

Comparison of Non-Homogeneous Poisson Process Software Reliability Models in Web Applications

Rabia Burcu Karaömer, *PROVEN Information Technologies, bkaraomer@proven.com.tr*

Oumout Chouseinoglou, *Hacettepe University Faculty of Engineering, Department of Industrial Engineering, Ankara, Turkey, Asst. Prof. Dr, uhus@hacettepe.edu.tr*

ABSTRACT *Software reliability is an important quality factor that effects project success. By modelling software reliability, it can be estimated when and with how much effort a project can be deployed. Consequently, this can contribute to the resource and schedule planning of a project. Therefore, software reliability models (SRM) are frequently used for measuring the maturity of a software. A number of studies exist in the literature that compare SRMs in terms of their modelling performance. However, there is a need of evaluating these SRMs by taking into account the software project domain. This study aims to compare the performance of SRMs in the context of Web applications. In accordance to this purpose, six different software reliability models, namely Goel-Okumoto, Musa Exponential, Inflected S-shaped, Delayed S-shaped, Yamada and Pham Nordmann Zhang Imperfect Fault Detection (PNZ), are evaluated by using the defect records of four Web application projects developed by a Turkish software organization. 100%, 70% and 50% of the recorded data is used as input to the maximum likelihood parameter estimation (MLPE) method and the results of these three cases are investigated and commented separately in the research. The goodness of fit and the predictive validity of the models to the project data are tested by calculating Mean Square Error (MSE), Mean Magnitude Relative Error (MMRE), Percentage Relative Error Deviation (PRED) and Average Balanced Predicted Relative Error (A.BPRE) measures. For each NHPP model 48 separate cases which are combinations of the three defect inflow data cases (100%, 70% and 50%), four projects and four measures, are investigated and ranked. It is shown that the NHPP models can be applied to Web applications and Delayed S-shaped model displays the best results among the alternatives. However, it is understood that the Goel-Okumoto and Yamada models give identical results and that these two models converge to each other with respect to the project defect data that has been used. Combined, these two models obtain the highest ranking scores and it is concluded that these two models perform better than the other models with respect to Web based software.*

Keywords :Web applications, software reliability models, failure rate, non-homogeneous Poisson process

Web Yazılım Projelerinde Homojen Olmayan Poisson Süreci Yazılım Güvenilirlik Modellerinin Karşılaştırılması

Öz Yazılım güvenilirliği, proje başarısını doğrudan etkileyen önemli kalite faktörlerinden biridir. Yazılım güvenilirliğinin modellenmesi ile bir projenin ne kadar zaman sonra ve ne kadar efor sonucunda kullanıcıya sunulabileceği tahmin edilebilir. Bu da proje kaynak ve takvim planlamasında yardımcı olabilmektedir. Bu amaçla yazılım

güvenilirlik modelleri yazılımların olgunluklarını ölçmede sıklıkla kullanılmaktadır. Literatürde yazılım güvenilirlik modellerinin karşılaştırılmasına yönelik birçok çalışma bulunmasına rağmen yazılım türünü dikkate alıp bu kapsamda yazılım güvenilirlik modellerinin karşılaştırılmasının yapılmasına ihtiyaç vardır. Bu çalışma, yazılım güvenilirliğini ölçmek için kullanılan modellerin Web yazılımlarındaki performanslarının karşılaştırılmasını hedeflemektedir. Bu amaçla Türkiye'de bir yazılım şirketinin dört ayrı Web yazılım projesinde tutulan hata kayıtları kullanılarak altı ayrı yazılım güvenilirlik modeli karşılaştırılmıştır. Karşılaştırmada kullanılan modeller; Üstel Homojen Olmayan Poisson Süreci Goel Okumoto, Musa Üstel, Büklümlü S Şekli Homojen Olmayan Poisson Süreci, Geciktirilmiş S Şekli Homojen Olmayan Poisson Süreci, Yamada ve Pham-Nordmann-Zhang Kesin Olmayan Hata Tespiti (PNZ) modelleridir. Yazılım güvenilirlik modellerinin uygulanması sırasında en çok olabilirlik tahmin yöntemi kullanılarak hata kayıt verilerine uygun model parametreleri, her bir model ve her bir proje için tahmin edilmiştir. En çok olabilirlik yöntemi ile model parametrelerinin tahmin edilmesi sırasında, elde bulunan hata kayıt verilerinin %100'ünün, %70'inin ve %50'sinin kullanılması durumları olmak üzere üç durum ayrı ayrı incelenmiş ve sonuçları değerlendirilmiştir. Tahmin edilen parametrelerle oluşturulan modellerin projelere uygunluğu hata kareler ortalaması, ortalama bağıl hata, yüzde bağıl hata sapması ve dengeli tahmini bağıl hata ölçümleri kullanılarak hesaplanmıştır. Her bir model için dört proje, üç hata durumu (%100, %70, %50) ve dört ölçüm sonucuna göre toplamda 48 farklı ölçüm alınmıştır. Bu 48 ölçüm içerisinde her bir ölçüm için en yüksek başarıya sahip model seçilmiş ve modeller buna göre sıralanmıştır. Çalışma sonucunda Homojen Olmayan Poisson Süreci modellerinin Web yazılımlarında kullanılabileceği gösterilmiş olup Geciktirilmiş S Şekli Homojen Olmayan Poisson Süreci yazılım güvenilirlik modeli 13 durum ile en çok durumda başarılı model olmuştur. Ancak Yamada ve Üstel Homojen Olmayan Poisson Süreci Goel Okumoto modellerinin birbirlerine benzer hata tahminleri yapıp birbirlerine yakınsadığı düşünülmektedir. Bu nedenle bu iki modelin toplamda 23 durum ile en çok durumda en iyi sonucu veren modeller olması ile kullanılan diğer yazılım güvenilirlik modellerine göre daha iyi modelleme yapacağı düşünülmektedir.

