International Journal of Reliability, Risk and Safety: Theory and Application

Online ISSN: 2676-3346

OPEN ACCESS

Vol. 7/ Issue 1/ Year 2024/ pp. 59-72
DOI: 10.22034/IJRRS.2024.7.1.8
Received: 28 February 2024, Revised: 12 May 2024, Accepted: 25 May 2024
Available online at: https://www.ijrrs.com

Original Research Article

Software Defect Discovery and Resolution Modeling incorporating Severity

Maskura Nafreen[1], Ying Shi[2] and Lance Fiondella[1*]

1- Department of Electrical and Computer Engineering, University of Massachusetts Dartmouth, North Dartmouth, Massachusetts, USA

2- Mission Software and Ground System Assurance Branch, NASA Goddard Space Flight Center, Greenbelt, Maryland, USA

*** lfiondella@umassd.edu**

Abstract

Traditional software reliability growth models only consider defect discovery data, yet the primary concern of software engineers is defect removal. Past attempts to model defect resolution emphasize approaches based on differential equations and queueing theory. However, these models do not explicitly identify the activities performed to remove defects and resources allocated to these activities according to their severity. Models should consider these practical factors to enable more detailed resource allocation and planning.

This paper presents a model to predict the number of defects resolved according to the discrete Cox proportional hazard model with covariates, demonstrating the approach with covariates on the number of low, medium, and high severity defects that were discovered but not resolved in successive intervals. A comparison with differential equation-based and distributional approaches reveals that the covariate model performs better on each goodness of fit measure considered and requires less time to apply. The covariate model also better tracks unresolved defects and exhibits low predictive error, even when as little as 10-20% of testing has been completed. These results suggest that collecting information on defect resolution activities and the corresponding effort dedicated could substantially improve defect resolution modeling to guide process improvement.

Keywords: Software defect resolution; Software defect severity; Software defect tracking; Software reliability; Software reliability modeling.

Nomenclature

Acronyms

AIC	Akaike information criterion
BIC	Bayesian information criterion
GEV	Generalized extreme value
LL	Log-likelihood function
MLE	Maximum-likelihood estimation
MTTR	Mean time to resolution
MVF	Mean value function
NASA	National Aeronautics and Space Administration
NHPP	Non-homogeneous Poisson process
PSSE	Predictive sum of squares error
SRGM	Software reliability growth model
SSE	Sum of squares error

Notation

$N(t)$	Number of defects detected by time t
$N_r(t)$	Number of defects resolved by time t
$m(t)$	Mean number of defects detected by time t
$m_r(t)$	Mean number of defects resolved by time t
$m_r^b(t)$	Mean number of defects resolved by time t assuming common discovery and resolution rate b
$\lambda(t)$	Defect discovery rate at time t
$\lambda_r(t)$	Defect resolution rate at time t
$\lambda_r^b(t)$	Defect resolution rate at time t assuming common discovery and resolution rate b

How to cite this article:
M. Nafreen, Y. Shi and L. Fiondella, "Software Defect Discovery and Resolution Modeling incorporating Severity," *International Journal of Reliability, Risk and Safety: Theory and Application*, vol. 7, no. 1, pp. 59-72, 2024, doi: 10.22034/IJRRS.2024.7.1.8

$\mathcal{D}$	Set of defects discovered
$\mathcal{R}$	Set of defects resolved
t_i	Time at which i^{th} defect was discovered
t_i^r	Time at which i^{th} defect was resolved
$t_{(i)}$	Time between discovery and resolution of i^{th} defect
$E[\hat{T}_r]$	Mean time to defect resolution
n	Total number of defects or intervals
w	Interval width
m	Number of covariates
$\mathbf{x}_{n\times m}$	Matrix of observed covariates
$\mathbf{x}_{i\times j}$	Effort dedicated to activity j in interval i
y_i	Number of defects resolved in interval i
y_i^s	Number of severity s defects resolved in interval i
p_{i,x_i}	Probability defect is resolved in interval i
$h(\cdot)$	Baseline hazard function
β_m	Vector of covariate coefficients

1. Introduction

Software reliability growth models (SRGM) to characterize the defect discovery process during testing have been the subject of study since the early 1970s [1]. While models to describe defect resolution were proposed as early as 1975 [2], defect resolution models [3] only began to receive consistent attention after 2000. Metrics-based models to characterize the number of defects detected in successive intervals [4] as a function of test activities is another topic that has recently enjoyed more thorough study. Since defect resolution, not discovery, is the true source of increased software reliability, defect resolution models to explicitly consider the activities performed to resolve defects of different severities are needed to allocate effort during software defect tracking.

Most past papers are based on NHPP SRGM, which we call defect discovery models. In contrast, the most common class of defect resolution models is based on the integrated defect discovery and resolution process modeling framework [3]. Queueing theoretic models of discovery and resolution [5,6] have also been proposed, including multi-priority queuing models [7] for the mean time to resolution according to defect severity, but were limited to homogeneous rates for fielded software and did not consider time-varying rates characteristic of a software testing process over a defined test schedule. Before this, most defect severity models were limited to NHPP SRGM for defect discovery composed of separate mean value functions [8] as well as homogeneous [9] and heterogeneous [10] mixtures of mean value functions. Recent research has demonstrated that metrics-based models [4], [11], also known as covariate models, characterize the defect discovery process in terms of the underlying test activities very well, suggesting covariate models may also effectively characterize the defect resolution process.

This paper presents a software defect resolution model based on the discrete Cox proportional hazard model with covariates and compares it with the software defect resolution models that performed best in a recent study connecting SRGM to defect tracking databases [12], including (i) an integrated defect discovery and resolution process model and (ii) a distributional approach that shifts the defect discovery model by the mean time to defect resolution. Novel models were enabled by a NASA software defect tracking data set [13], including the times when defects were discovered and resolved. This previous study concluded that more complex integrated defect discovery and resolution process models did not characterize the defect resolution process better. It was also observed that a semi-Markov process model of the defect tracking lifecycle did not significantly improve estimates of the mean time to defect resolution. Therefore, while the semi-Markov process model more closely represented the defect tracking process, it only attained marginal improvements in prediction because of sample size and data quality issues. Hence, it was concluded that better data collection practices could improve the utility of the semi-Markov process model. Despite the lack of disciplined data collection practices by software practitioners, it is necessary to create models that predict robustly and are not computationally intensive. Toward this end, the software defect resolution model based on the discrete Cox proportional hazard model with covariates can rely solely on information on open defects (discovered but not yet resolved) of different severity to predict the total number of defects resolved as well as more fine-grained predictions of the number of defects of a specific severity resolved.

Based on the observations above, this paper seeks to enhance the utility of covariate models and encourage their adoption, making the following primary contributions:

- A software defect resolution model incorporating covariates based on a discrete Cox proportional hazard rate to predict the removal of defects by severity in terms of activities or metrics associated with defect removal
- A method to assess the predictive ability of alternative models in terms of the number of open defects by severity

Our results indicate that the defect resolution model incorporating covariates outperformed differential equation-based and distributional approaches by order of magnitude on all measures of goodness of fit and required less runtime to apply. The covariate defect resolution model also achieved compelling visual fits to the number of open defects and achieved greater predictive accuracy earlier in the testing schedule. Thus, while covariates for the activities performed to resolve defects and the effort dedicated to these activities are not needed, they could

certainly improve the accuracy of models and promote process assessment and improvement.

The remainder of the paper is organized as follows: Section 2 summarizes related research. Section 3 describes defect discovery and resolution models. Section 4 describes model assessment techniques. Section 5 provides illustrations comparing the predictive accuracy and performance of alternative models. Section 6 offers conclusions and identifies future research.

2. Related Research

This section summarizes past research on defect discovery and resolution models for software. A chronology of contributions spanning the past 45 years is provided. Related developments are grouped logically, wherever possible. Models explicitly drawing upon queueing theory are also discussed. Since covariate SRGM are employed to characterize defect resolution for the first time in this paper, we also review key developments related to this class of models. The section concludes with a statement of the paper's novel contributions.

Early studies that sought to characterize software defect discovery and resolution include Schneidewind's [2] defect discovery model based on a discrete exponential mean value function and time lag resolution. Xie and Zhao [14] extended this model by assuming the defect resolution rate is proportional to the number of unresolved defects, demonstrating the Poisson thinning process could model the difference between unresolved defects. The inflection S-shaped model was proposed by Ohba [15] to describe scenarios where some defects needed to be resolved before others could be reached. In contrast, the delayed S-shaped model was developed by Yamada et al. [16] by incorporating a time delay to model this dependence. Kapur and Younes [17] modeled leading and dependent defects. Two SRGM with imperfect debugging were proposed by Yamada et al. [18], where new defects could be introduced when other defects were resolved. A non-homogeneous continuous-time Markov chain was employed by Gokhale et al. [19] to model defect repair and analyze the impact of fault removal policies on the number of defects remaining when testing was completed. More recent studies include Huang et al. [20], who showed that applying a time-dependent delay function can derive several existing SRGM. An integrated defect discovery and resolution process modeling framework was proposed by Lo and Huang [3], where the defect resolution process was expressed in terms of a time-varying resolution intensity as well as the difference between the number of defects discovered and resolved. Ullah et al. [21] conducted a comparative analysis of SRGM on discovery and resolution data sets from several dozen open-source software projects. Liu et al. [22] proposed an approach to estimate the parameters of a defect removal model for semi-grouped data consisting of the approximate times at which defects were discovered and resolved, while Yang et al. [23] modeled defect detection and correction of a multi-release open-source software and related optimal release problems. Cinque et al. [24] proposed an NHPP SRGM for debugging data documented in a bug-tracking system with defects of varying severity, improving the accuracy of predictions when debugging activities did not follow the modeling assumptions closely. Vizarreta et al. [25] found the inflection S-shaped model characterized the defect resolution of four successive releases of an open-source software-defined network controller well and subsequently [26] fit cumulative distribution functions for the time to resolution by the severity of defects. Xie et al. [27] proposed a defect resolution model in which defect resolution times are allowed to follow a variety of common distributions.

Applications of queueing theory to software defect discovery and resolution include the work of Dohi et al. [5], who proposed a model of software failure occurrences as an M/G/8 infinite server queue, unifying previously proposed software reliability growth models, including general order statistics models [28]. Dohi et al. [6] subsequently presented Bayesian estimation techniques for their infinite server queueing model and demonstrated improved goodness of fit over general order statistics models. Gokhale and Mullen [7] developed multi-priority queuing models for the software defect resolution process, considering the effect of queuing system structures, priority levels, and priority disciplines on the time to resolve defects of different severities. Lin et al. [29] implemented simulation procedures for G/G/8 and G/G/m infinite server queues for software defect detection and removal. Huang and Huang [30] showed how to incorporate finite and infinite server queueing models into software reliability modeling for defect detection and removal, assuming perfect and imperfect debugging. Zhang et al. [31] incorporated testing effort functions into finite server queueing defect detection and removal models. Kapur et al. [32] developed an M*/G/8 infinite server queue, where the mean time between defect discovery and removal varies according to defect severity. Later, Huang and Kuo [33] proposed an extended finite server M/M/c queueing model to address limited testing resources. Tokuno et al. [34] developed performability measures for models based on infinite server queuing to quantify the capacity of a software process to complete tasks within a time limit. Okamura and Dohi [35] proposed a generalized bivariate fault detection and correction process, a model with hyper-Erlang distributions, and expectation maximization algorithms to estimate model parameters.

Early covariate models for software reliability include the work of Khoshgoftaar et al. [36-38], who applied alternative estimation techniques for applied nonlinear regression with software metrics [39] as explanatory variables to predict the number of faults in a program module. Evanco and Lacovara [40] presented multiple regression models based on ordinary least

squares regression, Poisson, binomial, ordered response, and proportional hazards models for software reliability and subsequently [41] integrated the Poisson regression into a modified form of the Goel- Okumoto model [42]. Cid and Achcar [43] presented a Bayesian approach to the superposition of several independent NHPP in the presence of covariates. Ray et al. [44] developed a software reliability model for covariates based on hierarchical Bayesian methods. Gandy and Jensen [45] proposed a non-parametric software reliability model based on a multivariate counting process with additive intensity, incorporating covariates for open-source software composed of multiple sub-projects. Ishii et al. [46] used a bivariate NHPP SRGM to characterize defect discovery as a function of software testing activities such as calendar time, the number of test cases executed, and test execution time, while Kapur et al. [47] developed a bivariate NHPP SRGM including testing time and coverage metrics according to the Cobb-Douglas production function. Rinsaka et al. [11] combined the proportional hazards model and NHPP to produce a generalized defect detection process enabling a time-dependent covariate structure. Shibata et al. [4] extended this model to a cumulative Bernoulli trial process. Okamura et al. [48,49] proposed an SRGM for multiple test metrics based on logistic regression and a parameter estimation method based on logistic regression and the expectation-maximization algorithm. The logit and Cox proportional hazards [51] covariate models were generalized by Kuwa and Dohi [50]. Okamura and Dohi [52] extended a covariate model integrating Poisson regression and the non-homogeneous Poisson process to accommodate time series data on defect detection and software metrics of multiple modules. A unified model combining NHPP SRGM and generalized linear models was proposed by Okamura and Dohi [53] to encompass several common SRGM and the logistic and Poisson regression covariate SRGM. Wiper et al. [54] presented a neural network regression method incorporating covariates composed of software metrics to estimate inter-failure times or the number of failures with a Bayesian approach. Torrado et al. [55] developed a semi-parametric Bayesian model incorporating Gaussian processes to estimate software failures in successive time intervals, assuming that the software updates are performed after each interval and that information on software metrics is available. Nagaraju et al. [56] presented NHPP SRGM incorporating covariates based on the discrete Cox proportional hazards model and formulated the optimal test activity allocation problem to maximize defect discovery.

In contrast to past studies, this paper is the first to apply a covariate model to software defect resolution. The predictive and computational performance is compared with (i) the most common alternative, namely an integrated defect discovery and resolution process model [3], and (ii) a distributional approach [12], which demonstrated better predictive and computational performance than the integrated defect discovery and resolution process modeling approach. While queuing models offer additional rigor and possess intuitive appeal, this class of models was not considered because the data did not exhibit statistical evidence that either a priority or mixed queueing discipline was employed by the project to handle defects of low, medium, and high severity. Moreover, it was not possible to speak with members of the original project team to understand the queuing discipline employed during project execution or other relevant factors to construct a satisfactory model. Nevertheless, this paper is one of the few considering defect resolution by severity. It is also the first to apply covariate models to the resolution of defects of each severity level.

3. Software Defect Discovery and Resolution: Models Incorporating Severity

This section describes three methods to model the software defect resolution process, including (i) the integrated defect discovery and resolution process modeling framework of Lo and Huang [3], (ii) the distributional approach developed by Nafreen et al. [12], and (iii) the NHPP SRGM incorporating covariates based on the discrete Cox proportional hazards model of Nagaraju et al. [56], which is adapted to characterize defect resolution. Each model requires a pair of time series indicating the time defects were discovered and resolved, which can be unambiguously determined from a defect tracking database. Defect resolution models enable insights for software practitioners that can be made during the software testing process, including the average amount of time required to remove all defects of a specified severity discovered up to (i) the present time t and (ii) present time t as well as all additional defects anticipated to be discovered.

3.1 Integrated defect discovery and resolution processes

The integrated defect discovery and resolution processes modeling framework [3] expresses the defect discovery and resolution rates as a system of differential equations possessing the below form

$$\frac{dm(t)}{dt} = \lambda(t)\big(\omega - m(t)\big) \tag{1}$$

$$\frac{dm_r(t)}{dt} = \lambda_r(t)\big(m(t) - m_r(t)\big) \tag{2}$$

where $m(t)$ ($m_r(t)$) denotes the mean value function of the number of defects discovered (resolved) by time t, $\lambda(t)$ ($\lambda_r(t)$) the defect discovery (resolution) rate, and $\omega > 0$ is the number of defects that would be discovered and resolved with indefinite testing. Therefore, Equation (2) expresses the instantaneous rate of change in defect

resolution as the product of the defect resolution intensity multiplied by the number of defects discovered but unresolved at time t.

Non-homogeneous Poisson process models were applied to the defect discovery data because failure times were the only information recorded in the defect tracking database, not the testing effort or underlying activities. Analysis of the NASA data set [13] employed in this study found that among the Jelinski-Moranda [1], inflection S-shaped [15], Yamada Delayed S-shaped [16], Goel-Okumoto [42], Weibull [57], and Geometric model [58], the inflection S-shaped model characterized the overall defect discovery process and the discovery process of high, medium, and low severity defects best.

Thus, the mean value function for defect discovery is

$$m(t) = \omega \frac{1-e^{-bt}}{1+ce^{-bt}} \quad (3)$$

where b is a constant defect discovery rate, c the inflection

$$c = \frac{1-r}{r}, \quad r \in (0,1] \quad (4)$$

and r is the inflection rate.

Assuming defect resolution intensity $\lambda_r(t) = b$, identical to parameter b of Equation (3), solution of Equation (2) produces the MVF of the defect resolution process

$$m_r^b(t) = \omega\left(1-e^{-bt} + (1+c)\log\left(\frac{1+c}{c+e^{bt}}\right)e^{-bt}\right) \quad (5)$$

The log-likelihood function of defect resolution times data is

$$LL(\theta,\omega) = -m_r^b(t_n^r + x) + \sum_{i=1}^{n}\log\left(\lambda_r^b(t_i^r)\right) \quad (6)$$

where $m_r^b(t_n^r + x)$ is the MVF of the defect resolution process evaluated at the time at which the last defect was resolved (t_n^r) plus any additional time (x) [59] since this most recent defect was resolved, and

$$\lambda_r^b(t) = \frac{dm_r^b(t)}{dt} \quad (7)$$

The defect resolution process is fit by substituting Equations (5) and (7) as well as the resolution data directly into Equation (6) and maximizing.

