the number of defects.

The percentage of similarity between various versions of a product are found out. In Fig. 2, the versions whose percentage is around 85% is found out and is plotted. In following sections, the comments about these versions is made. We can observe that each product has atleast one combination of the versions whose percentage is Greater than 85%.

Distribution that best fits to the Defect curve: The three statistical distributions (Weibull, Gamma, Reyleigh) are used to estimate the defect curve. For each distribution, there are shape and scale parameters which we need to modify to estimate. The best fit is found by the R^2 and AIC values. These values are calculated in TABLE II. The bolded estimates are the best fit for the curve. We can observe from the table that the best distribution about most of the times is Weibull distribution model since R^2 is low for most of the versions in the products. Therefore we use Weibull distribution in next section to predict future versions.

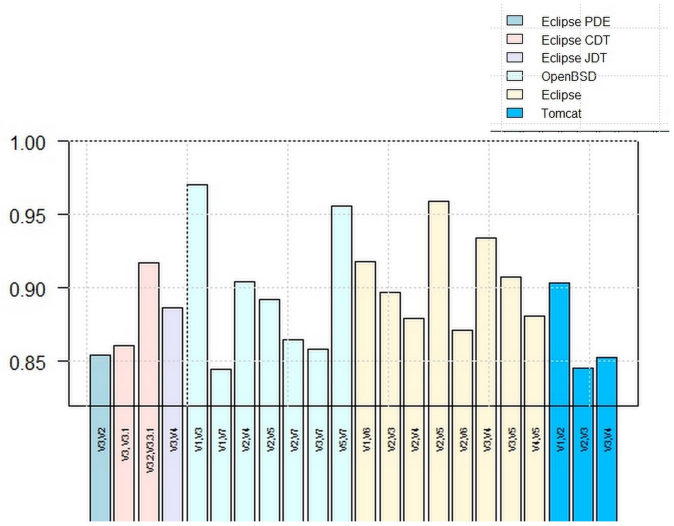

Fig. 2: Similarity between the defect curves

Prediction accuracy of the defect volume when using defect curves of prior releases as training data: For the research question Q3A "What is the Prediction accuracy of the defect volume when using defect curves of prior releases as training data", the shape and scale values of present version are used to estimate the future versions of the product and the Relative error is found out. So smaller the relative error , higher is the prediction accuracy of the future versions.

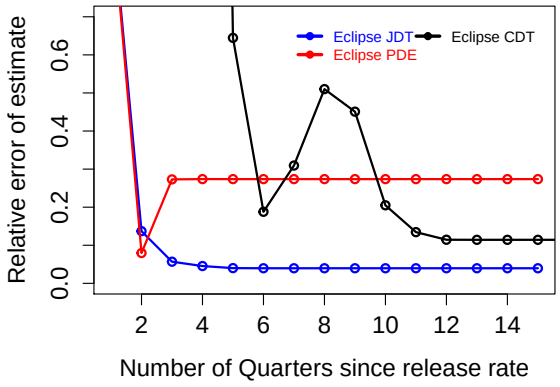

Fig. 3: Prediction accuracy of the future versions

In Fig.3, the Number of quarters of the future versions which are to be required to predict the furthur quarters of the future versions can be observed. The following conclusions can be drawn from the figure:

(i) As the number of quarters increases, the prediction accuracy of that version increases. Also we cannot predict the future versions without observing some defect values of that version.

TABLE II: Best distribution models that fit the curves

		R^2			AIC		
Product	Version	Weibull	Gamma	Rayleigh	Weibull	Gamma	Rayleigh
Eclipse JDT	V 3.2	***0.92***	0.89	0.91	***208.74***	214.1	209.29
	V 2	***0.93***	0.92	0.92	***217.77***	220.76	217.06
	V 3	***0.99***	0.98	0.99	***169.51***	181.69	169.71
Eclipse PDE	V 2.1	0.95	***0.98***	0.78	141.44	***129.75***	160.23
	V 2	***0.97***	0.95	0.73	***146.48***	152.28	175.09
	V 3	***0.94***	0.9	0.85	***165.8***	171.75	174.73
Eclipse CDT	V 2	***0.94***	0.92	0.3	***239.34***	245.05	284.84
	V 3	***0.98***	0.24	0.27	***133.19***	185.69	182.75
	V 4	0.97	***0.99***	0.25	218.18	***193.57***	285

TABLE III: Similarity between the versions

Data set	Present Version	Future Version	Similarity
JDT	V2	V3	0.917
PDE	V2	V3	0.854
CDT	V2	V4	0.713

(ii) The prediction accuracy depends on the similarity between the curves. This can be justified by the following table. We can observe from TABLE III that when the similarity between the curves is high. When the similarity is low then we need more quarters to predict the curve efficiently.

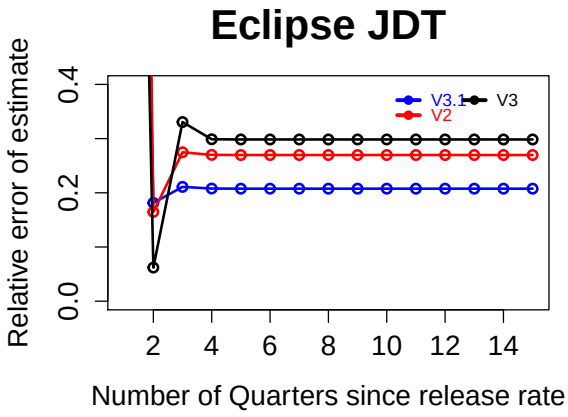

Fig. 4: Prediction accuracy of the Products from other products

Prediction accuracy of the defect volume when using defect curves of other projects: For the research question Q3A "What is the Prediction accuracy of the defect volume when using defect curves ofprevious versions as training data", we estimated the Eclipse JDT version using other Datasets. The linear regression model is applied and the estimates that are choosen are Software type, Cycle and the Cost of the License. We predicted the shape and scale parameters of the Eclipse JDT and then these parameters are then used in the Weibull distribution, which is the best distribution model. The relative error of the Eclipse JDT is shown in Fig.4.

The Fig. 4 is the plot between the relative error of all the versions in the product Eclipse JDT. The relative error is around 0.25 for all the 3 versions. When the nuber of quarters increases the prediction accuracy increases.The accuracy become linear after some quarters in all the 3 versions,Thus we can say that using the other products as the training data we can predict the defect curves of the future versions of products.

IV. Conclusion and Future Work

This paper discusses about the Defect Projection model which uses the defect curves of various datasets such as Eclipse, Tomcat, Open BSD . The similarity between the curves of various versions in a product is determined The curves are then fit into a statistical distribution (like Rayleigh, Weibull, Gamma models) and the best fit to the curve is determined. Then using the shape and scale parameters of the best fit, we can predict the defect curves of the future versions using the previous version trends. Also we can predict the product's curves using other products curves. Though the prediction accuracy from other products is not as good as from other versions we can predict to some extent.

The future work includes using better regression model to estimate other products trends. The data model which fits curve can be estimated from some other distribution than the three models discussed in this paper.

V. References

1.Tim Klinger P. Santhanam Tung Thanh Nguyen, Evelyn Duesterwald and Tien N. Nguyen. Characterizing defect trends in software support. June 2012.

2. A. Mockus, D. Weiss, and P. Zhang. Understanding and predicting effort in software projects. In ICSE, p274-284, 2003.

3. Li, Paul Luo, Mary Shaw, Jim Herbsleb, Bonnie Ray, and Peter Santhanam. Empirical evaluation of defect projection models for widely-deployed production software systems. Vol. 29, no. 6. ACM, 2004.