Anahtar Kelimeler: Web yazılımları, yazılım güvenilirlik modelleri, bozulma oranı, homojen olmayan poisson süreçleri

Introduction

Web applications, with the proliferation of the Internet, have become part of everyone's daily life, having an important impact on businesses, entertainment and education. Li, Das and Dowe (2014) define Web applications as systems that are typically composed of a back-end (a database) and front-ends (Web-pages) that the users interact with over a network by using a browser. Web applications, depending on whether they change based on the user inputs, interactions, sequences of interactions, etc. can be static or dynamic (Li, Das, & Dowe, 2014). Öztürk, Çavuşoğlu and Zengin (2015) differentiate Web applications from other software by regarding them as distributed systems that are multi-user heterogeneous systems, where different software work in the same architectural environments and embody software that respond according to input and server values. Moreover, Web applications have very high quality requirements and are considered to be highly interactive applications that employ several new languages, technologies and programming models (Qian & Miao, 2011). The

aforementioned features of Web applications and their pervasiveness necessitates them to be of high quality with low defectiveness. As any software, Web applications are also tested. Di Lucca and Fasolino (2006) state that the testing of Web applications is conducted in order to address major software engineering issues such as maintainability, testability, security, performance, correctness and reliability; whereas Qian and Miao (2011) point out that for ensuring the requested software quality of Web applications, testing will have a gradually increasing importance and relevance. However, several drawbacks exist regarding the testing of Web applications. First of all, the testing of Web applications is frequently neglected by developers. The main reasons for that are the market pressure, the very short time-to-market periods and the fact that the testing efforts, especially for Web applications, are considered to be too time-consuming and with insignificant returns (Hieatt & Mee, 2002; Di Lucca & Fasolino, 2006). Moreover, as stated by Torchiano, Ricca and Marchetto (2011), the number of studies to understand the specific nature of Web application defects is inadequate and Web application testing techniques and tools are more immature compared to the techniques and tools used for testing desktop or embedded applications. Similarly, Ferrara, De Meo, Fiumara and Baumgartner (2014) argue that the increase in the complexity of Web applications results to test approaches becoming more and more insufficient.

The concept of software testing is closely related to software reliability and one major approach of assessing the reliability of a Web application is using software reliability models (SRMs). The IEEE 1633: Recommended Practice of Software Reliability (IEEE Reliability Society, 2008) not only specifies the recommended procedures for software reliability assessment and prediction but also defines two major concepts that are used throughout this paper:

Software Reliability: is (a) the probability that software will not cause the failure of a system for a specified time under specified conditions, or (b) the ability of a program to perform a required function under stated conditions for a stated period of time.

SRM is a mathematical expression that specifies the general form of the software failure process as a function of factors such as fault introduction, fault removal, and the operational environment.

SRMs are used to estimate the total number of defects and the rate of software deterioration within a particular time period by using numeric data and are the result of applying reliability engineering theory to the software development domain (Rana, et al., 2014). The deterioration behavior of the software is predicted by using the known or assumed characteristics of the software in question (Lai & Garg, 2012). During the initial testing processes, some of the software defects are identified and resolved; as the tests progress over time the number of unidentified defects continuously decrease, in other words as the probability of finding a defect decreases as the reliability of the software increases (Yamada, 2014). As stated by Rana et al. (2014), the SRMs mainly try to answer two very practical question: “when the attained software quality is enough so that the testing can stop?” and “considering the remaining defects, is the software ready to be released?”. Therefore, we do believe that SRMs can provide a solution to the aforementioned testing problems that are experienced in Web applications

and in this research we focus on a specific category of SRMs, namely Non-Homogeneous Poisson Process (NHPP) models.

This study tries to answer the following three research questions (RQ):

- **RQ1:** Are the NHPP models applicable to Web applications?
- **RQ2:** How each one of the assessed NHPP models perform with respect to Web applications?
- **RQ3:** How good predictors of the future testing stages in a Web application development are the NHPP model parameters that are estimated by using the 70% and 50% of the defect data?

The rest of this paper is structured as follows: First we review the basic literature regarding SRMs and NHPP models, and provide the required background information. Section 3 presents the methodology used in our research. The results and the analysis of these results are given in Section 4. The last section concludes the paper, addresses the validity threats and gives the planned future work.