3.2 Distributional approach

The distributional approach [12] fits the time distribution between discovery and resolution of each defect. Online application of the distributional approach uses defect discovery and resolution data available up to time t. Since defect resolution is not immediate, the number of defects discovered may be strictly greater than the number of defects resolved. Maximum likelihood estimation techniques for censored data are employed with the following likelihood function to enable an unbiased estimate of the time to resolve defects.

$$Lik(\Theta \mid T) = \prod_{i\in\mathcal{R}}^{k} f\left(t_{(i)};\theta\right) \times \prod_{i\in\mathcal{D}}^{n-k} 1 - F(T-t_i;\theta) \quad (8)$$

where k is the number of defects in the set $\mathcal{R}$ that were discovered and resolved before time T, $(n-k)$ the number of defects in set $\mathcal{D}$ discovered by time T but unresolved, and θ the parameters of the distribution being fit to the data. Moreover, $t_{(i)}$ denotes the time between discovery t_i and resolution t_i^r of the i^{th} defect so that $t_{(i)} = t_i^r - t_i$, whereas $1 - F(T-t_i)$ is the probability that a defect discovered at time t_i is still unresolved at time T.

Given defect discovery and resolution data up until time t, Equation (8) is maximized with multiple alternative distributions and the one exhibiting the best fit is used to make predictions. The mean of this distribution is interpreted as the mean time to resolution (MTTR). Therefore, the MVF of the number of defects resolved by time t may be expressed as

$$m_r(t) = m(t - E[\hat{T}_r]) \quad (9)$$

which shifts the mean value function of the defect discovery process to the right by the mean time to resolve defects ($E[\hat{T}_r]$).

3.3 Covariate approach

The discrete Cox proportional hazard NHPP SRGM [56] links m covariates to the number of events in each of n successive intervals. In modeling defect resolution, these covariates can be (i) defect resolution activities or (ii) metrics related to defect resolution, such as the number of defects of each severity unresolved by the end of interval i. The covariates are denoted $\mathbf{x}_{n\times m}$, where $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$ corresponds to the duration each activity was performed or the value of the metrics in interval i.

The MVF of defects resolved through interval n is

$$m_r(\mathbf{x}) = \omega \sum_{i=1}^{n} p_{i,\mathbf{x}_i} \quad (10)$$

where the probability a defect is resolved in interval i, after going unresolved in the first $(i-1)$ intervals is as:

$$p_{i,\mathbf{x}_i} = \left(1 - \left(1-h(i)\right)^{g(\mathbf{x}_i)} \times \prod_{k=1}^{i-1}\left(1 - h(k)\right)^{g(\mathbf{x}_k)}\right. \quad (11)$$

$h(\cdot)$ is baseline hazard function, and

$$g(\mathbf{x}_i;\beta) = \exp(\beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_m x_{im}) \quad (12)$$

Three examples of hazard functions are the Geometric: $\lambda(k) = b$, Negative binomial of order two: $\lambda(k) = \frac{kb^2}{1+b(k-1)}$, and Discrete Weibull of order two: $\lambda(k) = 1 - b^{k^2-(k-1)^2}$, where $b \in (0,1)$ in all three cases.

The discrete Cox proportional hazard NHPP SRGM possesses the following log-likelihood function:

$$LL(\theta,\beta,\omega) = -\omega\sum_{i=1}^{n} p_{i,\mathbf{x}_i} + \sum_{i=1}^{n} y_i \ln(\omega) + \sum_{i=1}^{n} y_i \ln\left(p_{i,\mathbf{x}_i}\right) - \sum_{i=1}^{n} \ln(y_i!) \quad (13)$$

where $\mathbf{y}_n$ is the number of defects resolved in each of the n intervals and y_i is the number of defects resolved in interval i. Given covariates $\mathbf{x}$ and defect resolution vector $\mathbf{y}$, model fitting identifies the numerical estimates $\widehat{\omega}$ (the total number of defects to be resolved), $\widehat{\beta}$ (vector of covariate coefficients), and $\hat{b}$ (hazard function parameter).

4. Model Assessment

Model assessment quantifies how well a model performs on a data set. This section provides a self-contained summary of complementary measures, including the sum of squares error [60] and the predictive sum of squares error [61], as well as the Akaike information criterion [62] and Bayesian information criterion [63]. Lower values of these measures are preferred. Ideally, a single model performs best on all measures. However, this rarely occurs in practice. If no single model outperforms all of its competitors on all measures model selection becomes subjective based on expert experience and factors such as the amount of data available, testing stage, and predictive horizon.

4.1 Sum of squares error (SSE)

SSE measures the disagreement between the empirical defect resolution time data and estimates of a defect resolution model.

$$SSE = \sum_{i=1}^{n}\left(N_r(i) - \widehat{m}(i)\right)^2 \quad (14)$$

where $N_r(i)$ is the number of defects resolved by time t_i or the end of interval i and $\widehat{m}_r(i)$ is the model estimate of the number of defects resolved.

4.2 Predictive sum of squares error (PSSE)

PSSE fits a model to the first $k < n$ failure times or intervals and computes the disagreement between empirical data and model estimates on the remaining $n - k$ times or intervals not used to perform model fitting.

$$PSSE = \sum_{i=k+1}^{n}\left(N_r(i) - \widehat{m}(i)\right)^2 \quad (15)$$

SSE is the special case where $k = 0$.

4.3 Akaike Information Criterion (AIC)

The Akaike Information Criterion is an information-theoretic measure of goodness of- fit, which is based on the concept of entropy, measuring the information lost when a model is applied. The AIC quantifies the tradeoff between a model's complexity and characterization of the observed data. The AIC of model i is a function of the number of model parameters (p) and maximized log-likelihood.

$$AIC_i = 2p - 2LL(\widehat{\Theta} \mid T) \quad (16)$$

4.4 Bayesian information criterion (BIC)

BIC is similar to AIC, but the penalty term also includes the sample size (n). The BIC of model i is

$$BIC_i = p\log(n) - 2LL(\widehat{\Theta} \mid T) \quad (17)$$

5. Illustrations

This section compares the alternative defect discovery and resolution models described in Section 3 according to the model assessment techniques given in Section 4 and their computational performance. Section 5.1 describes the data extracted from a NASA defect tracking database to which the models were applied. Section 5.2 conducts a retrospective analysis common in historical defect discovery modeling papers using all available data. Next, we describe the steps of the distributional approach to determine the mean time to resolution, and the covariates and interval width were selected for the covariate approach. Tradeoffs between the goodness of fit and runtime posed by the interval width of the covariate approach are also examined. Section 5.3 performs a novel analysis based on the defect and resolution discovery processes, namely an analysis of the open defects for progress tracking throughout the testing process. Section 5.4 assesses predictive accuracy. Each analysis is performed on the entire dataset consisting of defects of all three severities and analyses broken down by low, medium, and high severity.

5.1 Data Description

The dataset [13] considered was extracted from a defect-tracking database employed by NASA on a major project spanning multiple years. After cleaning, the database consisted of $n = 455$ rows, each of which corresponded to a defect, including low ($n_3 = 61$), medium ($n_2 = 381$), and high severity ($n_1 = 13$) defects. The database also documented when the defect entered any of the 13 possible states from discovery to resolution. However, a recent study [12] demonstrated that the sample size was too small to apply a semi-Markov process model of the defect tracking process. Therefore, for this study, we created two-time series, one composed of the time each defect was discovered and a second for the time each defect was resolved. Unlike past defect discovery and resolution modeling efforts, the information contained in rows of the database also allowed us to compute the individual times between resolution and discovery, enabling the application of the distributional and covariate approaches.

5.2 Retrospective Analysis

Figure 1 shows $N(t)$ and $N_r(t)$ (the discovery and resolution counting processes) and the corresponding

discovery and resolution model fit the three approaches described in Section 3.

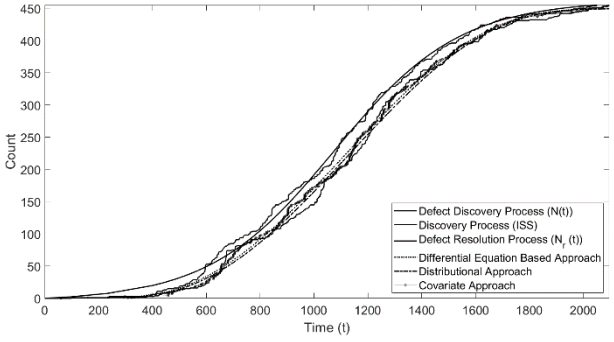


Figure 1. Defect discovery and resolution processes with models fitted to all data

The first counting process (left step function) indicates when the defects were discovered. The fit of the inflection S-shaped model (Equation (3)) is also shown. The second counting process (right step function) indicates when defects were resolved. The fits of the differential equation-based (Equation (5)), distributional (Equation (9)), and covariate model (Equation (10)) are also shown.

Intuition suggests that information on unresolved defects of all severities would be needed to accurately characterize the overall defect resolution process consisting of low, medium, and high-severity defects. Therefore, the defect resolution data was discretized into intervals of 20-time units for the covariate model. The covariates of the i^{th} interval

$$\mathbf{x}_i = < \mathbf{x}_{i,low}, \mathbf{x}_{i,medium}, \mathbf{x}_{i,high} >$$

were determined as the number of defects of a given severity discovered by the beginning of interval i but not yet resolved and y_i was the number of defects of any severity resolved in that interval. Based on this division of resolution time data into intervals of 20-time units, the covariate model with Discrete Weibull hazard rate and parameters

$$\hat{\beta}_{Low} = 0.0202, \hat{\beta}_{Medium} = 0.0519, \hat{\beta}_{High} = 0.1003$$

achieved the best fit, suggesting that high-severity defects contributed most to defect resolution, followed by medium and low-severity defects. Thus, while medium severity defects were the most common, followed by low and high severity, respectively, high severity defects most significantly influenced the defect resolution process as characterized by the covariate model parameters. Careful examination of the data indicated that defects were often resolved in batches. Therefore, one plausible explanation for the numerical parameters is that low and medium-severity defects were often resolved when high-severity defects were resolved. Still, high-severity defects drove the defect resolution planning process.

Table 1 summarizes the performance of each model concerning the goodness of fit measures described in Section 4.

Table 1. Comparison of defect resolution models

Approach	SSE	PSSE (90%)	AIC	BIC	Runtime (s)
D-E	3.77×10^4	2.99×10^3	3.54×10^4	3.83×10^4	175.2
Distributional	3.43×10^4	1.68×10^3	3.32×10^4	3.64×10^4	150.1
Covariate	$\mathbf{1.72 \times 10^3}$	$\mathbf{7.94 \times 10^2}$	$\mathbf{1.29 \times 10^3}$	$\mathbf{1.45 \times 10^3}$	**110.2**

Values in bold indicate the best model concerning each measure. Specifically, the discrete Cox proportional hazard model with covariates denoting the number of defects of high, medium, and low severity discovered but unresolved achieved an order of magnitude lower sum of squares error, PSSE, AIC, and BIC and required less time to apply.

5.2.1 Distributional Approach:

The mean time to resolution used to plot the mean value function of the defect resolution curve for the distributional approach in Figure 1 ($E[\hat{T}_r] = 59.74$) was determined with the special case of Equation (8), composed of all discovery and resolution times for high, medium, and low severity defects through the time at which the n^{th} defect was resolved. Seventeen possible distributions were considered, including the Beta, Birnbaum- Saunders, Exponential, Extreme value, Gamma, Generalized extreme value, Generalized Pareto, Inverse Gaussian, Logistic, Log-logistic, Lognormal, Nakagami, Normal, Rayleigh, Rician, t location-scale, and Weibull distributions. The generalized extreme value (GEV) distribution possessing the following maximum likelihood estimates

$$\hat{T}^r \sim GEV(\mu = 57.4437, \sigma = 22.6722, \xi = -0.0959)$$

exhibited the best fit for the times between defect discovery and resolution for the Akaike and Bayesian Information Criterion.

Figure 2 shows the fitted Generalized extreme value distribution and histogram of times between defect discovery and resolution. Thus, the plot of the distributional approach is simply the inflexion S-shaped model that was fit to the defect discovery data given by equation (3) shifted to the right by the mean time to resolution ($E[\hat{T}_r]$), as defined in Equation (9).

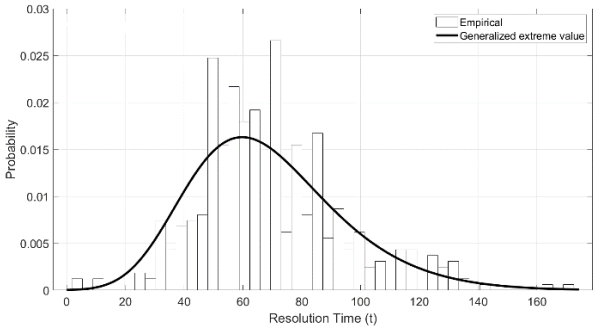


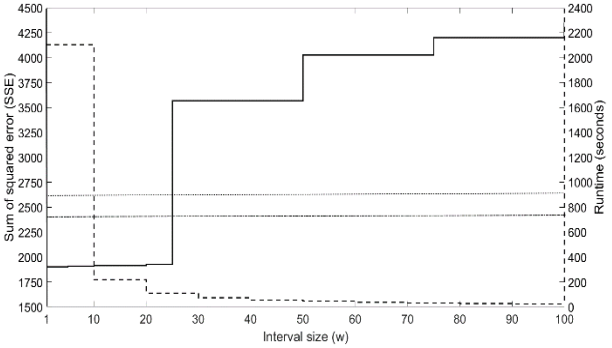
Figure 2. Empirical distribution of time between defect discovery and resolution of all defects

5.2.2 Covariate data analysis methodology:

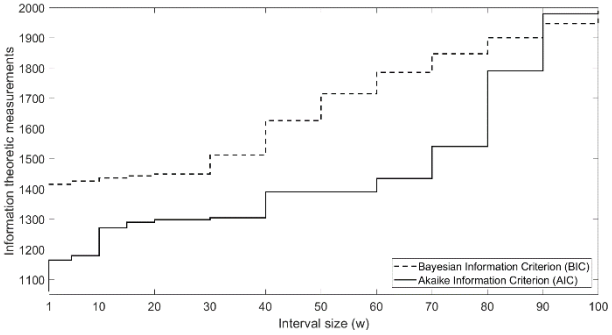
Dividing resolution time data into 20-unit intervals for the covariate model considered in Figure 1 and Table 1 is arbitrary. Thus, to assess interval width on the accuracy of the covariate model, Figure 3a shows the SSE of predictions made with widths $w \in \{5,10, \ldots,100\}$ as well as intervals of width one. Figure 3a indicates that a width 25 or less performed best and that decreasing w from five to one did not significantly decrease the SSE. Moreover, since the time of the last resolution was $t_n^r \approx 2096$, there were 21 intervals for $w = 100$, and 2096 intervals for $w = 1$. In cases where t_n^r/w was not an integer; the width of the final interval was reduced. For example, the width of the final interval was 96 when $w = 100$. Since this was the final interval and most defects had been discovered and resolved, this modest amount of non-uniformity in the width of the intervals did not impact the model fit substantially.

Figure 3a also shows that intervals of width $15 \leq w \leq 25$ did not substantially increase the time required to fit the model to achieve an order of magnitude improvement in model fit over the alternative approaches, whereas the differential equation-based and distributional approaches required 886.2 and 749.5 seconds to apply respectively and the SSE was an order of magnitude worse as noted in Table 1. To demonstrate that smaller intervals are also justified from an information theoretical standpoint, Figure 3b shows the impact of interval width on BIC and AIC.

Since smaller values are preferred, intervals of unit width achieved the best fit. Nevertheless, intervals of width 20 may be adequate since the relative error between the BIC of models with widths 20 and 1 was just 0.0251 ((1450.1-1414.6)/1414.6) and the relative error in the AIC of models with widths twenty and one was 0.2169 ((1290.1- 1060.1)/1060.1). While the relative error of the AIC is not as small as the BIC's, it should be noted that AIC does not penalize the sample size. Thus, BIC may be a better measure of goodness of fit because decreasing w from twenty to one increases the sample size by a factor of twenty. The slight decrease in BIC from twenty to one suggests that the penalty associated with this increase in sample nullifies most gains in maximum likelihood attained.



(a)



(b)

Figure 3. (a) Impact of interval width on SSE and time required to fit Covariate models **(b)** Impact of interval width on BIC and AIC of Covariate models

5.2.3 Retrospective analysis by severity:

Since the response variable of the covariate model is the vector of defects $\mathbf{y}_n$ resolved in each of the n intervals, the model is not capable of simultaneously predicting the number of defects of low, medium, and high severity defects resolved in each interval with Equation (13). However, it is straightforward to apply Equation (13) with the three covariates $\mathbf{x}_i$ as inputs and $\mathbf{y}_n^s$, the vector of defects resolved in each of the n intervals, where $\mathbf{y}_i^s$ is the number of defects of severity $s \in \{1,2,3\}$ resolved in interval i as the response.

Table 2 summarizes the performance of each resolution model for low, medium, and high-severity defects, respectively.

Table 2. Comparison of defect resolution models by severity

Approach	S	SSE	PSSE (90%)	AIC	BIC	Runtime (s)
D-E Based Distributional Covariate	3	5.12×10^3 4.67×10^3 $\mathbf{9.16 \times 10^2}$	6.02×10^2 4.23×10^2 $\mathbf{1.62 \times 10^1}$	4.82×10^3 4.52×10^3 $\mathbf{1.76 \times 10^2}$	5.22×10^3 4.96×10^3 $\mathbf{1.97 \times 10^2}$	24.8 20.5 **14.1**
D-E Based Distributional Covariate	2	3.10×10^4 2.47×10^4 $\mathbf{5.73 \times 10^3}$	6.29×10^3 5.26×10^3 $\mathbf{5.54 \times 10^2}$	2.94×10^4 2.78×10^4 $\mathbf{1.09 \times 10^3}$	3.23×10^4 3.04×10^4 $\mathbf{1.25 \times 10^3}$	145.6 124.7 **91.3**
D-E Based Distributional Covariate	1	1.07×10^3 9.21×10^2 $\mathbf{1.72 \times 10^2}$	3.86×10^2 5.12×10^2 $\mathbf{9.08 \times 10^1}$	1.04×10^3 9.32×10^2 $\mathbf{3.59 \times 10^1}$	1.09×10^3 1.04×10^3 $\mathbf{4.45 \times 10^1}$	5.2 4.1 **3.3**

Values in bold indicate the model that performed best concerning each measure. Table 2 indicates that the discrete Cox proportional hazard model with covariates and second-order Discrete Weibull hazard function outperformed the alternatives by an order of magnitude for each of the three severities and exhibited better runtime performance.

Figure 4 shows the generalized Pareto, generalized extreme value, and exponential distributions that best fit the times between discovery and resolution of low, medium, and high severity defects according to the BIC

and AIC and the corresponding histogram of times between defect discovery and resolution. The primary observation derived from these fitted distributions was that the mean time to resolution of low, medium, and high severity defects were 70.27, 66.19, and 54.76 days, indicating that higher severity defects were resolved more quickly. This suggests that, on average, higher-severity defects received more attention. Efforts to document the effort dedicated to resolving these defects will further enhance the applicability of the covariate approach.

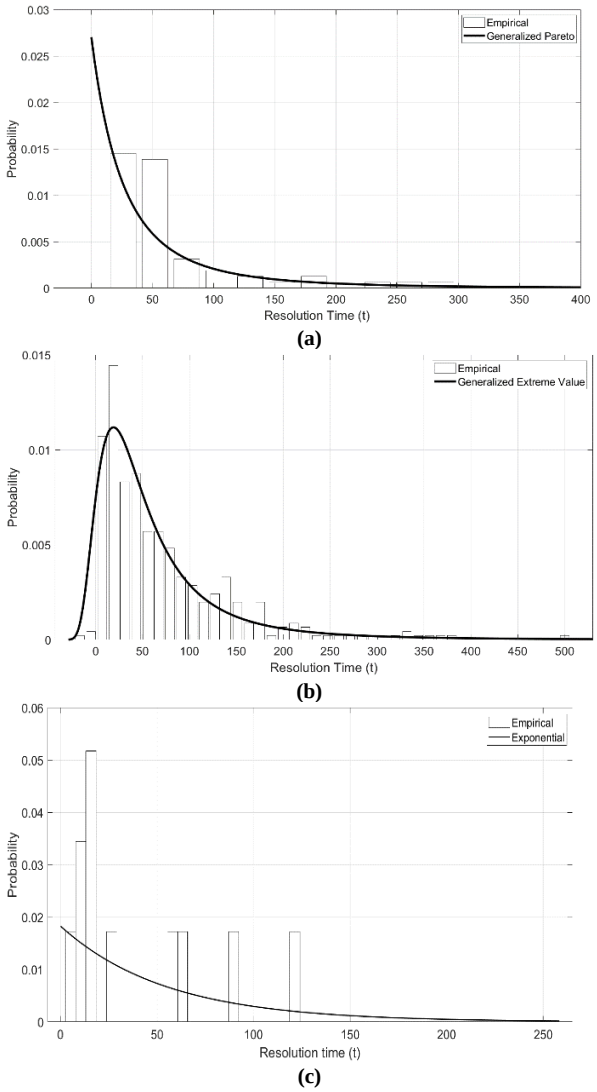


Figure 4. (a) Empirical distribution of time between discovery and resolution of low severity defects **(b)** Empirical distribution of time between discovery and resolution of medium severity defects **(c)** Empirical distribution of time between discovery and resolution of high-severity defects

5.3 Analysis of unresolved defects

To provide an alternative perspective, Figure 5 shows the number of unresolved defects at the end of each interval and the predictions of the fitted models. Increases (decreases) in the step function indicate times at which defects were discovered (resolved) more quickly than they were resolved (discovered). Thus, the value on the y-axis represents the difference between the number of defects discovered by time t ($N(t)$) and the number of defects resolved by time t ($N_r(t)$). At the peak ($t = 1200$), nearly 50 defects were unresolved.

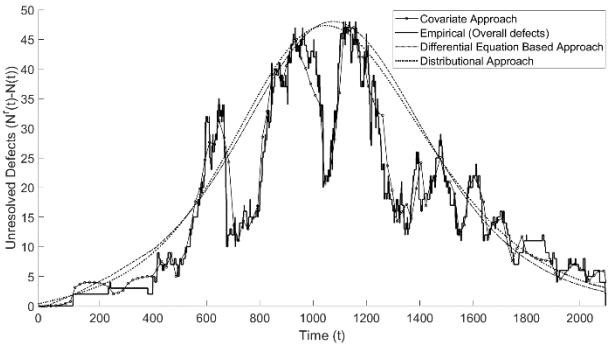


Figure 5. Open defects (discovered but not yet resolved) and fitted models

Figure 5 indicates that the differential equation-based and distributional approaches only capture primary trends according to their parametric form, whereas the covariate approach tracks the unresolved defects remarkably well. The number of defects discovered but not resolved at time t or interval i was computed by subtracting the MVF of the defect discovery process ($\widehat{m}(t)$) of Equation (3) from the fitted MVF of the defect resolution process ($\widehat{m}_r^b(t)$ or $m_r(x)$) from Equation (5), (9) or Equation (3) for the differential equation-based, distributional, and covariate approach, respectively.

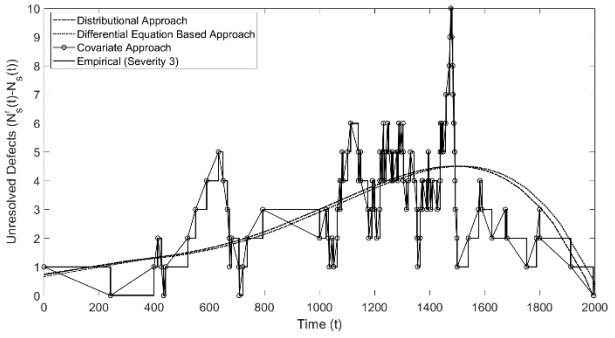
Table 3 summarizes the model assessments for unresolved defects of any severity, indicating the covariate approach outperforms the differential equation-based and distributional approaches by an order of magnitude, reflecting the superior fit of the covariate model fit to the number of unresolved defects in each interval observed in Figure 5. To fairly compare continuous and discrete models, SSE and PSSE were computed at the end of each discrete interval to avoid favoring the discrete model when the number of intervals was smaller than the number of defects resolved.

Table 3. Comparison of models on unresolved defects

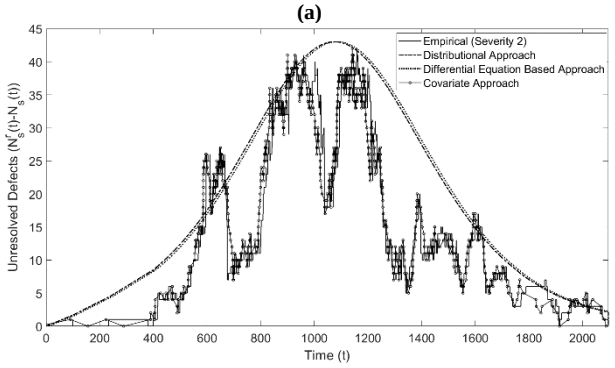
Approach	SSE	PSSE (90%)	AIC	BIC	Runtime (s)
D-E	5.28×10^4	2.10×10^2	7.10×10^4	7.23×10^4	330.2
Distributional	3.03×10^4	2.10×10^2	7.05×10^4	7.06×10^4	312.1
Covariate	$\mathbf{1.92 \times 10^3}$	$\mathbf{4.59 \times 10^1}$	$\mathbf{3.59 \times 10^3}$	$\mathbf{3.71 \times 10^3}$	**186.1**

5.3.1 Analysis of unresolved defects by severity:

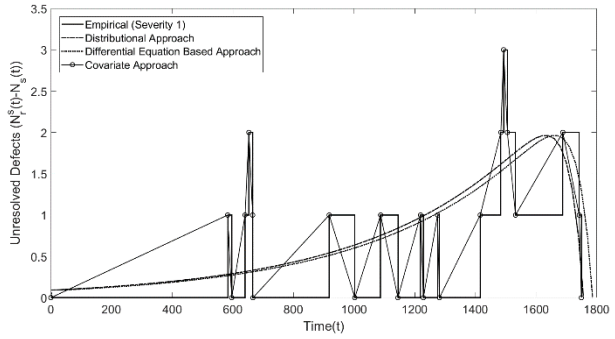
Figure 6 shows the number of unresolved defects of low, medium, and high severity defects and corresponding model fits. In each case, differential equation-based and distributional approaches only capture a single peak in the trend. Still, the covariate approach tracks the number of open defects extremely well, even the less frequent low and high-severity defects.



(a)



(b)



(c)

Figure 6. (a) Open defects of low severity ($s = 3$) and fitted models, **(b)** Open defects of medium severity ($s = 2$) and fitted models, **(c)** Open defects of high severity ($s = 1$) and fitted models

Table 4 summarizes the model assessments for low, medium, and high-severity unresolved defects.

Table 4. Comparison of models on unresolved defects by severity

Approach	S	SSE	PSSE (90%)	AIC	BIC	Runtime (s)
D-E Based	3	3.10×10^3	3.43×10^2	3.87×10^3	4.12×10^3	45.2
Distributional		2.62×10^3	1.58×10^2	3.31×10^3	3.67×10^3	42.8
Covariate		$\mathbf{1.71 \times 10^2}$	$\mathbf{1.01 \times 10^1}$	$\mathbf{4.61 \times 10^2}$	$\mathbf{4.85 \times 10^2}$	**24.8**
D-E Based	2	3.82×10^4	2.66×10^3	2.81×10^4	2.99×10^4	277.3
Distributional		2.11×10^4	1.09×10^3	2.74×10^4	2.91×10^4	262.1
Covariate		$\mathbf{2.66 \times 10^3}$	$\mathbf{1.11 \times 10^2}$	$\mathbf{2.85 \times 10^3}$	$\mathbf{2.93 \times 10^3}$	**156.7**
D-E Based	1	1.79×10^2	1.85×10^1	1.86×10^3	2.01×10^2	9.6
Distributional		1.49×10^2	5.84×10^1	$\mathbf{1.15 \times 10^2}$	$\mathbf{1.54 \times 10^2}$	10.9
Covariate		$\mathbf{2.27 \times 10^1}$	$\mathbf{1.43 \times 10^1}$	5.71×10^2	5.98×10^2	**5.3**

Once again, the covariate approach performed best on virtually all measures of goodness of fit and required less time to apply. However, the distributional approach performed best on high severity ($s = 1$) possibly because of the low sample size.

5.4 Assessment of Predictive Accuracy

Ideally, a model should be simple and accurately predict the distant future with little data. To compare the predictive accuracy of the defect resolution models, this section performs an online assessment of the models with the predictive SSE measure. The defect resolution processes of the differential equation-based and covariate approaches (Equations (5) and (10) respectively) were fit to the resolution time data extracted from the defect tracking database, whereas the distributional approach identified an SRGM that fit the available defect discovery data best, estimated the MTTR with Equation (8), and then substituted the MTTR into Equation (9). The PSSE was subsequently computed according to Equation (15) as the sum of squares difference between the actual number of defects observed and model predictions at each resolution time t_i^r or each interval i. For the sake of comparison, the amount of data provided for defect discovery and resolution model fitting was performed in increments of 20, the width of the intervals in the covariate approach.

Figure 7 shows the online assessment of the defect resolution models with PSSE for defects of all three severities combined. Times before $t = 200$ are excluded, so primary trends are distinguishable since PSSE was initially very large and would skew the remainder of the graph.

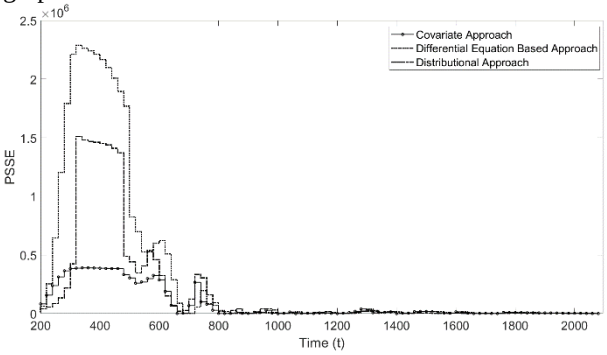


Figure 7. PSSE of models on defects of three severities combined

Figure 7 indicates that the covariate approach exhibited substantially lower error than the alternatives and sustained the highest accuracy throughout the remainder of the defect discovery and resolution process. The covariate approach is accurate because of the availability of information on the number of open defects by severity. In contrast, the distributional approach exhibits error since few of the times between defect discovery and resolution shown in Figure 2 were observed before $t = 600$. Predictions of the differential

equation-based approach were the worst because parametric models implicitly make rigid assumptions about the shape of the defect resolution curve.

Figure 8 shows the online assessment results of the defect resolution models with PSSE by severity.

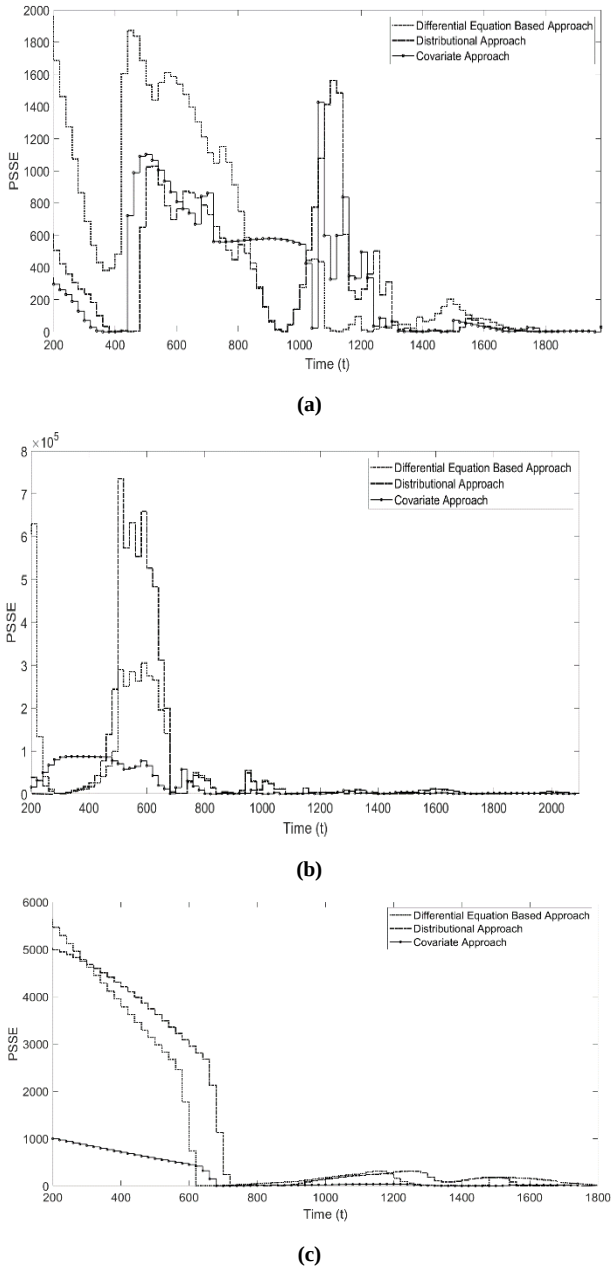


Figure 8. (a) PSSE of models on low-severity defects, **(b)** PSSE of models on medium-severity defects, **(c)** PSSE of models on high-severity defects

Figure 8a indicates that the distributional and covariate approaches predicted approximately the same prior to $t = 800$ for low-severity defects. In contrast, Figures 8b and 8c, respectively, show that the predictions of the covariate approach were best for medium severity defects at times $t > 400$ and at all times for high severity defects. The prediction errors may be partly explained by the sample size of low and high-severity defects, the parametric forms for the differential equation-based and distributional models, and undocumented factors within the software test process. The excellent predictive accuracy of the covariate approach on high-severity defects is promising. However, disciplined collection of covariates related to defect resolution efforts could substantially reduce prediction errors, supporting risk mitigation efforts to ensure high-severity defects are removed prior to fielding.