Related Work and Background

Reliability is an important factor with respect to the success of Web applications. As stated by Offutt (2002), the majority of Web applications perform within high competition environments, where users select the software to use based on the level this software meets their requirements and can easily switch to some other application if their requirements are not met. When assessing its reliability, the different characteristics of a software should be taken into consideration, one of them being the type of the software being assessed. The differences of Web applications with respect to other software are given in detail in several studies (Offutt, 2002; Mendes, 2014; Murugesan, Deshpande, Hansen, & Ginige, 2001). A list of testing techniques and approaches currently utilized for Web based applications are given in detail in (Li, Das, & Dowe, 2014), (Di Lucca & Fasolino, 2006) and (Fasolino, Amalfitano, & Tramontana, 2013). The work of Garousi, Mesbah, Betin-Can and Mirshokraie (2013) is a detailed systematic mapping study of Web application testing and based on that mapping, a systematic literature review of Web application testing is given by Doğan, Serdar and Garousi (2014). In these two systematic literature reviews the authors discuss the emerging trends in Web application testing concluding that Web testing is an active area of research with an increasing number of publications. Similarly, Qian and Miao (2011) state that with respect to Web based software testing many issues have not been sufficiently investigated yet, and many open questions, one of them being the reliability of Web applications, still need to be addressed. Moreover, Hieatt and Mee (2002) argue that adequate, efficient and cost effective testing approaches for testing Web applications are needed, increasing the significance of the research on Web software reliability.

As software reliability is considered to be a must-be quality characteristic, many SRMs have been proposed and applied to practical use (Yamada, 2014). Pham (2006) states that SRMs can be grouped to two: the *deterministic models* that are used to study the number of distinct

operators and operands, the number of errors and the number of machine instructions in the program, and the *probabilistic models* that represent the failure occurrences and the fault removals as probabilistic events. The probabilistic SRMs can be classified into different groups, namely error seeding, failure rate, curve fitting, reliability growth, Markov structure, time-series and NHPP. This study focuses only on the NHPP models.

NHPP models are based on estimating the mean value function of the cumulative number of defects that are observed up to a certain point in time and thus provide an analytical framework for describing the defect identification during testing (Yamada, 2014). The mean value function of NHPP models' cumulative defects identified up to point t is generally in the form of

$$m(t) = \int_0^t \lambda(s) ds$$

where $\lambda(t)$ is the defect density function. Similarly, the reliability is defined with the following function:

$$R(t) = e^{-m(t)}$$

The most common NHPP models that are used in this study are given in **Table 1** where $a(t)$ are the total defects at time t , identified and not, whereas $b(t)$ stands for the defect identification rate at time t (Pham, 2007). The NHPP models make these specific assumptions: the defects in a software occur randomly and are independent of each other, the cumulative sum of the defects up to time t follow the NHPP, at any point in the process the defect density rate in the software is relative to the yet unidentified defects in the software and when a defect is identified efforts are undertaken to fix it and these efforts are irrelevant of the defect location (Pham, 2006; Xie, Hong, & Wohlin, 2003).

Table 1 NHPP models used in this study

		Mean value function	Total defects	Defect identification
NHPP Model	Goel – Okumoto	$m(t) = a(1 - e^{-bt})$	$a(t) = a$	$b(t) = b$
	Delayed S-shaped	$m(t) = a(1 - (1 + bt)e^{-bt})$	$a(t) = a$	$b(t) = \frac{(b^2t)}{(1 + bt)}$
	Inflection S-shaped	$m(t) = \frac{a(1 - e^{-bt})}{(1 + \beta e^{-bt})}$	$a(t) = a$	$b(t) = \frac{b}{(1 + \beta e^{-bt})}$
	Musa Exponential	$m(t) = a(1 - e^{-ct/nT})$	$a(t) = a$	$b(t) = \frac{c}{nT}$
	Yamada	$m(t) = \frac{ab(e^{at} - e^{-bt})}{(\alpha + b)}$	$a(t) = ae^{\alpha t}$	$b(t) = b$
	Pham Nordmann Zhang (PNZ)	$m(t) = a - ae^{-bt}(1 + (b + d)t + bdt^2)$	$a(t) = a$	$b(t) = (b^{2t} / (1 + bt)) - (d / (1 + dt))$

NHPP models are approaches that their popularity are continuously increasing within the SRM domain (Lai & Garg, 2012; Pham, 2006). Pham (2003), while comparing the performances of NHPP models investigates also the effect of environmental factors and used budget model to the software reliability. Rana et al. (2014) evaluate eight SRMs (namely Musa-Okumoto, Goel-Okumoto, Inflection S-shaped, Delayed S-shaped, Rayleigh, Logistic, Gompertz and Linear) on 11 large software projects within the embedded software domain from three different companies. The authors mainly try to answer which SRMs are the best to assist decisions for optimal allocation of testing resources and which SRMs assess best the release readiness of a software system. They observe that Gompertz model is the best for software development processes that are either V-model based or follow lean and agile software development processes, and the Logistic model for waterfall development process. Whereas for assessing the release readiness of a software Logistic and Gompertz perform best from the perspective of asymptote precision and also when the shape of defect inflow is predicted. In a similar study by Ullah, Morisio and Vetro (2012) eight SRMs (namely Musa-Okumoto, Inflection S-shaped, Goel-Okumoto, Delayed S-shaped, Logistic, Gompertz, Yamada Exponential and Generalized Goel Model) are evaluated for fifty different defect record data sets coming from system test phases, field defects and Open Source Software projects. All models are tested for their performances and investigated for each situation. The authors argue that when all data sets are considered Musa-Okumoto fits best, but for all Open Source Software data sets all examined models do fit. When the prediction capability is considered for industrial data sets Musa-Okumoto, Inflection S-shaped and Goel-Okumoto models are the best predictors, whereas for Open Source Software Gompertz and Yamada are the best predictors. When fitting and prediction capability are combined in the industrial datasets Musa-Okumoto and Inflection S-shaped models are the best performers, and for the Open Source Software data sets the best performers are Gompertz and Inflection S-shaped models. Aydın and Tarhan (2014) investigate if defect density estimations and project life cycle are related, and for a project with an iterative software development life cycle the performance of Rayleigh and Linear Regression models are assessed. Comparison results of actual and predicted defect density values show that at module level the Rayleigh Model and at project level the Linear Regression Model produce more reliable results.