6. Conclusion and Future Research

This paper presented a model for the number of defects detected and resolved according to the discrete Cox proportional hazard model incorporating covariates describing metrics or activities that could serve as predictors. Defect resolution activities and the amount of effort dedicated to each were not explicitly documented in the NASA defect tracking database. So, the number of low, medium, and high-severity unresolved defects were used as covariates. The illustrations showed that the covariate approach outperformed other models by an order of magnitude on all goodness of fit measures considered and required less time to apply, exhibiting similar performance when applied to subsets of data for low, medium, and high severity defects. A similar analysis of the number of unresolved defects demonstrated compelling evidence that the covariate approach tracked the data much better than the alternative approaches. Finally, the covariate approach exhibited low predictive error, even when only 10- 20% of testing had elapsed.

Future research will seek to improve the efficiency of the model fitting procedure for the covariate approach when (i) the data consists of a large number of intervals and (ii) the number of covariates describing the effort allocated to distinct defect resolution activities in each interval is large.

Acknowledgments

This material is based upon work supported by the National Aeronautics and Space Administration under Grant Number 80NSSC20K0276 and the U.S. National Science Foundation under Grant Number 1749635. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the official policy or position of the National Aeronautics and Space Administration or the U.S. National Science Foundation.

Conflict of Interests

No conflict of interest has been expressed by the authors.

7. References

[1] Z. Jelinski and P. Moranda, "SOFTWARE RELIABILITY RESEARCH," in *Statistical Computer Performance Evaluation*, W. Freiberger Ed.: Academic Press, 1972, pp. 465-484.

[2] N. F. Schneidewind, "Analysis of error processes in computer software," in *Proceedings of the international conference on Reliable software*, 1975, pp. 337-346, doi: https://doi.org/10.1145/800027.808456.

[3] J.-H. Lo and C.-Y. Huang, "An integration of fault detection and correction processes in software reliability analysis," *Journal of Systems and Software*, vol. 79, no. 9, pp. 1312-1323, 2006/09/01/ 2006, doi: https://doi.org/10.1016/j.jss.2005.12.006.

[4] K. Shibata, K. Rinsaka, and T. Dohi, "Metrics-Based Software Reliability Models Using Non-homogeneous Poisson Processes," in *2006 17th International Symposium on Software Reliability Engineering*, 7-10 Nov. 2006 2006, pp. 52-61, doi: https://doi.org/10.1109/ISSRE.2006.28.

[5] T. Dohi, T. Matsuoka, and S. Osaki, "An infinite server queuing model for assessment of the software reliability," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 85, no. 3, pp. 43-51, 2002/03/01 2002, doi: https://doi.org/10.1002/ecjc.1078.

[6] T. Dohi, S. Osaki, and K. S. Trivedi, "An infinite server queueing approach for describing software reliability growth: unified modeling and estimation framework," in *11th Asia-Pacific Software Engineering Conference*, 30 Nov.-3 Dec. 2004 2004, pp. 110-119, doi: https://doi.org/10.1109/APSEC.2004.29.

[7] S. S. Gokhale and R. E. Mullen, "Queuing models for field defect resolution process," in *2006 17th International Symposium on Software Reliability Engineering*, 2006: IEEE, pp. 353-362, doi: https://doi.org/10.1109/ISSRE.2006.38.

[8] P. N. Misra, "Software reliability analysis," *IBM Systems Journal*, vol. 22, no. 3, pp. 262-270, 1983, doi: https://doi.org/10.1147/sj.223.0262.

[9] S. Yamada, S. Osaki, and H. Narihisa, "A software reliability growth model with two types of errors," *RAIRO-Operations Research*, vol. 19, no. 1, pp. 87-104, 1985, doi: https://doi.org/10.1051/ro/1985190100871.

[10] L. Fiondella and S. S. Gokhale, "Software Reliability Models Incorporating Testing Effort," *OPSEARCH*, vol. 45, no. 4, pp. 351-368, 2008/12/01 2008, doi: https://doi.org/10.1007/BF03398825.

[11] K. Rinsaka, K. Shibata, and T. Dohi, "Proportional intensity-based software reliability modeling with time-dependent metrics," in *30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, 2006, vol. 1: IEEE, pp. 369-376, doi: https://doi.org/10.1109/COMPSAC.2006.68.

[12] M. Nafreen, M. Luperon, L. Fiondella, V. Nagaraju, Y. Shi, and T. Wandji, "Connecting software reliability growth models to software defect tracking," in *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, 2020: IEEE, pp. 138-147, doi: https://doi.org/10.1109/ISSRE5003.2020.00022.

[13] H. Sukhwani, J. Alonso, K. S. Trivedi, and I. Mcginnis, "Software reliability analysis of NASA space flight software: A practical experience," in *2016 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, 2016: IEEE, pp. 386-397, doi: https://doi.org/10.1109/QRS.2016.50.

[14] M. Xie and M. Zhao, "The Schneidewind software reliability model revisited," in *Proceedings Third International Symposium on Software Reliability Engineering*, 7-10 October 1992 1992, pp. 184-192, doi: https://doi.ieeecomputersociety.org/10.1109/ISSRE.1992.285846.

[15] M. Ohba, "Inflection S-shaped software reliability growth model," in *Stochastic Models in Reliability Theory: Proceedings of a Symposium Held in Nagoya, Japan, April 23–24, 1984*, 1984: Springer, pp. 144-162, doi: https://doi.org/10.1007/978-3-642-45587-2_10.

[16] S. Yamada, M. Ohba, and S. Osaki, "S-shaped reliability growth modeling for software error detection," *IEEE Transactions on reliability*, vol. 32, no. 5, pp. 475-484, 1983, doi: https://doi.org/10.1109/TR.1983.5221735.

[17] P. Kapur and S. Younes, "Software reliability growth model with error dependency," *Microelectronics Reliability*, vol. 35, no. 2, pp. 273-278, 1995, doi: https://doi.org/10.1016/0026-2714(94)00054-R.

[18] S. Yamada, K. Tokuno, and S. Osaki, "Imperfect debugging models with fault introduction rate for software reliability assessment," *International Journal of Systems Science*, vol. 23, no. 12, pp. 2241-2252, 1992, doi: https://doi.org/10.1080/00207729208949452.

[19] S. S. Gokhale, P. N. Marinos, M. Lyn, and K. S. Trivedi, "Effect of repair policies on software reliability," in *Proceedings of COMPASS'97: 12th Annual Conference on Computer Assurance*, 1997: IEEE, pp. 105-116, doi: https://doi.org/10.1109/CMPASS.1997.613262.

[20] C.-Y. Huang, C.-T. Lin, S.-Y. Kuo, M. R. Lyu, and C.-C. Sue, "Software reliability growth models incorporating fault dependency with various debugging time lags," in *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.*, 2004: IEEE, pp. 186-191, doi: https://doi.org/10.1109/CMPSAC.2004.1342826.

[21] N. Ullah, M. Morisio, and A. Vetro, "A comparative analysis of software reliability growth models using defects data of closed and open source software," in

2012 35th Annual IEEE Software Engineering Workshop, 2012: IEEE, pp. 187-192, doi: https://doi.org/10.1109/SEW.2012.26.

[22] Y. Liu, M. Xie, J. Yang, and M. Zhao, "A new framework and application of software reliability estimation based on fault detection and correction processes," in *2015 IEEE International Conference on Software Quality, Reliability and Security*, 2015: IEEE, pp. 65-74, doi: https://doi.org/10.1109/QRS.2015.20.

[23] J. Yang, Y. Liu, M. Xie, and M. Zhao, "Modeling and analysis of reliability of multi-release open source software incorporating both fault detection and correction processes," *Journal of Systems and Software*, vol. 115, pp. 102-110, 2016/05/01/ 2016, doi: https://doi.org/10.1016/j.jss.2016.01.025.

[24] M. Cinque, D. Cotroneo, A. Pecchia, R. Pietrantuono, and S. Russo, "Debugging-workflow-aware software reliability growth analysis," *Software Testing, Verification and Reliability*, vol. 27, no. 7, p. e1638, 2017, doi: https://doi.org/10.1002/stvr.1638.

[25] P. Vizarreta *et al.*, "Assessing the maturity of SDN controllers with software reliability growth models," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1090-1104, 2018, doi: https://doi.org/10.1109/TNSM.2018.2848105.

[26] P. Vizarreta, E. Sakic, W. Kellerer, and C. M. Machuca, "Mining software repositories for predictive modelling of defects in sdn controller," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019: IEEE, pp. 80-88.

[27] R. Xie, H. Qiu, Q. Zhai, and R. Peng, "A model of software fault detection and correction processes considering heterogeneous faults," *Quality and Reliability Engineering International*, vol. 39, no. 8, pp. 3428-3444, 2023, doi: https://doi.org/10.1002/qre.3172.

[28] H. Joe, "Statistical inference for general-order-statistics and nonhomogeneous-Poisson-process software reliability models," *IEEE Transactions on Software Engineering*, vol. 15, no. 11, pp. 1485-1490, 1989, doi: https://doi.org/10.1109/32.41340.

[29] C.-T. Lin, C.-Y. Huang, and C.-C. Sue, "Measuring and assessing software reliability growth through simulation-based approaches," in *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, 2007, vol. 1: IEEE, pp. 439-448, doi: https://doi.org/10.1109/COMPSAC.2007.141.

[30] C.-Y. Huang and W.-C. Huang, "Software reliability analysis and measurement using finite and infinite server queueing models," *IEEE Transactions on Reliability*, vol. 57, no. 1, pp. 192-203, 2008.

[31] N. Zhang, G. Cui, and H.-w. Liu, "A finite queuing model with generalized modified Weibull testing effort for software reliability," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, 2011, vol. 1: IEEE, pp. 401-406, doi: https://doi.org/10.1109/ICCSNT.2011.6181985.

[32] P. Kapur, H. Pham, S. Anand, and K. Yadav, "A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation," *IEEE Transactions on Reliability*, vol. 60, no. 1, pp. 331-340, 2011, doi: https://doi.org/10.1109/TR.2010.2103590.

[33] C.-Y. Huang and T.-Y. Kuo, "Queueing-theory-based models for software reliability analysis and management," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 540-550, 2017, doi: https://doi.org/10.1109/TETC.2014.2388454.

[34] K. Tokuno, T. Nagata, and S. Yamada, "Stochastic software performability evaluation based on NHPP reliability growth model," *International Journal of Reliability, Quality and Safety Engineering*, vol. 18, no. 05, pp. 431-444, 2011, doi: https://doi.org/10.1142/S0218539311004172.

[35] H. Okamura and T. Dohi, "A generalized bivariate modeling framework of fault detection and correction processes," in *2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE)*, 2017: IEEE, pp. 35-45, doi: https://doi.org/10.1109/ISSRE.2017.22.

[36] T. M. Khoshgoftaar and J. C. Munson, "Predicting software development errors using software complexity metrics," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 2, pp. 253-261, 1990, doi: https://doi.org/10.1109/49.46879.

[37] T. M. Khoshgoftaar, B. B. Bhattacharyya, and G. D. Richardson, "Predicting software errors, during development, using nonlinear regression models: a comparative study," *IEEE Transactions on Reliability*, vol. 41, no. 3, pp. 390-395, 1992, doi: https://doi.org/10.1109/24.159804.

[38] T. M. Khoshgoftaar, J. C. Munson, B. B. Bhattacharya, and G. D. Richardson, "Predictive modeling techniques of software quality from software measures," *IEEE Transactions on Software Engineering*, vol. 18, no. 11, pp. 979-987, 1992, doi: https://doi.org/10.1109/32.177367.

[39] Y. Shi, M. Li, S. Arndt, and C. Smidts, "Metric-based software reliability prediction approach and its application," *Empirical Software Engineering*, vol. 22, pp. 1579-1633, 2017, doi: https://doi.org/10.1007/s10664-016-9425-9.

[40] W. M. Evanco and R. Lacovara, "A model-based framework for the integration of software metrics," *Journal of Systems and Software*, vol. 26, no. 1, pp. 77-86, 1994/07/01/ 1994, doi: https://doi.org/10.1016/0164-1212(94)90098-1.

[41] W. M. Evanco, "Poisson analyses of defects for small software components," *Journal of Systems and Software*, vol. 38, no. 1, pp. 27-35, 1997/07/01/ 1997, doi: https://doi.org/10.1016/S0164-1212(97)00063-0.

[42] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other

performance measures," *IEEE transactions on Reliability*, vol. 28, no. 3, pp. 206-211, 1979, doi: https://doi.org/10.1109/TR.1979.5220566.

[43] J. E. RamÍRez Cid and J. Alberto Achcar, "Software Reliability Considering the Superposition of Non-homogeneous Poisson Processes in the Presence of a Covariate," *Statistics*, vol. 36, no. 3, pp. 259-269, 2002/01/01 2002, doi: https://doi.org/10.1080/02331880212854.

[44] B. K. Ray, Z. Liu, and N. Ravishanker, "Dynamic reliability models for software using time-dependent covariates," *Technometrics*, vol. 48, no. 1, pp. 1-10, 2006, doi: https://doi.org/10.1198/004017005000000292.

[45] A. Gandy and U. Jensen, "A non-parametric approach to software reliability," *Applied Stochastic Models in Business and Industry*, vol. 20, no. 1, pp. 3-15, 2004, doi: https://doi.org/10.1002/asmb.510.

[46] T. Ishii, T. Fujiwara, and T. Dohi, "Bivariate extension of software reliability modeling with number of test cases," *International Journal of Reliability, Quality and Safety Engineering*, vol. 15, no. 01, pp. 1-17, 2008, doi: https://doi.org/10.1142/S0218539308002897.

[47] P. Kapur, A. G. Aggarwal, and A. Tandon, "Two dimensional software reliability growth model with faults of different severity," *Communications in Dependability and Quality Management*, vol. 13, no. 4, pp. 98-110, 2010.

[48] H. Okamura, Y. Etani, and T. Dohi, "A multi-factor software reliability model based on logistic regression," in *2010 IEEE 21st International Symposium on Software Reliability Engineering*, 2010: IEEE, pp. 31-40, doi: https://doi.org/10.1109/ISSRE.2010.14.

[49] H. Okamura, Y. Etani, and T. Dohi, "Quantifying the effectiveness of testing efforts on software fault detection with a logit software reliability growth model," in *2011 Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement*, 2011: IEEE, pp. 62-68, doi: https://doi.org/10.1109/IWSM-MENSURA.2011.26.

[50] D. Kuwa and T. Dohi, "Generalized Logit Regression-Based Software Reliability Modeling with Metrics Data," in *2013 IEEE 37th Annual Computer Software and Applications Conference*, 2013: IEEE, pp. 246-255, doi: https://doi.org/10.1109/COMPSAC.2013.41.

[51] D. Kuwa, T. Dohi, and H. Okamura, "Generalized Cox proportional hazards regression-based software reliability modeling with metrics data," in *2013 IEEE 19th Pacific Rim International Symposium on Dependable Computing*, 2013: IEEE, pp. 328-337, doi: https://doi.org/10.1109/PRDC.2013.55.

[52] H. Okamura and T. Dohi, "A novel framework of software reliability evaluation with software reliability growth models and software metrics," in *2014 IEEE 15th International Symposium on High-Assurance Systems Engineering*, 2014: IEEE, pp. 97-104, doi: https://doi.org/10.1109/HASE.2014.22.

[53] H. Okamura and T. Dohi, "Towards comprehensive software reliability evaluation in open source software," in *2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE)*, 2015: IEEE, pp. 121-129, doi: https://doi.org/10.1109/ISSRE.2015.7381806.

[54] M. Wiper, A. Palacios, and J. Marín, "Bayesian software reliability prediction using software metrics information," *Quality Technology & Quantitative Management*, vol. 9, no. 1, pp. 35-44, 2012, doi: https://doi.org/10.1080/16843703.2012.11673276.

[55] N. Torrado, M. P. Wiper, and R. E. Lillo, "Software reliability modeling with software metrics data via Gaussian processes," *IEEE Transactions on Software Engineering*, vol. 39, no. 8, pp. 1179-1186, 2012, doi: https://doi.org/10.1109/TSE.2012.87.

[56] V. Nagaraju, C. Jayasinghe, and L. Fiondella, "Optimal test activity allocation for covariate software reliability and security models," *Journal of Systems and Software*, vol. 168, p. 110643, 2020/10/01/ 2020, doi: https://doi.org/10.1016/j.jss.2020.110643.

[57] S. Yamada, J. Hishitani, and S. Osaki, "Software-reliability growth with a Weibull test-effort: a model and application," *IEEE Transactions on Reliability*, vol. 42, no. 1, pp. 100-106, 1993, doi: https://doi.org/10.1109/24.210278.

[58] P. Moranda, "Prediction of software reliability during debugging," in *Proc. 1975 Annu. Reliab. Maintainance Symp.*, 1975.

[59] W. Farr, "Software reliability modeling survey," in *Handbook of software reliability engineering*: McGraw-Hill, Inc., 1996, pp. 71–117.

[60] T. J. Archdeacon, *Correlation and regression analysis: a historian's guide*. Univ of Wisconsin Press, 1994.

[61] K. Sharma, R. Garg, C. K. Nagpal, and 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, 2010, doi: https://doi.org/10.1109/TR.2010.2048657.

[62] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716-723, 1974, doi: https://doi.org/10.1109/TAC.1974.1100705.

[63] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, pp. 461-464, 1978, doi: https://doi.org/10.1214/aos/1176344136.

International Journal of Reliability, Risk and Safety: Theory and Application

Online ISSN: 2676-3346

OPEN ACCESS

Vol. 7/ Issue 1/ Year 2024/ pp. 59-72
DOI: 10.22034/IJRRS.2024.7.1.8
Received: 28 February 2024, Revised: 12 May 2024, Accepted: 25 May 2024
Available online at: https://www.ijrrs.com

Original Research Article

Software Defect Discovery and Resolution Modeling incorporating Severity

Maskura Nafreen[1], Ying Shi[2] and Lance Fiondella[1*]

1- Department of Electrical and Computer Engineering, University of Massachusetts Dartmouth, North Dartmouth, Massachusetts, USA

2- Mission Software and Ground System Assurance Branch, NASA Goddard Space Flight Center, Greenbelt, Maryland, USA

*** lfiondella@umassd.edu**

Abstract

Traditional software reliability growth models only consider defect discovery data, yet the primary concern of software engineers is defect removal. Past attempts to model defect resolution emphasize approaches based on differential equations and queueing theory. However, these models do not explicitly identify the activities performed to remove defects and resources allocated to these activities according to their severity. Models should consider these practical factors to enable more detailed resource allocation and planning.

This paper presents a model to predict the number of defects resolved according to the discrete Cox proportional hazard model with covariates, demonstrating the approach with covariates on the number of low, medium, and high severity defects that were discovered but not resolved in successive intervals. A comparison with differential equation-based and distributional approaches reveals that the covariate model performs better on each goodness of fit measure considered and requires less time to apply. The covariate model also better tracks unresolved defects and exhibits low predictive error, even when as little as 10-20% of testing has been completed. These results suggest that collecting information on defect resolution activities and the corresponding effort dedicated could substantially improve defect resolution modeling to guide process improvement.

Keywords: Software defect resolution; Software defect severity; Software defect tracking; Software reliability; Software reliability modeling.

Nomenclature

Acronyms

AIC	Akaike information criterion
BIC	Bayesian information criterion
GEV	Generalized extreme value
LL	Log-likelihood function
MLE	Maximum-likelihood estimation
MTTR	Mean time to resolution
MVF	Mean value function
NASA	National Aeronautics and Space Administration
NHPP	Non-homogeneous Poisson process
PSSE	Predictive sum of squares error
SRGM	Software reliability growth model
SSE	Sum of squares error

Notation

$N(t)$	Number of defects detected by time t
$N_r(t)$	Number of defects resolved by time t
$m(t)$	Mean number of defects detected by time t
$m_r(t)$	Mean number of defects resolved by time t
$m_r^b(t)$	Mean number of defects resolved by time t assuming common discovery and resolution rate b
$\lambda(t)$	Defect discovery rate at time t
$\lambda_r(t)$	Defect resolution rate at time t
$\lambda_r^b(t)$	Defect resolution rate at time t assuming common discovery and resolution rate b

How to cite this article:
M. Nafreen, Y. Shi and L. Fiondella, "Software Defect Discovery and Resolution Modeling incorporating Severity," *International Journal of Reliability, Risk and Safety: Theory and Application*, vol. 7, no. 1, pp. 59-72, 2024, doi: 10.22034/IJRRS.2024.7.1.8

$\mathcal{D}$	Set of defects discovered
$\mathcal{R}$	Set of defects resolved
t_i	Time at which i^{th} defect was discovered
t_i^r	Time at which i^{th} defect was resolved
$t_{(i)}$	Time between discovery and resolution of i^{th} defect
$E[\hat{T}_r]$	Mean time to defect resolution
n	Total number of defects or intervals
w	Interval width
m	Number of covariates
$\mathbf{x}_{n\times m}$	Matrix of observed covariates
$\mathbf{x}_{i\times j}$	Effort dedicated to activity j in interval i
y_i	Number of defects resolved in interval i
y_i^s	Number of severity s defects resolved in interval i
p_{i,x_i}	Probability defect is resolved in interval i
$h(\cdot)$	Baseline hazard function
β_m	Vector of covariate coefficients

1. Introduction

Software reliability growth models (SRGM) to characterize the defect discovery process during testing have been the subject of study since the early 1970s [1]. While models to describe defect resolution were proposed as early as 1975 [2], defect resolution models [3] only began to receive consistent attention after 2000. Metrics-based models to characterize the number of defects detected in successive intervals [4] as a function of test activities is another topic that has recently enjoyed more thorough study. Since defect resolution, not discovery, is the true source of increased software reliability, defect resolution models to explicitly consider the activities performed to resolve defects of different severities are needed to allocate effort during software defect tracking.

Most past papers are based on NHPP SRGM, which we call defect discovery models. In contrast, the most common class of defect resolution models is based on the integrated defect discovery and resolution process modeling framework [3]. Queueing theoretic models of discovery and resolution [5,6] have also been proposed, including multi-priority queuing models [7] for the mean time to resolution according to defect severity, but were limited to homogeneous rates for fielded software and did not consider time-varying rates characteristic of a software testing process over a defined test schedule. Before this, most defect severity models were limited to NHPP SRGM for defect discovery composed of separate mean value functions [8] as well as homogeneous [9] and heterogeneous [10] mixtures of mean value functions. Recent research has demonstrated that metrics-based models [4], [11], also known as covariate models, characterize the defect discovery process in terms of the underlying test activities very well, suggesting covariate models may also effectively characterize the defect resolution process.

This paper presents a software defect resolution model based on the discrete Cox proportional hazard model with covariates and compares it with the software defect resolution models that performed best in a recent study connecting SRGM to defect tracking databases [12], including (i) an integrated defect discovery and resolution process model and (ii) a distributional approach that shifts the defect discovery model by the mean time to defect resolution. Novel models were enabled by a NASA software defect tracking data set [13], including the times when defects were discovered and resolved. This previous study concluded that more complex integrated defect discovery and resolution process models did not characterize the defect resolution process better. It was also observed that a semi-Markov process model of the defect tracking lifecycle did not significantly improve estimates of the mean time to defect resolution. Therefore, while the semi-Markov process model more closely represented the defect tracking process, it only attained marginal improvements in prediction because of sample size and data quality issues. Hence, it was concluded that better data collection practices could improve the utility of the semi-Markov process model. Despite the lack of disciplined data collection practices by software practitioners, it is necessary to create models that predict robustly and are not computationally intensive. Toward this end, the software defect resolution model based on the discrete Cox proportional hazard model with covariates can rely solely on information on open defects (discovered but not yet resolved) of different severity to predict the total number of defects resolved as well as more fine-grained predictions of the number of defects of a specific severity resolved.

Based on the observations above, this paper seeks to enhance the utility of covariate models and encourage their adoption, making the following primary contributions:

- A software defect resolution model incorporating covariates based on a discrete Cox proportional hazard rate to predict the removal of defects by severity in terms of activities or metrics associated with defect removal
- A method to assess the predictive ability of alternative models in terms of the number of open defects by severity

Our results indicate that the defect resolution model incorporating covariates outperformed differential equation-based and distributional approaches by order of magnitude on all measures of goodness of fit and required less runtime to apply. The covariate defect resolution model also achieved compelling visual fits to the number of open defects and achieved greater predictive accuracy earlier in the testing schedule. Thus, while covariates for the activities performed to resolve defects and the effort dedicated to these activities are not needed, they could

certainly improve the accuracy of models and promote process assessment and improvement.

The remainder of the paper is organized as follows: Section 2 summarizes related research. Section 3 describes defect discovery and resolution models. Section 4 describes model assessment techniques. Section 5 provides illustrations comparing the predictive accuracy and performance of alternative models. Section 6 offers conclusions and identifies future research.

2. Related Research

This section summarizes past research on defect discovery and resolution models for software. A chronology of contributions spanning the past 45 years is provided. Related developments are grouped logically, wherever possible. Models explicitly drawing upon queueing theory are also discussed. Since covariate SRGM are employed to characterize defect resolution for the first time in this paper, we also review key developments related to this class of models. The section concludes with a statement of the paper's novel contributions.

Early studies that sought to characterize software defect discovery and resolution include Schneidewind's [2] defect discovery model based on a discrete exponential mean value function and time lag resolution. Xie and Zhao [14] extended this model by assuming the defect resolution rate is proportional to the number of unresolved defects, demonstrating the Poisson thinning process could model the difference between unresolved defects. The inflection S-shaped model was proposed by Ohba [15] to describe scenarios where some defects needed to be resolved before others could be reached. In contrast, the delayed S-shaped model was developed by Yamada et al. [16] by incorporating a time delay to model this dependence. Kapur and Younes [17] modeled leading and dependent defects. Two SRGM with imperfect debugging were proposed by Yamada et al. [18], where new defects could be introduced when other defects were resolved. A non-homogeneous continuous-time Markov chain was employed by Gokhale et al. [19] to model defect repair and analyze the impact of fault removal policies on the number of defects remaining when testing was completed. More recent studies include Huang et al. [20], who showed that applying a time-dependent delay function can derive several existing SRGM. An integrated defect discovery and resolution process modeling framework was proposed by Lo and Huang [3], where the defect resolution process was expressed in terms of a time-varying resolution intensity as well as the difference between the number of defects discovered and resolved. Ullah et al. [21] conducted a comparative analysis of SRGM on discovery and resolution data sets from several dozen open-source software projects. Liu et al. [22] proposed an approach to estimate the parameters of a defect removal model for semi-grouped data consisting of the approximate times at which defects were discovered and resolved, while Yang et al. [23] modeled defect detection and correction of a multi-release open-source software and related optimal release problems. Cinque et al. [24] proposed an NHPP SRGM for debugging data documented in a bug-tracking system with defects of varying severity, improving the accuracy of predictions when debugging activities did not follow the modeling assumptions closely. Vizarreta et al. [25] found the inflection S-shaped model characterized the defect resolution of four successive releases of an open-source software-defined network controller well and subsequently [26] fit cumulative distribution functions for the time to resolution by the severity of defects. Xie et al. [27] proposed a defect resolution model in which defect resolution times are allowed to follow a variety of common distributions.

Applications of queueing theory to software defect discovery and resolution include the work of Dohi et al. [5], who proposed a model of software failure occurrences as an M/G/8 infinite server queue, unifying previously proposed software reliability growth models, including general order statistics models [28]. Dohi et al. [6] subsequently presented Bayesian estimation techniques for their infinite server queueing model and demonstrated improved goodness of fit over general order statistics models. Gokhale and Mullen [7] developed multi-priority queuing models for the software defect resolution process, considering the effect of queuing system structures, priority levels, and priority disciplines on the time to resolve defects of different severities. Lin et al. [29] implemented simulation procedures for G/G/8 and G/G/m infinite server queues for software defect detection and removal. Huang and Huang [30] showed how to incorporate finite and infinite server queueing models into software reliability modeling for defect detection and removal, assuming perfect and imperfect debugging. Zhang et al. [31] incorporated testing effort functions into finite server queueing defect detection and removal models. Kapur et al. [32] developed an M*/G/8 infinite server queue, where the mean time between defect discovery and removal varies according to defect severity. Later, Huang and Kuo [33] proposed an extended finite server M/M/c queueing model to address limited testing resources. Tokuno et al. [34] developed performability measures for models based on infinite server queuing to quantify the capacity of a software process to complete tasks within a time limit. Okamura and Dohi [35] proposed a generalized bivariate fault detection and correction process, a model with hyper-Erlang distributions, and expectation maximization algorithms to estimate model parameters.

Early covariate models for software reliability include the work of Khoshgoftaar et al. [36-38], who applied alternative estimation techniques for applied nonlinear regression with software metrics [39] as explanatory variables to predict the number of faults in a program module. Evanco and Lacovara [40] presented multiple regression models based on ordinary least

squares regression, Poisson, binomial, ordered response, and proportional hazards models for software reliability and subsequently [41] integrated the Poisson regression into a modified form of the Goel- Okumoto model [42]. Cid and Achcar [43] presented a Bayesian approach to the superposition of several independent NHPP in the presence of covariates. Ray et al. [44] developed a software reliability model for covariates based on hierarchical Bayesian methods. Gandy and Jensen [45] proposed a non-parametric software reliability model based on a multivariate counting process with additive intensity, incorporating covariates for open-source software composed of multiple sub-projects. Ishii et al. [46] used a bivariate NHPP SRGM to characterize defect discovery as a function of software testing activities such as calendar time, the number of test cases executed, and test execution time, while Kapur et al. [47] developed a bivariate NHPP SRGM including testing time and coverage metrics according to the Cobb-Douglas production function. Rinsaka et al. [11] combined the proportional hazards model and NHPP to produce a generalized defect detection process enabling a time-dependent covariate structure. Shibata et al. [4] extended this model to a cumulative Bernoulli trial process. Okamura et al. [48,49] proposed an SRGM for multiple test metrics based on logistic regression and a parameter estimation method based on logistic regression and the expectation-maximization algorithm. The logit and Cox proportional hazards [51] covariate models were generalized by Kuwa and Dohi [50]. Okamura and Dohi [52] extended a covariate model integrating Poisson regression and the non-homogeneous Poisson process to accommodate time series data on defect detection and software metrics of multiple modules. A unified model combining NHPP SRGM and generalized linear models was proposed by Okamura and Dohi [53] to encompass several common SRGM and the logistic and Poisson regression covariate SRGM. Wiper et al. [54] presented a neural network regression method incorporating covariates composed of software metrics to estimate inter-failure times or the number of failures with a Bayesian approach. Torrado et al. [55] developed a semi-parametric Bayesian model incorporating Gaussian processes to estimate software failures in successive time intervals, assuming that the software updates are performed after each interval and that information on software metrics is available. Nagaraju et al. [56] presented NHPP SRGM incorporating covariates based on the discrete Cox proportional hazards model and formulated the optimal test activity allocation problem to maximize defect discovery.

In contrast to past studies, this paper is the first to apply a covariate model to software defect resolution. The predictive and computational performance is compared with (i) the most common alternative, namely an integrated defect discovery and resolution process model [3], and (ii) a distributional approach [12], which demonstrated better predictive and computational performance than the integrated defect discovery and resolution process modeling approach. While queuing models offer additional rigor and possess intuitive appeal, this class of models was not considered because the data did not exhibit statistical evidence that either a priority or mixed queueing discipline was employed by the project to handle defects of low, medium, and high severity. Moreover, it was not possible to speak with members of the original project team to understand the queuing discipline employed during project execution or other relevant factors to construct a satisfactory model. Nevertheless, this paper is one of the few considering defect resolution by severity. It is also the first to apply covariate models to the resolution of defects of each severity level.

3. Software Defect Discovery and Resolution: Models Incorporating Severity

This section describes three methods to model the software defect resolution process, including (i) the integrated defect discovery and resolution process modeling framework of Lo and Huang [3], (ii) the distributional approach developed by Nafreen et al. [12], and (iii) the NHPP SRGM incorporating covariates based on the discrete Cox proportional hazards model of Nagaraju et al. [56], which is adapted to characterize defect resolution. Each model requires a pair of time series indicating the time defects were discovered and resolved, which can be unambiguously determined from a defect tracking database. Defect resolution models enable insights for software practitioners that can be made during the software testing process, including the average amount of time required to remove all defects of a specified severity discovered up to (i) the present time t and (ii) present time t as well as all additional defects anticipated to be discovered.

3.1 Integrated defect discovery and resolution processes

The integrated defect discovery and resolution processes modeling framework [3] expresses the defect discovery and resolution rates as a system of differential equations possessing the below form

$$\frac{dm(t)}{dt} = \lambda(t)\big(\omega - m(t)\big) \tag{1}$$

$$\frac{dm_r(t)}{dt} = \lambda_r(t)\big(m(t) - m_r(t)\big) \tag{2}$$

where $m(t)$ ($m_r(t)$) denotes the mean value function of the number of defects discovered (resolved) by time t, $\lambda(t)$ ($\lambda_r(t)$) the defect discovery (resolution) rate, and $\omega > 0$ is the number of defects that would be discovered and resolved with indefinite testing. Therefore, Equation (2) expresses the instantaneous rate of change in defect

resolution as the product of the defect resolution intensity multiplied by the number of defects discovered but unresolved at time t.

Non-homogeneous Poisson process models were applied to the defect discovery data because failure times were the only information recorded in the defect tracking database, not the testing effort or underlying activities. Analysis of the NASA data set [13] employed in this study found that among the Jelinski-Moranda [1], inflection S-shaped [15], Yamada Delayed S-shaped [16], Goel-Okumoto [42], Weibull [57], and Geometric model [58], the inflection S-shaped model characterized the overall defect discovery process and the discovery process of high, medium, and low severity defects best.

Thus, the mean value function for defect discovery is

$$m(t) = \omega \frac{1-e^{-bt}}{1+ce^{-bt}} \quad (3)$$

where b is a constant defect discovery rate, c the inflection

$$c = \frac{1-r}{r}, \quad r \in (0,1] \quad (4)$$

and r is the inflection rate.

Assuming defect resolution intensity $\lambda_r(t) = b$, identical to parameter b of Equation (3), solution of Equation (2) produces the MVF of the defect resolution process

$$m_r^b(t) = \omega\left(1-e^{-bt} + (1+c)\log\left(\frac{1+c}{c+e^{bt}}\right)e^{-bt}\right) \quad (5)$$

The log-likelihood function of defect resolution times data is

$$LL(\theta,\omega) = -m_r^b(t_n^r + x) + \sum_{i=1}^{n}\log\left(\lambda_r^b(t_i^r)\right) \quad (6)$$

where $m_r^b(t_n^r + x)$ is the MVF of the defect resolution process evaluated at the time at which the last defect was resolved (t_n^r) plus any additional time (x) [59] since this most recent defect was resolved, and

$$\lambda_r^b(t) = \frac{dm_r^b(t)}{dt} \quad (7)$$

The defect resolution process is fit by substituting Equations (5) and (7) as well as the resolution data directly into Equation (6) and maximizing.