Methodology

Following Robson's (2002) classification this research is a case study that aims to evaluate the applicability and performance results of a specific group of SRMs, namely NHPP models, in the context of Web application development. Based on the guidelines and taxonomy of Runeson and Höst (2009), this study is an interpretive case study that uses a fixed design principle, organized as an embedded case study where each of the four projects analyzed constitute the units of analysis. The overview of the case study design is given in **Figure 1**.

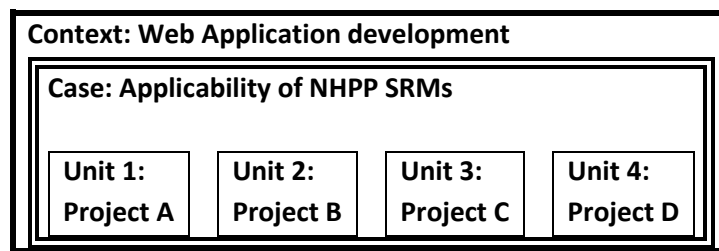


Figure 1 Overview of Case Study Design

The case study was conducted at ALTAIR Software and Defence Technologies Inc., a software development organization with a ISO 9001:2008 Quality Certificate, employing approximately 30 employees that work in different project groups with varying responsibilities as developers, testers, project managers and quality assurance specialists. The details of the four Web application projects used in the case study are given in **Table 2**. The defect data were collected from the JIRA Issue & Project Tracking Software used by the Organization and semi-structured interviews were conducted with the responsible project team members to complement the defect data. The cumulative normalized defect inflow profiles for the four projects analyzed are presented in **Figure 2**.

SRMs can be fitted to the observed defect data using statistical techniques such as maximum likelihood parameter estimation (MLPE) or curve fitting techniques like non-linear least square. As Zhao and Xie (1996) state MLPE is the most common approach used while estimating the parameters of NHPP models, therefore in this research the MLPE is used and the function sets were solved with the use of MATLAB and MATLAB Optimization Toolbox.

Table 2 Overview of Projects Within this Embedded Case Study

	Project A	Project B	Project C	Project D
Development Lifecycle	Spiral	Spiral	Waterfall	Waterfall
Line of Codes ¹	40000	10300	12900	13343
Status	Ongoing	Ongoing	Completed	Completed
Vendee	R&D	R&D	Customer X	Customer Y
Development Environment	Windows Eclipse	Windows Eclipse	Windows Eclipse	Windows Eclipse
Programming Language	Java	Java	Java	Java
Deployment Platform	Linux	Linux	Linux	Linux
Implemented Test Types	Unit tests, System tests	Unit tests, System tests	Unit tests, System tests	Unit tests, System tests

To assess which NHPP SRMs are performing best and when it is more appropriate for them to be applied during a project timeline, the approach proposed by Rana et al. (2014) was employed in this study. The NHPP SRMs in question were evaluated by using full and partial data sets. For each of the four projects the defect data was divided into four sets containing all, 70% and 50% defect data points respectively (observed region), corresponding to the same

¹ The Line of Code information for the ongoing projects was obtained at 8.11.2015.

level of project completion with respect to project timeline. Then the data points in the observed region for each set were used for model fitting, while the defect data points outside of this region (predicted region, 0%, 30% and 50% respectively) were used to assess the predictive power of the SRMs.

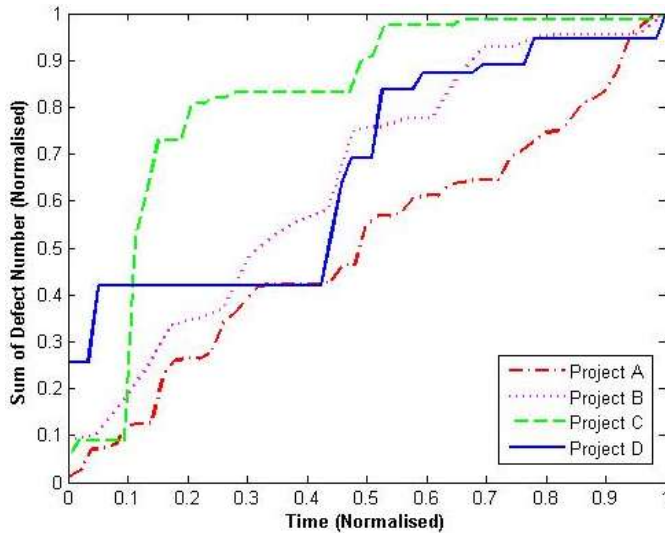


Figure 2 Normalized Cumulative Defect Inflow Profiles for the Analyzed Four Projects

Following the identification of the data sets, the SRMs were evaluated with respect to goodness-of-fit and predictive validity criteria (Ahmad, Khan, & Rafi, 2011). The methods of Mean Square Error (MSE), Mean Magnitude Relative Error (MMRE), Percentage Relative Error Deviation (PRED) and Average Balanced Predicted Relative Error (A.BPRE) were used. In MSE, the error is the difference between actual and predicted defect values and the lower the MSE, the better is the model's goodness-of-fit. Magnitude Relative Error (MRE) computes the absolute percentage of error between actual defects (ea) and estimated defects (ee), for each project:

$$MRE_i = \frac{|ea_i - ee_i|}{ea_i}$$

MMRE calculates the average of MRE over all investigated items (n):

$$MMRE = \frac{1}{n} \sum_{i=1}^n MRE_i$$

$PRED(q)$ is used to count the percentage of defect estimates that fall within less than or equal to q of the actual values.

$$PRED(q) = \frac{\lambda}{N}$$

where λ is the number of projects where $MRE_i \leq q$ and N is the number of all estimates. As stated by Chouseinoglou and Aydın (2013), an estimation model with lower MMRE and

higher $PRED(q)$ can be interpreted is more accurate than other models and a model whose $MMRE \leq 0.25$ and $PRED(0.25) \geq 0.75$ is considered to be a good one. Finally, the A.BPRE, introduced by Rana et al. (2014), is based on Predicted Relative Error (PRE):

$$PRE_{(i)} = \frac{(ea_i - ee_i)}{ee_i}$$

To obtain more balanced results with respect to positive or negative deviations, the Balanced Predicted Relative Error (BPRE) is used:

$$BPRE_{(i)} = \frac{(ee_i - ea_i)}{ee_i + 2\eta(ea_i - ee_i)}, \quad \eta = \begin{cases} 0, & ee_i > ea_i \\ 1, & ee_i < ea_i \end{cases}$$

The BPRE obtains values in the range of -1 to 1. A value closer to 0 denotes that the model is producing better estimations. The function of the A.BPRE is as follows:

$$A.BPRE = \frac{\sum_{i=1}^n BPRE_{(i)}}{n}$$

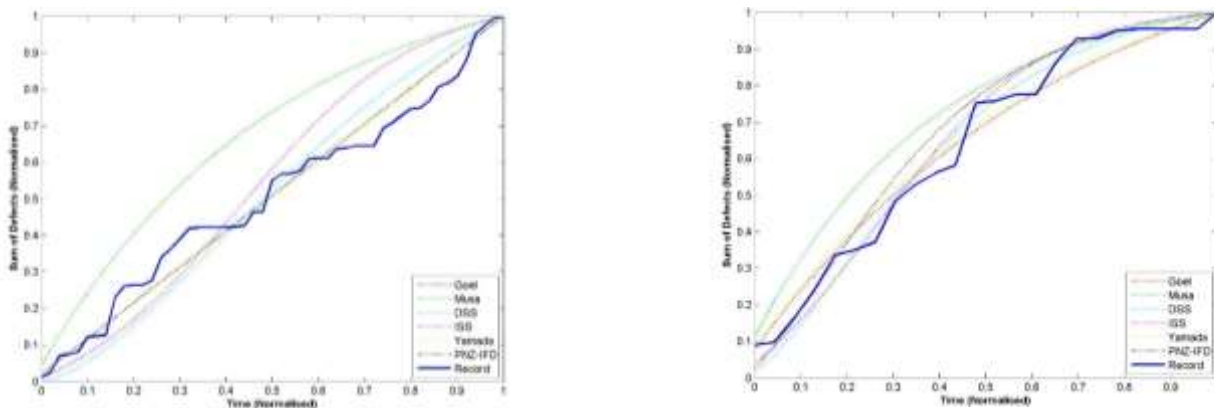
Results and Analysis

Since each examined project has different defect inflow characteristics, each project was analyzed separately. Initially the three datasets were created with respect to the amount of defect data to be used (100%, 70%, 50%) and then the parameters for each SRM NHPP model and for each dataset were calculated with the use of MLPE. The calculated NHPP model parameters are given in **Table 3**. When the parameters are analyzed it is observed that the models estimate that the actual total defects for Project A are much more than what it is observed and that only 80%-90% of the total defects for Project B have been already identified. This is in accordance with the fact that these projects (A and B) are not completed yet and are still ongoing projects. On the contrary, all investigated models denote that almost all defects have been identified for Project C and that it is ready to be released to customer. Project D is a completed project, but the recorded defect data shows that the project is still producing errors and the model parameters can be interpreted as that Project D has reached a specific maturity but still a significant number of defects exist in the software. Based on the parameters it is estimated that something as 10%-15% of all defects are yet to be discovered in the software. It can be argued that if Project D is released to the customer in its current state, the probability that it will crash is higher than Project C. The graphs of each investigated NHPP model drawn based on the parameters calculated for 100%, 70% and 50% of data are given in **Figure 3**, **Figure 4** and **Figure 5**. Each of the NHPP models are evaluated with respect to the four performance methods, namely MSE, MMRE, PRED and A.BPRE. The values of the four performance methods for each project (A, B, C and D) and each input data (%100, %70 and %50) are given in **Table 4**. For MSE, MMRE and A.BPRE the closer to 0 values denote a better result, whereas for PRED the values closer to 1 are interpreted as a better result. In **Table 4**, each superscript value next to the calculated value shows the ranking of that performance measure within the column. For example, the MSE value for Project A / 100% is given as **314.2668²**, here 314.2668

is the actual calculated value whereas the superscript (2) denotes that this MSE value is the second best (ranks second) among all MSE values for Project A / 100%. Several models have generated similar PRED values, in this case the PRED values were ranked based on their respective MMRE values, as PRED is calculated by using MMRE. The calculated MSE values for 70%-30% (observed and predicted) and 50%-50% (observed and predicted) of each project are given in **Figure 6**. The models are sorted for each project from the highest total MSE to the lowest one. The absolute A.BPRE values of each NHPP Model for all four projects and three data sets (100%, 70% and 50%) are given in **Figure 7**. The NHPP models are sorted in increasing order of total A.BPRE. It is observed in the graph that Yamada is the model that displays the best results with respect to the A.BPRE values.

In RQ1 we tried to examine whether the NHPP models are applicable to Web applications. When the defect inflow data of the Web applications that were used in this study is examined it is observed that defect creation is a random process and that they do not follow a specific period. With respect to these characteristics it can be argued that the defect creation of the examined Web applications satisfy the assumptions of the NHPP models. The results of the examined NHPP models were communicated with project team members and they have ratified the results. Based on these findings, we conclude that SRM NHPP models are applicable to Web applications.

In order to answer RQ2, that is how each one of the assessed NHPP models perform with respect to Web applications, the performance measurements obtained from each one of the assessed NHPP models are examined. Our initial observation is that a single model does not outperform all others significantly. Thus, the results are summarized and the best and second best performing models for each case (100%, 70% and 50% in all four projects) are given in **Table 5**. The values in Table 5 are read as: $A_{50\%}$ standing for the measurement for Project A conducted with the 50% of the given defect data. Therefore each of the NHPP models are evaluated based on data collected from four projects (A, B, C and D), with three datasets (%100, %70 and %50) and by using four different measurement approaches (MSE, MMRE, PRED and A.BPRE), thus resulting to a total of 48 measurements. As it can be seen the Delayed S-shaped model performs best in 13 cases, the Yamada model in 12 and the Goel-Okumoto model in 11 cases.



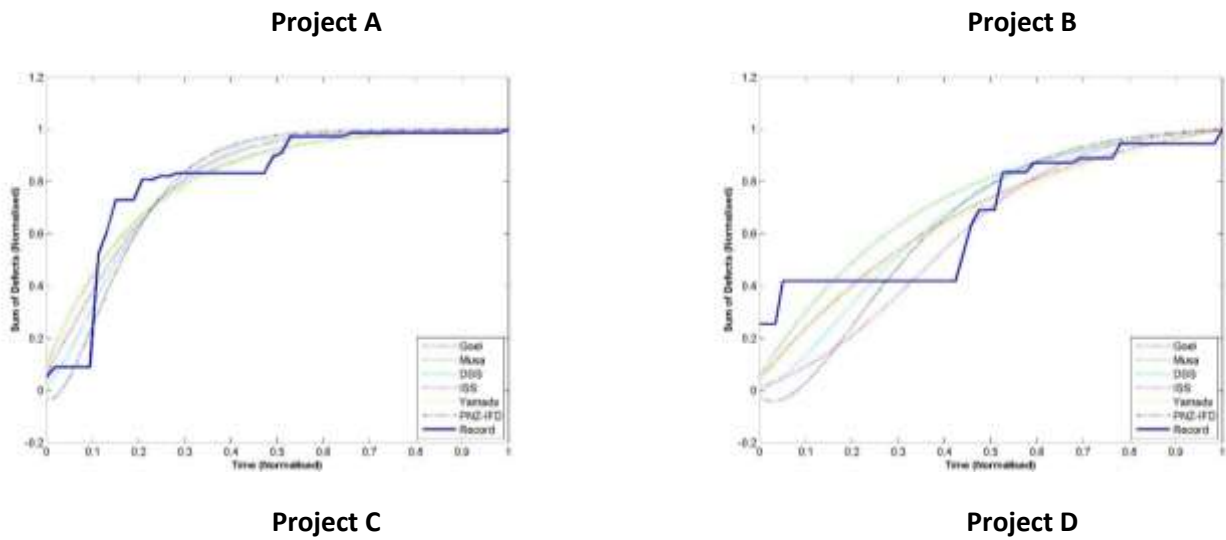


Figure 3 The graphs of NHPP models drawn based on the parameters calculated for 100% of defect data