3.2 Distributional approach

The distributional approach [12] fits the time distribution between discovery and resolution of each defect. Online application of the distributional approach uses defect discovery and resolution data available up to time t. Since defect resolution is not immediate, the number of defects discovered may be strictly greater than the number of defects resolved. Maximum likelihood estimation techniques for censored data are employed with the following likelihood function to enable an unbiased estimate of the time to resolve defects.

$$Lik(\Theta \mid T) = \prod_{i\in\mathcal{R}}^{k} f\left(t_{(i)};\theta\right) \times \prod_{i\in\mathcal{D}}^{n-k} 1 - F(T-t_i;\theta) \quad (8)$$

where k is the number of defects in the set $\mathcal{R}$ that were discovered and resolved before time T, $(n-k)$ the number of defects in set $\mathcal{D}$ discovered by time T but unresolved, and θ the parameters of the distribution being fit to the data. Moreover, $t_{(i)}$ denotes the time between discovery t_i and resolution t_i^r of the i^{th} defect so that $t_{(i)} = t_i^r - t_i$, whereas $1 - F(T-t_i)$ is the probability that a defect discovered at time t_i is still unresolved at time T.

Given defect discovery and resolution data up until time t, Equation (8) is maximized with multiple alternative distributions and the one exhibiting the best fit is used to make predictions. The mean of this distribution is interpreted as the mean time to resolution (MTTR). Therefore, the MVF of the number of defects resolved by time t may be expressed as

$$m_r(t) = m(t - E[\hat{T}_r]) \quad (9)$$

which shifts the mean value function of the defect discovery process to the right by the mean time to resolve defects ($E[\hat{T}_r]$).

3.3 Covariate approach

The discrete Cox proportional hazard NHPP SRGM [56] links m covariates to the number of events in each of n successive intervals. In modeling defect resolution, these covariates can be (i) defect resolution activities or (ii) metrics related to defect resolution, such as the number of defects of each severity unresolved by the end of interval i. The covariates are denoted $\mathbf{x}_{n\times m}$, where $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$ corresponds to the duration each activity was performed or the value of the metrics in interval i.

The MVF of defects resolved through interval n is

$$m_r(\mathbf{x}) = \omega \sum_{i=1}^{n} p_{i,\mathbf{x}_i} \quad (10)$$

where the probability a defect is resolved in interval i, after going unresolved in the first $(i-1)$ intervals is as:

$$p_{i,\mathbf{x}_i} = \left(1 - \left(1-h(i)\right)^{g(\mathbf{x}_i)} \times \prod_{k=1}^{i-1}\left(1 - h(k)\right)^{g(\mathbf{x}_k)}\right. \quad (11)$$

$h(\cdot)$ is baseline hazard function, and

$$g(\mathbf{x}_i;\beta) = \exp(\beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_m x_{im}) \quad (12)$$

Three examples of hazard functions are the Geometric: $\lambda(k) = b$, Negative binomial of order two: $\lambda(k) = \frac{kb^2}{1+b(k-1)}$, and Discrete Weibull of order two: $\lambda(k) = 1 - b^{k^2-(k-1)^2}$, where $b \in (0,1)$ in all three cases.

The discrete Cox proportional hazard NHPP SRGM possesses the following log-likelihood function:

$$LL(\theta,\beta,\omega) = -\omega\sum_{i=1}^{n} p_{i,\mathbf{x}_i} + \sum_{i=1}^{n} y_i \ln(\omega) + \sum_{i=1}^{n} y_i \ln\left(p_{i,\mathbf{x}_i}\right) - \sum_{i=1}^{n} \ln(y_i!) \quad (13)$$

where $\mathbf{y}_n$ is the number of defects resolved in each of the n intervals and y_i is the number of defects resolved in interval i. Given covariates $\mathbf{x}$ and defect resolution vector $\mathbf{y}$, model fitting identifies the numerical estimates $\widehat{\omega}$ (the total number of defects to be resolved), $\widehat{\beta}$ (vector of covariate coefficients), and $\hat{b}$ (hazard function parameter).

4. Model Assessment

Model assessment quantifies how well a model performs on a data set. This section provides a self-contained summary of complementary measures, including the sum of squares error [60] and the predictive sum of squares error [61], as well as the Akaike information criterion [62] and Bayesian information criterion [63]. Lower values of these measures are preferred. Ideally, a single model performs best on all measures. However, this rarely occurs in practice. If no single model outperforms all of its competitors on all measures model selection becomes subjective based on expert experience and factors such as the amount of data available, testing stage, and predictive horizon.

4.1 Sum of squares error (SSE)

SSE measures the disagreement between the empirical defect resolution time data and estimates of a defect resolution model.

$$SSE = \sum_{i=1}^{n}\left(N_r(i) - \widehat{m}(i)\right)^2 \quad (14)$$

where $N_r(i)$ is the number of defects resolved by time t_i or the end of interval i and $\widehat{m}_r(i)$ is the model estimate of the number of defects resolved.

4.2 Predictive sum of squares error (PSSE)

PSSE fits a model to the first $k < n$ failure times or intervals and computes the disagreement between empirical data and model estimates on the remaining $n - k$ times or intervals not used to perform model fitting.

$$PSSE = \sum_{i=k+1}^{n}\left(N_r(i) - \widehat{m}(i)\right)^2 \quad (15)$$

SSE is the special case where $k = 0$.

4.3 Akaike Information Criterion (AIC)

The Akaike Information Criterion is an information-theoretic measure of goodness of- fit, which is based on the concept of entropy, measuring the information lost when a model is applied. The AIC quantifies the tradeoff between a model's complexity and characterization of the observed data. The AIC of model i is a function of the number of model parameters (p) and maximized log-likelihood.

$$AIC_i = 2p - 2LL(\widehat{\Theta} \mid T) \quad (16)$$

4.4 Bayesian information criterion (BIC)

BIC is similar to AIC, but the penalty term also includes the sample size (n). The BIC of model i is

$$BIC_i = p\log(n) - 2LL(\widehat{\Theta} \mid T) \quad (17)$$

5. Illustrations

This section compares the alternative defect discovery and resolution models described in Section 3 according to the model assessment techniques given in Section 4 and their computational performance. Section 5.1 describes the data extracted from a NASA defect tracking database to which the models were applied. Section 5.2 conducts a retrospective analysis common in historical defect discovery modeling papers using all available data. Next, we describe the steps of the distributional approach to determine the mean time to resolution, and the covariates and interval width were selected for the covariate approach. Tradeoffs between the goodness of fit and runtime posed by the interval width of the covariate approach are also examined. Section 5.3 performs a novel analysis based on the defect and resolution discovery processes, namely an analysis of the open defects for progress tracking throughout the testing process. Section 5.4 assesses predictive accuracy. Each analysis is performed on the entire dataset consisting of defects of all three severities and analyses broken down by low, medium, and high severity.

5.1 Data Description

The dataset [13] considered was extracted from a defect-tracking database employed by NASA on a major project spanning multiple years. After cleaning, the database consisted of $n = 455$ rows, each of which corresponded to a defect, including low ($n_3 = 61$), medium ($n_2 = 381$), and high severity ($n_1 = 13$) defects. The database also documented when the defect entered any of the 13 possible states from discovery to resolution. However, a recent study [12] demonstrated that the sample size was too small to apply a semi-Markov process model of the defect tracking process. Therefore, for this study, we created two-time series, one composed of the time each defect was discovered and a second for the time each defect was resolved. Unlike past defect discovery and resolution modeling efforts, the information contained in rows of the database also allowed us to compute the individual times between resolution and discovery, enabling the application of the distributional and covariate approaches.

5.2 Retrospective Analysis

Figure 1 shows $N(t)$ and $N_r(t)$ (the discovery and resolution counting processes) and the corresponding

discovery and resolution model fit the three approaches described in Section 3.

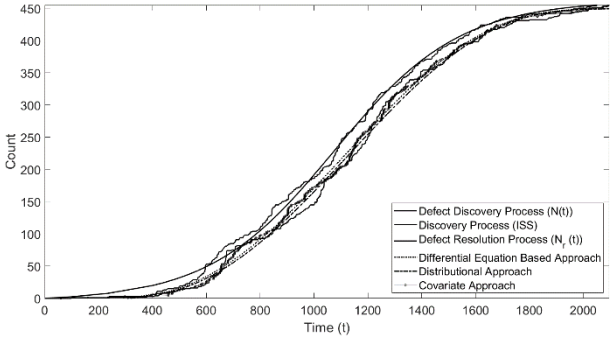


Figure 1. Defect discovery and resolution processes with models fitted to all data

The first counting process (left step function) indicates when the defects were discovered. The fit of the inflection S-shaped model (Equation (3)) is also shown. The second counting process (right step function) indicates when defects were resolved. The fits of the differential equation-based (Equation (5)), distributional (Equation (9)), and covariate model (Equation (10)) are also shown.

Intuition suggests that information on unresolved defects of all severities would be needed to accurately characterize the overall defect resolution process consisting of low, medium, and high-severity defects. Therefore, the defect resolution data was discretized into intervals of 20-time units for the covariate model. The covariates of the i^{th} interval

$$\mathbf{x}_i = < \mathbf{x}_{i,low}, \mathbf{x}_{i,medium}, \mathbf{x}_{i,high} >$$

were determined as the number of defects of a given severity discovered by the beginning of interval i but not yet resolved and y_i was the number of defects of any severity resolved in that interval. Based on this division of resolution time data into intervals of 20-time units, the covariate model with Discrete Weibull hazard rate and parameters

$$\hat{\beta}_{Low} = 0.0202, \hat{\beta}_{Medium} = 0.0519, \hat{\beta}_{High} = 0.1003$$

achieved the best fit, suggesting that high-severity defects contributed most to defect resolution, followed by medium and low-severity defects. Thus, while medium severity defects were the most common, followed by low and high severity, respectively, high severity defects most significantly influenced the defect resolution process as characterized by the covariate model parameters. Careful examination of the data indicated that defects were often resolved in batches. Therefore, one plausible explanation for the numerical parameters is that low and medium-severity defects were often resolved when high-severity defects were resolved. Still, high-severity defects drove the defect resolution planning process.

Table 1 summarizes the performance of each model concerning the goodness of fit measures described in Section 4.

Table 1. Comparison of defect resolution models

Approach	SSE	PSSE (90%)	AIC	BIC	Runtime (s)
D-E	3.77×10^4	2.99×10^3	3.54×10^4	3.83×10^4	175.2
Distributional	3.43×10^4	1.68×10^3	3.32×10^4	3.64×10^4	150.1
Covariate	$\mathbf{1.72 \times 10^3}$	$\mathbf{7.94 \times 10^2}$	$\mathbf{1.29 \times 10^3}$	$\mathbf{1.45 \times 10^3}$	**110.2**

Values in bold indicate the best model concerning each measure. Specifically, the discrete Cox proportional hazard model with covariates denoting the number of defects of high, medium, and low severity discovered but unresolved achieved an order of magnitude lower sum of squares error, PSSE, AIC, and BIC and required less time to apply.

5.2.1 Distributional Approach:

The mean time to resolution used to plot the mean value function of the defect resolution curve for the distributional approach in Figure 1 ($E[\hat{T}_r] = 59.74$) was determined with the special case of Equation (8), composed of all discovery and resolution times for high, medium, and low severity defects through the time at which the n^{th} defect was resolved. Seventeen possible distributions were considered, including the Beta, Birnbaum- Saunders, Exponential, Extreme value, Gamma, Generalized extreme value, Generalized Pareto, Inverse Gaussian, Logistic, Log-logistic, Lognormal, Nakagami, Normal, Rayleigh, Rician, t location-scale, and Weibull distributions. The generalized extreme value (GEV) distribution possessing the following maximum likelihood estimates

$$\hat{T}^r \sim GEV(\mu = 57.4437, \sigma = 22.6722, \xi = -0.0959)$$

exhibited the best fit for the times between defect discovery and resolution for the Akaike and Bayesian Information Criterion.

Figure 2 shows the fitted Generalized extreme value distribution and histogram of times between defect discovery and resolution. Thus, the plot of the distributional approach is simply the inflexion S-shaped model that was fit to the defect discovery data given by equation (3) shifted to the right by the mean time to resolution ($E[\hat{T}_r]$), as defined in Equation (9).

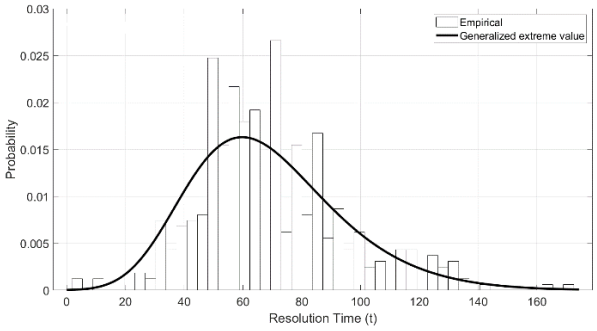


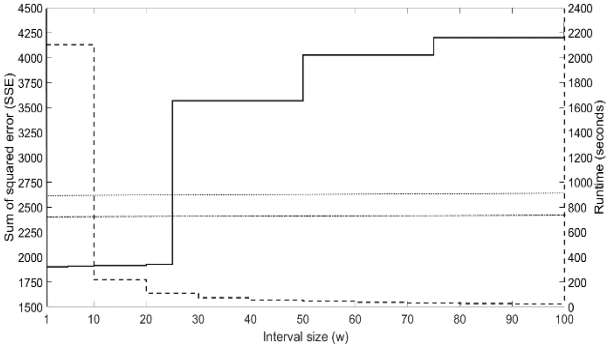
Figure 2. Empirical distribution of time between defect discovery and resolution of all defects

5.2.2 Covariate data analysis methodology:

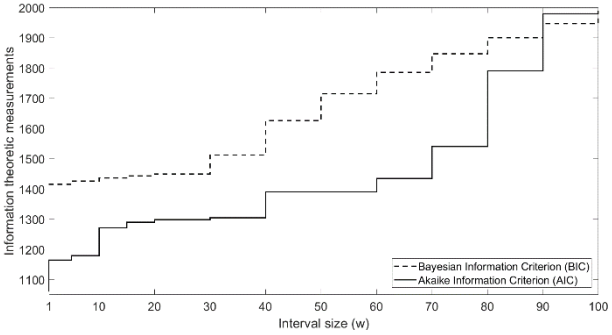
Dividing resolution time data into 20-unit intervals for the covariate model considered in Figure 1 and Table 1 is arbitrary. Thus, to assess interval width on the accuracy of the covariate model, Figure 3a shows the SSE of predictions made with widths $w \in \{5,10, \ldots,100\}$ as well as intervals of width one. Figure 3a indicates that a width 25 or less performed best and that decreasing w from five to one did not significantly decrease the SSE. Moreover, since the time of the last resolution was $t_n^r \approx 2096$, there were 21 intervals for $w = 100$, and 2096 intervals for $w = 1$. In cases where t_n^r/w was not an integer; the width of the final interval was reduced. For example, the width of the final interval was 96 when $w = 100$. Since this was the final interval and most defects had been discovered and resolved, this modest amount of non-uniformity in the width of the intervals did not impact the model fit substantially.

Figure 3a also shows that intervals of width $15 \leq w \leq 25$ did not substantially increase the time required to fit the model to achieve an order of magnitude improvement in model fit over the alternative approaches, whereas the differential equation-based and distributional approaches required 886.2 and 749.5 seconds to apply respectively and the SSE was an order of magnitude worse as noted in Table 1. To demonstrate that smaller intervals are also justified from an information theoretical standpoint, Figure 3b shows the impact of interval width on BIC and AIC.

Since smaller values are preferred, intervals of unit width achieved the best fit. Nevertheless, intervals of width 20 may be adequate since the relative error between the BIC of models with widths 20 and 1 was just 0.0251 ((1450.1-1414.6)/1414.6) and the relative error in the AIC of models with widths twenty and one was 0.2169 ((1290.1- 1060.1)/1060.1). While the relative error of the AIC is not as small as the BIC's, it should be noted that AIC does not penalize the sample size. Thus, BIC may be a better measure of goodness of fit because decreasing w from twenty to one increases the sample size by a factor of twenty. The slight decrease in BIC from twenty to one suggests that the penalty associated with this increase in sample nullifies most gains in maximum likelihood attained.



(a)



(b)

Figure 3. (a) Impact of interval width on SSE and time required to fit Covariate models **(b)** Impact of interval width on BIC and AIC of Covariate models

5.2.3 Retrospective analysis by severity:

Since the response variable of the covariate model is the vector of defects $\mathbf{y}_n$ resolved in each of the n intervals, the model is not capable of simultaneously predicting the number of defects of low, medium, and high severity defects resolved in each interval with Equation (13). However, it is straightforward to apply Equation (13) with the three covariates $\mathbf{x}_i$ as inputs and $\mathbf{y}_n^s$, the vector of defects resolved in each of the n intervals, where $\mathbf{y}_i^s$ is the number of defects of severity $s \in \{1,2,3\}$ resolved in interval i as the response.

Table 2 summarizes the performance of each resolution model for low, medium, and high-severity defects, respectively.