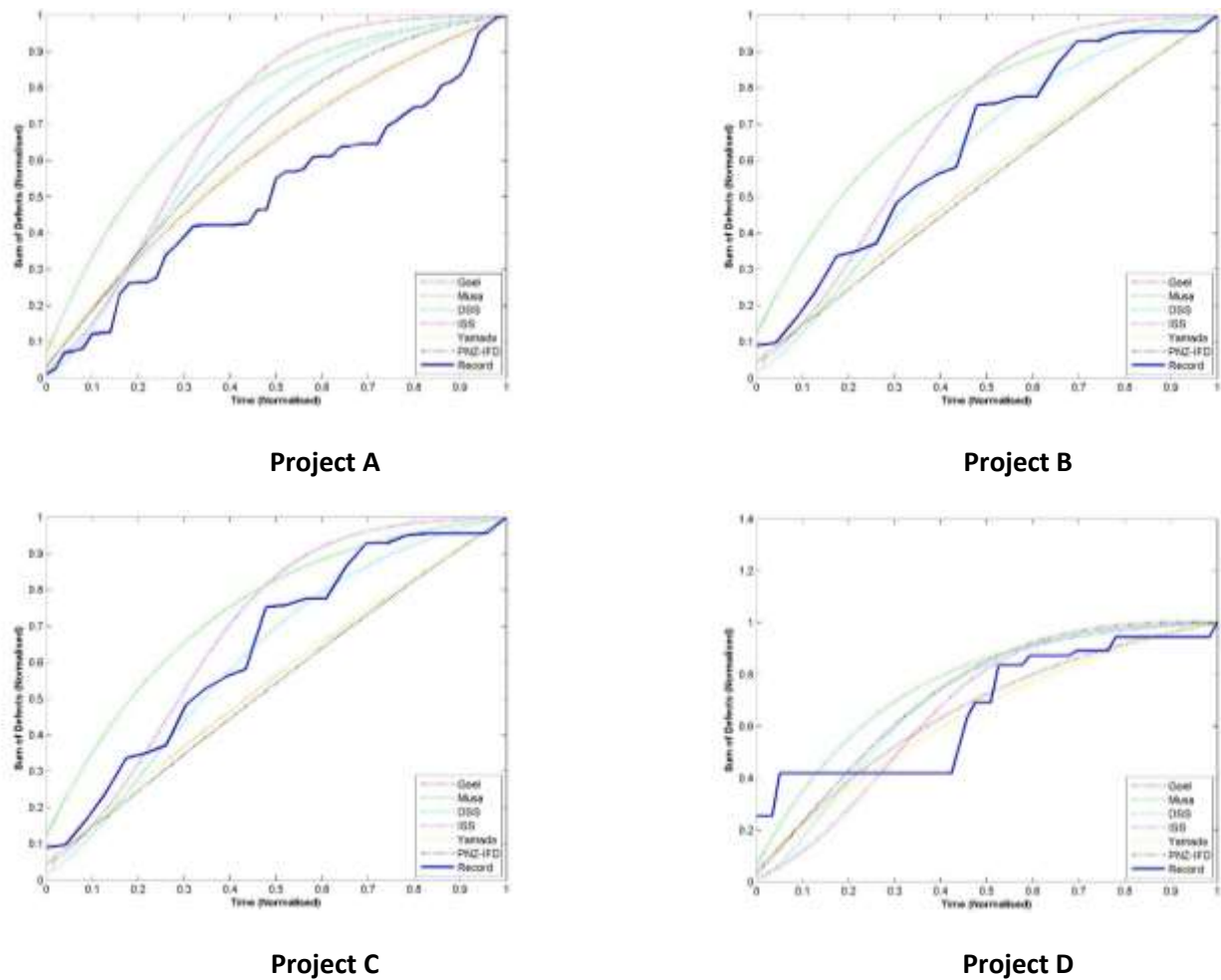


Figure 4 The graphs of NHPP models drawn based on the parameters calculated for 70% of defect data