Table 2. Comparison of defect resolution models by severity

Approach	S	SSE	PSSE (90%)	AIC	BIC	Runtime (s)
D-E Based Distributional Covariate	3	5.12×10^3 4.67×10^3 $\mathbf{9.16 \times 10^2}$	6.02×10^2 4.23×10^2 $\mathbf{1.62 \times 10^1}$	4.82×10^3 4.52×10^3 $\mathbf{1.76 \times 10^2}$	5.22×10^3 4.96×10^3 $\mathbf{1.97 \times 10^2}$	24.8 20.5 **14.1**
D-E Based Distributional Covariate	2	3.10×10^4 2.47×10^4 $\mathbf{5.73 \times 10^3}$	6.29×10^3 5.26×10^3 $\mathbf{5.54 \times 10^2}$	2.94×10^4 2.78×10^4 $\mathbf{1.09 \times 10^3}$	3.23×10^4 3.04×10^4 $\mathbf{1.25 \times 10^3}$	145.6 124.7 **91.3**
D-E Based Distributional Covariate	1	1.07×10^3 9.21×10^2 $\mathbf{1.72 \times 10^2}$	3.86×10^2 5.12×10^2 $\mathbf{9.08 \times 10^1}$	1.04×10^3 9.32×10^2 $\mathbf{3.59 \times 10^1}$	1.09×10^3 1.04×10^3 $\mathbf{4.45 \times 10^1}$	5.2 4.1 **3.3**

Values in bold indicate the model that performed best concerning each measure. Table 2 indicates that the discrete Cox proportional hazard model with covariates and second-order Discrete Weibull hazard function outperformed the alternatives by an order of magnitude for each of the three severities and exhibited better runtime performance.

Figure 4 shows the generalized Pareto, generalized extreme value, and exponential distributions that best fit the times between discovery and resolution of low, medium, and high severity defects according to the BIC

and AIC and the corresponding histogram of times between defect discovery and resolution. The primary observation derived from these fitted distributions was that the mean time to resolution of low, medium, and high severity defects were 70.27, 66.19, and 54.76 days, indicating that higher severity defects were resolved more quickly. This suggests that, on average, higher-severity defects received more attention. Efforts to document the effort dedicated to resolving these defects will further enhance the applicability of the covariate approach.

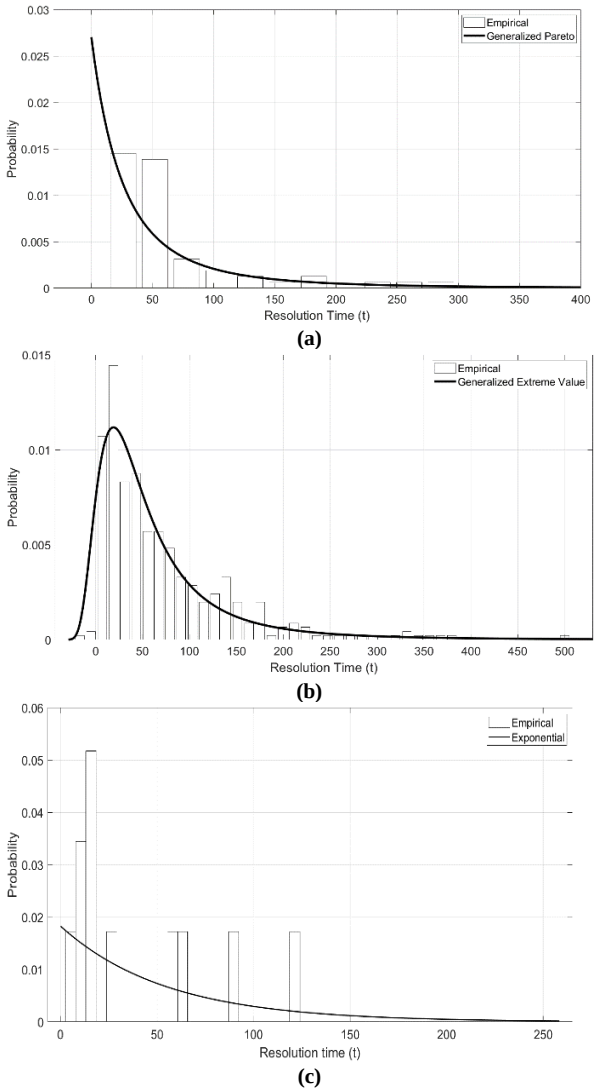


Figure 4. (a) Empirical distribution of time between discovery and resolution of low severity defects **(b)** Empirical distribution of time between discovery and resolution of medium severity defects **(c)** Empirical distribution of time between discovery and resolution of high-severity defects

5.3 Analysis of unresolved defects

To provide an alternative perspective, Figure 5 shows the number of unresolved defects at the end of each interval and the predictions of the fitted models. Increases (decreases) in the step function indicate times at which defects were discovered (resolved) more quickly than they were resolved (discovered). Thus, the value on the y-axis represents the difference between the number of defects discovered by time t ($N(t)$) and the number of defects resolved by time t ($N_r(t)$). At the peak ($t = 1200$), nearly 50 defects were unresolved.

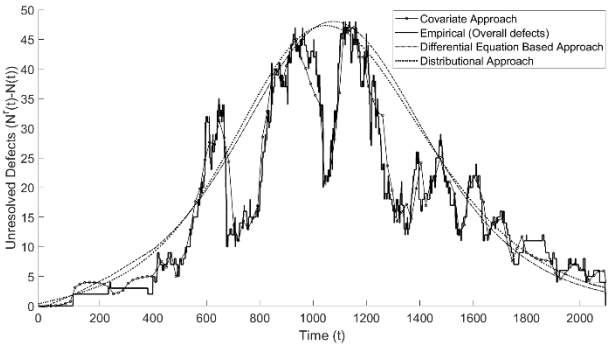


Figure 5. Open defects (discovered but not yet resolved) and fitted models

Figure 5 indicates that the differential equation-based and distributional approaches only capture primary trends according to their parametric form, whereas the covariate approach tracks the unresolved defects remarkably well. The number of defects discovered but not resolved at time t or interval i was computed by subtracting the MVF of the defect discovery process ($\widehat{m}(t)$) of Equation (3) from the fitted MVF of the defect resolution process ($\widehat{m}_r^b(t)$ or $m_r(x)$) from Equation (5), (9) or Equation (3) for the differential equation-based, distributional, and covariate approach, respectively.

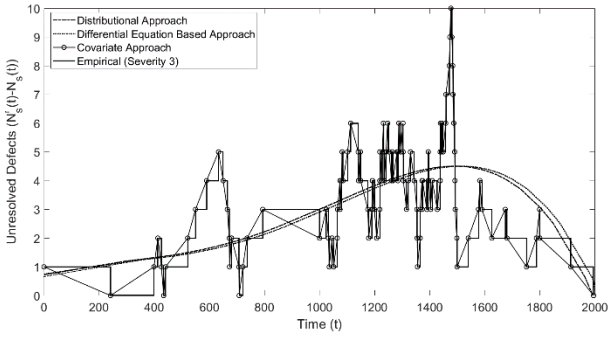
Table 3 summarizes the model assessments for unresolved defects of any severity, indicating the covariate approach outperforms the differential equation-based and distributional approaches by an order of magnitude, reflecting the superior fit of the covariate model fit to the number of unresolved defects in each interval observed in Figure 5. To fairly compare continuous and discrete models, SSE and PSSE were computed at the end of each discrete interval to avoid favoring the discrete model when the number of intervals was smaller than the number of defects resolved.

Table 3. Comparison of models on unresolved defects

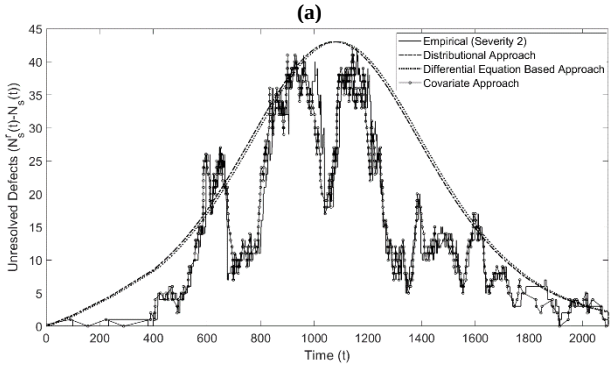
Approach	SSE	PSSE (90%)	AIC	BIC	Runtime (s)
D-E	5.28×10^4	2.10×10^2	7.10×10^4	7.23×10^4	330.2
Distributional	3.03×10^4	2.10×10^2	7.05×10^4	7.06×10^4	312.1
Covariate	$\mathbf{1.92 \times 10^3}$	$\mathbf{4.59 \times 10^1}$	$\mathbf{3.59 \times 10^3}$	$\mathbf{3.71 \times 10^3}$	**186.1**

5.3.1 Analysis of unresolved defects by severity:

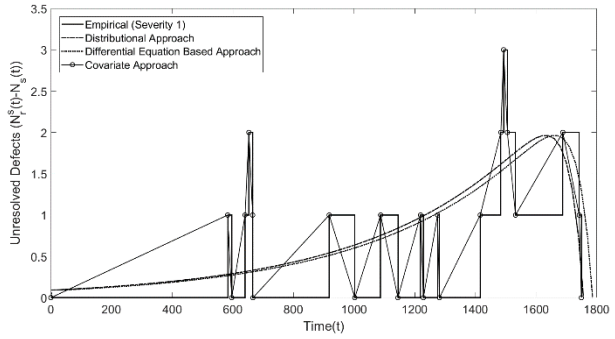
Figure 6 shows the number of unresolved defects of low, medium, and high severity defects and corresponding model fits. In each case, differential equation-based and distributional approaches only capture a single peak in the trend. Still, the covariate approach tracks the number of open defects extremely well, even the less frequent low and high-severity defects.



(a)



(b)



(c)

Figure 6. (a) Open defects of low severity ($s = 3$) and fitted models, **(b)** Open defects of medium severity ($s = 2$) and fitted models, **(c)** Open defects of high severity ($s = 1$) and fitted models

Table 4 summarizes the model assessments for low, medium, and high-severity unresolved defects.

Table 4. Comparison of models on unresolved defects by severity

Approach	S	SSE	PSSE (90%)	AIC	BIC	Runtime (s)
D-E Based	3	3.10×10^3	3.43×10^2	3.87×10^3	4.12×10^3	45.2
Distributional		2.62×10^3	1.58×10^2	3.31×10^3	3.67×10^3	42.8
Covariate		$\mathbf{1.71 \times 10^2}$	$\mathbf{1.01 \times 10^1}$	$\mathbf{4.61 \times 10^2}$	$\mathbf{4.85 \times 10^2}$	**24.8**
D-E Based	2	3.82×10^4	2.66×10^3	2.81×10^4	2.99×10^4	277.3
Distributional		2.11×10^4	1.09×10^3	2.74×10^4	2.91×10^4	262.1
Covariate		$\mathbf{2.66 \times 10^3}$	$\mathbf{1.11 \times 10^2}$	$\mathbf{2.85 \times 10^3}$	$\mathbf{2.93 \times 10^3}$	**156.7**
D-E Based	1	1.79×10^2	1.85×10^1	1.86×10^3	2.01×10^2	9.6
Distributional		1.49×10^2	5.84×10^1	$\mathbf{1.15 \times 10^2}$	$\mathbf{1.54 \times 10^2}$	10.9
Covariate		$\mathbf{2.27 \times 10^1}$	$\mathbf{1.43 \times 10^1}$	5.71×10^2	5.98×10^2	**5.3**

Once again, the covariate approach performed best on virtually all measures of goodness of fit and required less time to apply. However, the distributional approach performed best on high severity ($s = 1$) possibly because of the low sample size.

5.4 Assessment of Predictive Accuracy

Ideally, a model should be simple and accurately predict the distant future with little data. To compare the predictive accuracy of the defect resolution models, this section performs an online assessment of the models with the predictive SSE measure. The defect resolution processes of the differential equation-based and covariate approaches (Equations (5) and (10) respectively) were fit to the resolution time data extracted from the defect tracking database, whereas the distributional approach identified an SRGM that fit the available defect discovery data best, estimated the MTTR with Equation (8), and then substituted the MTTR into Equation (9). The PSSE was subsequently computed according to Equation (15) as the sum of squares difference between the actual number of defects observed and model predictions at each resolution time t_i^r or each interval i. For the sake of comparison, the amount of data provided for defect discovery and resolution model fitting was performed in increments of 20, the width of the intervals in the covariate approach.

Figure 7 shows the online assessment of the defect resolution models with PSSE for defects of all three severities combined. Times before $t = 200$ are excluded, so primary trends are distinguishable since PSSE was initially very large and would skew the remainder of the graph.

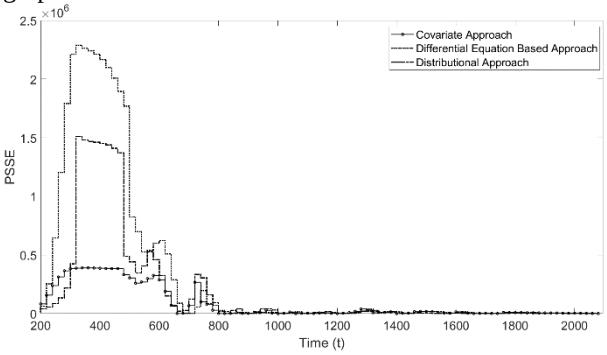


Figure 7. PSSE of models on defects of three severities combined

Figure 7 indicates that the covariate approach exhibited substantially lower error than the alternatives and sustained the highest accuracy throughout the remainder of the defect discovery and resolution process. The covariate approach is accurate because of the availability of information on the number of open defects by severity. In contrast, the distributional approach exhibits error since few of the times between defect discovery and resolution shown in Figure 2 were observed before $t = 600$. Predictions of the differential

equation-based approach were the worst because parametric models implicitly make rigid assumptions about the shape of the defect resolution curve.

Figure 8 shows the online assessment results of the defect resolution models with PSSE by severity.

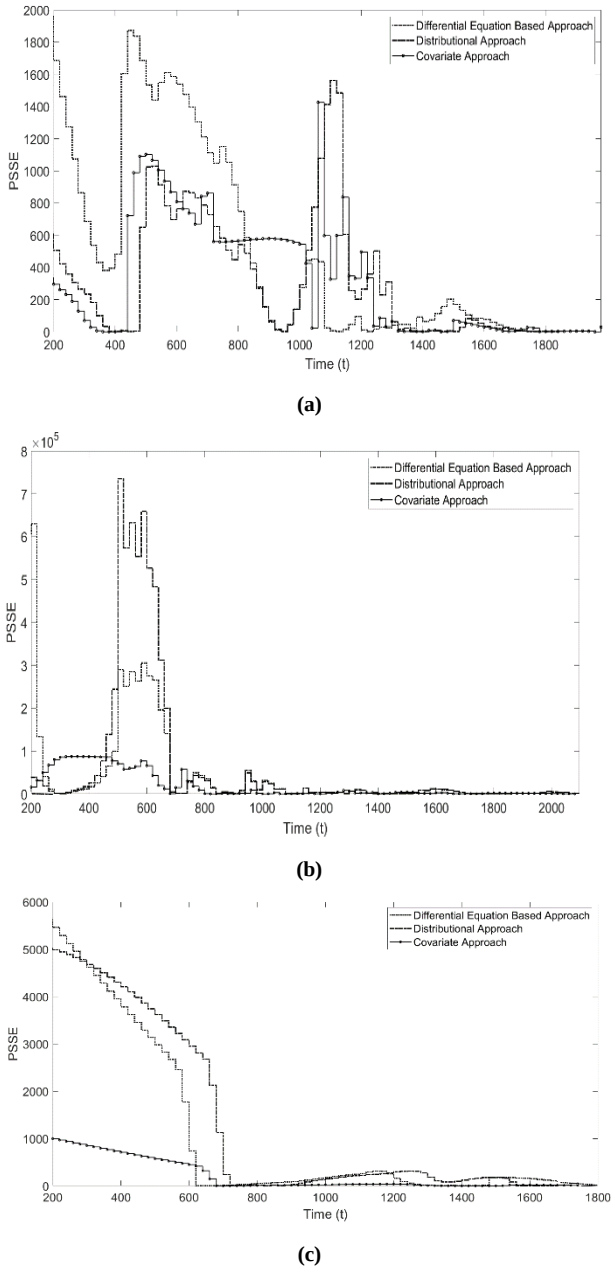


Figure 8. (a) PSSE of models on low-severity defects, **(b)** PSSE of models on medium-severity defects, **(c)** PSSE of models on high-severity defects

Figure 8a indicates that the distributional and covariate approaches predicted approximately the same prior to $t = 800$ for low-severity defects. In contrast, Figures 8b and 8c, respectively, show that the predictions of the covariate approach were best for medium severity defects at times $t > 400$ and at all times for high severity defects. The prediction errors may be partly explained by the sample size of low and high-severity defects, the parametric forms for the differential equation-based and distributional models, and undocumented factors within the software test process. The excellent predictive accuracy of the covariate approach on high-severity defects is promising. However, disciplined collection of covariates related to defect resolution efforts could substantially reduce prediction errors, supporting risk mitigation efforts to ensure high-severity defects are removed prior to fielding.

6. Conclusion and Future Research

This paper presented a model for the number of defects detected and resolved according to the discrete Cox proportional hazard model incorporating covariates describing metrics or activities that could serve as predictors. Defect resolution activities and the amount of effort dedicated to each were not explicitly documented in the NASA defect tracking database. So, the number of low, medium, and high-severity unresolved defects were used as covariates. The illustrations showed that the covariate approach outperformed other models by an order of magnitude on all goodness of fit measures considered and required less time to apply, exhibiting similar performance when applied to subsets of data for low, medium, and high severity defects. A similar analysis of the number of unresolved defects demonstrated compelling evidence that the covariate approach tracked the data much better than the alternative approaches. Finally, the covariate approach exhibited low predictive error, even when only 10- 20% of testing had elapsed.

Future research will seek to improve the efficiency of the model fitting procedure for the covariate approach when (i) the data consists of a large number of intervals and (ii) the number of covariates describing the effort allocated to distinct defect resolution activities in each interval is large.

Acknowledgments

This material is based upon work supported by the National Aeronautics and Space Administration under Grant Number 80NSSC20K0276 and the U.S. National Science Foundation under Grant Number 1749635. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the official policy or position of the National Aeronautics and Space Administration or the U.S. National Science Foundation.

Conflict of Interests

No conflict of interest has been expressed by the authors.

7. References

[1] Z. Jelinski and P. Moranda, "SOFTWARE RELIABILITY RESEARCH," in *Statistical Computer Performance Evaluation*, W. Freiberger Ed.: Academic Press, 1972, pp. 465-484.

[2] N. F. Schneidewind, "Analysis of error processes in computer software," in *Proceedings of the international conference on Reliable software*, 1975, pp. 337-346, doi: https://doi.org/10.1145/800027.808456.

[3] J.-H. Lo and C.-Y. Huang, "An integration of fault detection and correction processes in software reliability analysis," *Journal of Systems and Software*, vol. 79, no. 9, pp. 1312-1323, 2006/09/01/ 2006, doi: https://doi.org/10.1016/j.jss.2005.12.006.

[4] K. Shibata, K. Rinsaka, and T. Dohi, "Metrics-Based Software Reliability Models Using Non-homogeneous Poisson Processes," in *2006 17th International Symposium on Software Reliability Engineering*, 7-10 Nov. 2006 2006, pp. 52-61, doi: https://doi.org/10.1109/ISSRE.2006.28.

[5] T. Dohi, T. Matsuoka, and S. Osaki, "An infinite server queuing model for assessment of the software reliability," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 85, no. 3, pp. 43-51, 2002/03/01 2002, doi: https://doi.org/10.1002/ecjc.1078.

[6] T. Dohi, S. Osaki, and K. S. Trivedi, "An infinite server queueing approach for describing software reliability growth: unified modeling and estimation framework," in *11th Asia-Pacific Software Engineering Conference*, 30 Nov.-3 Dec. 2004 2004, pp. 110-119, doi: https://doi.org/10.1109/APSEC.2004.29.

[7] S. S. Gokhale and R. E. Mullen, "Queuing models for field defect resolution process," in *2006 17th International Symposium on Software Reliability Engineering*, 2006: IEEE, pp. 353-362, doi: https://doi.org/10.1109/ISSRE.2006.38.

[8] P. N. Misra, "Software reliability analysis," *IBM Systems Journal*, vol. 22, no. 3, pp. 262-270, 1983, doi: https://doi.org/10.1147/sj.223.0262.

[9] S. Yamada, S. Osaki, and H. Narihisa, "A software reliability growth model with two types of errors," *RAIRO-Operations Research*, vol. 19, no. 1, pp. 87-104, 1985, doi: https://doi.org/10.1051/ro/1985190100871.

[10] L. Fiondella and S. S. Gokhale, "Software Reliability Models Incorporating Testing Effort," *OPSEARCH*, vol. 45, no. 4, pp. 351-368, 2008/12/01 2008, doi: https://doi.org/10.1007/BF03398825.

[11] K. Rinsaka, K. Shibata, and T. Dohi, "Proportional intensity-based software reliability modeling with time-dependent metrics," in *30th Annual International Computer Software and Applications Conference (COMPSAC'06)*, 2006, vol. 1: IEEE, pp. 369-376, doi: https://doi.org/10.1109/COMPSAC.2006.68.

[12] M. Nafreen, M. Luperon, L. Fiondella, V. Nagaraju, Y. Shi, and T. Wandji, "Connecting software reliability growth models to software defect tracking," in *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, 2020: IEEE, pp. 138-147, doi: https://doi.org/10.1109/ISSRE5003.2020.00022.

[13] H. Sukhwani, J. Alonso, K. S. Trivedi, and I. Mcginnis, "Software reliability analysis of NASA space flight software: A practical experience," in *2016 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, 2016: IEEE, pp. 386-397, doi: https://doi.org/10.1109/QRS.2016.50.

[14] M. Xie and M. Zhao, "The Schneidewind software reliability model revisited," in *Proceedings Third International Symposium on Software Reliability Engineering*, 7-10 October 1992 1992, pp. 184-192, doi: https://doi.ieeecomputersociety.org/10.1109/ISSRE.1992.285846.

[15] M. Ohba, "Inflection S-shaped software reliability growth model," in *Stochastic Models in Reliability Theory: Proceedings of a Symposium Held in Nagoya, Japan, April 23–24, 1984*, 1984: Springer, pp. 144-162, doi: https://doi.org/10.1007/978-3-642-45587-2_10.

[16] S. Yamada, M. Ohba, and S. Osaki, "S-shaped reliability growth modeling for software error detection," *IEEE Transactions on reliability*, vol. 32, no. 5, pp. 475-484, 1983, doi: https://doi.org/10.1109/TR.1983.5221735.

[17] P. Kapur and S. Younes, "Software reliability growth model with error dependency," *Microelectronics Reliability*, vol. 35, no. 2, pp. 273-278, 1995, doi: https://doi.org/10.1016/0026-2714(94)00054-R.

[18] S. Yamada, K. Tokuno, and S. Osaki, "Imperfect debugging models with fault introduction rate for software reliability assessment," *International Journal of Systems Science*, vol. 23, no. 12, pp. 2241-2252, 1992, doi: https://doi.org/10.1080/00207729208949452.

[19] S. S. Gokhale, P. N. Marinos, M. Lyn, and K. S. Trivedi, "Effect of repair policies on software reliability," in *Proceedings of COMPASS'97: 12th Annual Conference on Computer Assurance*, 1997: IEEE, pp. 105-116, doi: https://doi.org/10.1109/CMPASS.1997.613262.

[20] C.-Y. Huang, C.-T. Lin, S.-Y. Kuo, M. R. Lyu, and C.-C. Sue, "Software reliability growth models incorporating fault dependency with various debugging time lags," in *Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.*, 2004: IEEE, pp. 186-191, doi: https://doi.org/10.1109/CMPSAC.2004.1342826.

[21] N. Ullah, M. Morisio, and A. Vetro, "A comparative analysis of software reliability growth models using defects data of closed and open source software," in

2012 35th Annual IEEE Software Engineering Workshop, 2012: IEEE, pp. 187-192, doi: https://doi.org/10.1109/SEW.2012.26.

[22] Y. Liu, M. Xie, J. Yang, and M. Zhao, "A new framework and application of software reliability estimation based on fault detection and correction processes," in *2015 IEEE International Conference on Software Quality, Reliability and Security*, 2015: IEEE, pp. 65-74, doi: https://doi.org/10.1109/QRS.2015.20.

[23] J. Yang, Y. Liu, M. Xie, and M. Zhao, "Modeling and analysis of reliability of multi-release open source software incorporating both fault detection and correction processes," *Journal of Systems and Software*, vol. 115, pp. 102-110, 2016/05/01/ 2016, doi: https://doi.org/10.1016/j.jss.2016.01.025.

[24] M. Cinque, D. Cotroneo, A. Pecchia, R. Pietrantuono, and S. Russo, "Debugging-workflow-aware software reliability growth analysis," *Software Testing, Verification and Reliability*, vol. 27, no. 7, p. e1638, 2017, doi: https://doi.org/10.1002/stvr.1638.

[25] P. Vizarreta *et al.*, "Assessing the maturity of SDN controllers with software reliability growth models," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1090-1104, 2018, doi: https://doi.org/10.1109/TNSM.2018.2848105.

[26] P. Vizarreta, E. Sakic, W. Kellerer, and C. M. Machuca, "Mining software repositories for predictive modelling of defects in sdn controller," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019: IEEE, pp. 80-88.

[27] R. Xie, H. Qiu, Q. Zhai, and R. Peng, "A model of software fault detection and correction processes considering heterogeneous faults," *Quality and Reliability Engineering International*, vol. 39, no. 8, pp. 3428-3444, 2023, doi: https://doi.org/10.1002/qre.3172.

[28] H. Joe, "Statistical inference for general-order-statistics and nonhomogeneous-Poisson-process software reliability models," *IEEE Transactions on Software Engineering*, vol. 15, no. 11, pp. 1485-1490, 1989, doi: https://doi.org/10.1109/32.41340.

[29] C.-T. Lin, C.-Y. Huang, and C.-C. Sue, "Measuring and assessing software reliability growth through simulation-based approaches," in *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, 2007, vol. 1: IEEE, pp. 439-448, doi: https://doi.org/10.1109/COMPSAC.2007.141.

[30] C.-Y. Huang and W.-C. Huang, "Software reliability analysis and measurement using finite and infinite server queueing models," *IEEE Transactions on Reliability*, vol. 57, no. 1, pp. 192-203, 2008.

[31] N. Zhang, G. Cui, and H.-w. Liu, "A finite queuing model with generalized modified Weibull testing effort for software reliability," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, 2011, vol. 1: IEEE, pp. 401-406, doi: https://doi.org/10.1109/ICCSNT.2011.6181985.

[32] P. Kapur, H. Pham, S. Anand, and K. Yadav, "A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation," *IEEE Transactions on Reliability*, vol. 60, no. 1, pp. 331-340, 2011, doi: https://doi.org/10.1109/TR.2010.2103590.

[33] C.-Y. Huang and T.-Y. Kuo, "Queueing-theory-based models for software reliability analysis and management," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 540-550, 2017, doi: https://doi.org/10.1109/TETC.2014.2388454.

[34] K. Tokuno, T. Nagata, and S. Yamada, "Stochastic software performability evaluation based on NHPP reliability growth model," *International Journal of Reliability, Quality and Safety Engineering*, vol. 18, no. 05, pp. 431-444, 2011, doi: https://doi.org/10.1142/S0218539311004172.

[35] H. Okamura and T. Dohi, "A generalized bivariate modeling framework of fault detection and correction processes," in *2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE)*, 2017: IEEE, pp. 35-45, doi: https://doi.org/10.1109/ISSRE.2017.22.

[36] T. M. Khoshgoftaar and J. C. Munson, "Predicting software development errors using software complexity metrics," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 2, pp. 253-261, 1990, doi: https://doi.org/10.1109/49.46879.

[37] T. M. Khoshgoftaar, B. B. Bhattacharyya, and G. D. Richardson, "Predicting software errors, during development, using nonlinear regression models: a comparative study," *IEEE Transactions on Reliability*, vol. 41, no. 3, pp. 390-395, 1992, doi: https://doi.org/10.1109/24.159804.

[38] T. M. Khoshgoftaar, J. C. Munson, B. B. Bhattacharya, and G. D. Richardson, "Predictive modeling techniques of software quality from software measures," *IEEE Transactions on Software Engineering*, vol. 18, no. 11, pp. 979-987, 1992, doi: https://doi.org/10.1109/32.177367.

[39] Y. Shi, M. Li, S. Arndt, and C. Smidts, "Metric-based software reliability prediction approach and its application," *Empirical Software Engineering*, vol. 22, pp. 1579-1633, 2017, doi: https://doi.org/10.1007/s10664-016-9425-9.

[40] W. M. Evanco and R. Lacovara, "A model-based framework for the integration of software metrics," *Journal of Systems and Software*, vol. 26, no. 1, pp. 77-86, 1994/07/01/ 1994, doi: https://doi.org/10.1016/0164-1212(94)90098-1.

[41] W. M. Evanco, "Poisson analyses of defects for small software components," *Journal of Systems and Software*, vol. 38, no. 1, pp. 27-35, 1997/07/01/ 1997, doi: https://doi.org/10.1016/S0164-1212(97)00063-0.

[42] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other

performance measures," *IEEE transactions on Reliability*, vol. 28, no. 3, pp. 206-211, 1979, doi: https://doi.org/10.1109/TR.1979.5220566.

[43] J. E. RamÍRez Cid and J. Alberto Achcar, "Software Reliability Considering the Superposition of Non-homogeneous Poisson Processes in the Presence of a Covariate," *Statistics*, vol. 36, no. 3, pp. 259-269, 2002/01/01 2002, doi: https://doi.org/10.1080/02331880212854.

[44] B. K. Ray, Z. Liu, and N. Ravishanker, "Dynamic reliability models for software using time-dependent covariates," *Technometrics*, vol. 48, no. 1, pp. 1-10, 2006, doi: https://doi.org/10.1198/004017005000000292.

[45] A. Gandy and U. Jensen, "A non-parametric approach to software reliability," *Applied Stochastic Models in Business and Industry*, vol. 20, no. 1, pp. 3-15, 2004, doi: https://doi.org/10.1002/asmb.510.

[46] T. Ishii, T. Fujiwara, and T. Dohi, "Bivariate extension of software reliability modeling with number of test cases," *International Journal of Reliability, Quality and Safety Engineering*, vol. 15, no. 01, pp. 1-17, 2008, doi: https://doi.org/10.1142/S0218539308002897.

[47] P. Kapur, A. G. Aggarwal, and A. Tandon, "Two dimensional software reliability growth model with faults of different severity," *Communications in Dependability and Quality Management*, vol. 13, no. 4, pp. 98-110, 2010.

[48] H. Okamura, Y. Etani, and T. Dohi, "A multi-factor software reliability model based on logistic regression," in *2010 IEEE 21st International Symposium on Software Reliability Engineering*, 2010: IEEE, pp. 31-40, doi: https://doi.org/10.1109/ISSRE.2010.14.

[49] H. Okamura, Y. Etani, and T. Dohi, "Quantifying the effectiveness of testing efforts on software fault detection with a logit software reliability growth model," in *2011 Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement*, 2011: IEEE, pp. 62-68, doi: https://doi.org/10.1109/IWSM-MENSURA.2011.26.

[50] D. Kuwa and T. Dohi, "Generalized Logit Regression-Based Software Reliability Modeling with Metrics Data," in *2013 IEEE 37th Annual Computer Software and Applications Conference*, 2013: IEEE, pp. 246-255, doi: https://doi.org/10.1109/COMPSAC.2013.41.

[51] D. Kuwa, T. Dohi, and H. Okamura, "Generalized Cox proportional hazards regression-based software reliability modeling with metrics data," in *2013 IEEE 19th Pacific Rim International Symposium on Dependable Computing*, 2013: IEEE, pp. 328-337, doi: https://doi.org/10.1109/PRDC.2013.55.

[52] H. Okamura and T. Dohi, "A novel framework of software reliability evaluation with software reliability growth models and software metrics," in *2014 IEEE 15th International Symposium on High-Assurance Systems Engineering*, 2014: IEEE, pp. 97-104, doi: https://doi.org/10.1109/HASE.2014.22.

[53] H. Okamura and T. Dohi, "Towards comprehensive software reliability evaluation in open source software," in *2015 IEEE 26th International Symposium on Software Reliability Engineering (ISSRE)*, 2015: IEEE, pp. 121-129, doi: https://doi.org/10.1109/ISSRE.2015.7381806.

[54] M. Wiper, A. Palacios, and J. Marín, "Bayesian software reliability prediction using software metrics information," *Quality Technology & Quantitative Management*, vol. 9, no. 1, pp. 35-44, 2012, doi: https://doi.org/10.1080/16843703.2012.11673276.

[55] N. Torrado, M. P. Wiper, and R. E. Lillo, "Software reliability modeling with software metrics data via Gaussian processes," *IEEE Transactions on Software Engineering*, vol. 39, no. 8, pp. 1179-1186, 2012, doi: https://doi.org/10.1109/TSE.2012.87.

[56] V. Nagaraju, C. Jayasinghe, and L. Fiondella, "Optimal test activity allocation for covariate software reliability and security models," *Journal of Systems and Software*, vol. 168, p. 110643, 2020/10/01/ 2020, doi: https://doi.org/10.1016/j.jss.2020.110643.

[57] S. Yamada, J. Hishitani, and S. Osaki, "Software-reliability growth with a Weibull test-effort: a model and application," *IEEE Transactions on Reliability*, vol. 42, no. 1, pp. 100-106, 1993, doi: https://doi.org/10.1109/24.210278.

[58] P. Moranda, "Prediction of software reliability during debugging," in *Proc. 1975 Annu. Reliab. Maintainance Symp.*, 1975.

[59] W. Farr, "Software reliability modeling survey," in *Handbook of software reliability engineering*: McGraw-Hill, Inc., 1996, pp. 71–117.

[60] T. J. Archdeacon, *Correlation and regression analysis: a historian's guide*. Univ of Wisconsin Press, 1994.

[61] K. Sharma, R. Garg, C. K. Nagpal, and 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, 2010, doi: https://doi.org/10.1109/TR.2010.2048657.

[62] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716-723, 1974, doi: https://doi.org/10.1109/TAC.1974.1100705.

[63] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, pp. 461-464, 1978, doi: https://doi.org/10.1214/aos/1176344136.