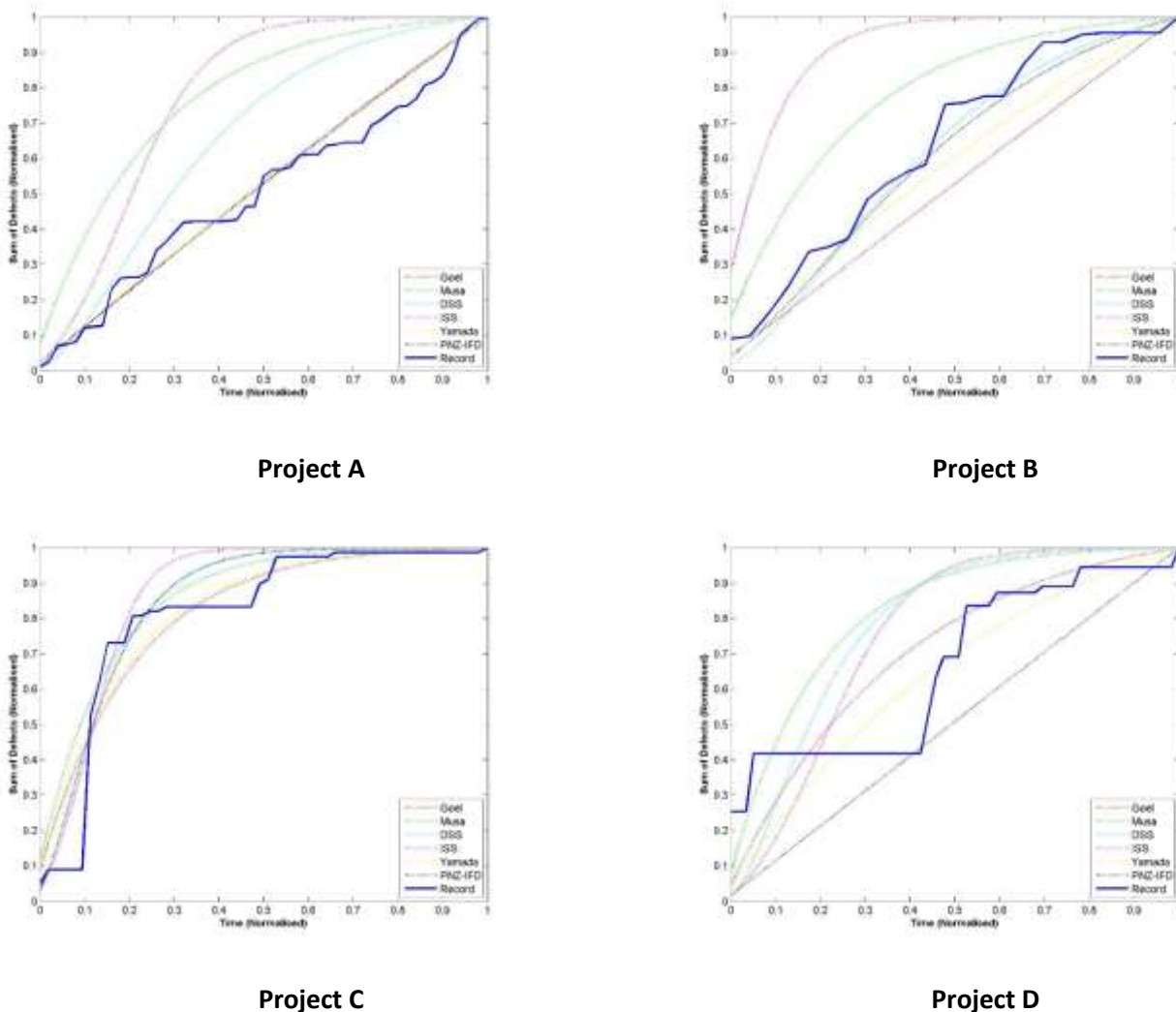


Figure 5 The graphs of NHPP models drawn based on the parameters calculated for 50% of defect data

Moreover, it is observed that Yamada and Goel-Okumoto models display very close results with respect to the four measurement types. These two models differentiate from each other with respect to their total defect ($a(t)$) functions: in the Goel-Okumoto model the total defect function is $a(t) = a$ whereas for the Yamada model this is given as $a(t) = ae^{\alpha t}$. The α parameters that were computed in this research and are given in **Table 3** are very small (near-zero) values, thus converging the two total defect functions. The small α values are interpreted as the introduction of very few new defects over time to the developed software. Because of these α values the two models converge to each other. Based on these findings we conclude that Yamada and Goel-Okumoto models give the best results in Web projects. On the other the worst performing model almost in all cases is the Pham Nordmann Zhang.

When the performance measurements for the models that were constructed by using the 70% and 50% of defect datasets are examined it is seen that the models perform good for projects C and D (the completed projects), but that the performance significantly drops for projects A and B (the ongoing projects). The outputs were discussed with project team members in order to understand RQ3, that is how good predictors of the future testing stages in a Web

application development are the NHPP model parameters that are estimated by using the 70% and 50% of the defect data. As stated by Rana et al. (2014), BPRE and A.BPRE can be used to assess the release readiness of a software. When **Figure 7** and **Table 5** are examined it is seen that Yamada and Delayed S-Shaped models perform best with respect to their A.BPRE values and therefore we can argue that Yamada and Delayed S-Shaped models are the best models for the assessment of the release readiness of a Web application.

Table 3 Calculated NHPP Model Parameters for Each Project

		Model Parameters	Project A			Project B			Project C			Project D		
			100%	70%	50%	100%	70%	50%	100%	70%	50%	100%	70%	50%
NHPP Model	Goel – Okumoto	<i>a</i>	1.6265e+09	439.8380	2958.697	287.9912	970.9838	4863.1588	78.6601	80.3144	76.595	63.4720	67.4027	51.882
		<i>b</i>	4.7497e-09	0.0239	0.00293	0.0620	0.0141	0.00293	0.0885	0.0839	0.0908	0.0336	0.0309	0.0439
	Delayed S-shaped	<i>a</i>	456.3914	281.34574	250.66	240.3114	236.3742	198.73	78.5387	78.9622	72.90	58.1371	53.225	41.233
		<i>c</i>	784.0297	592.0804	435.65	586.6168	431.6572	313.59	388.6127	284.5075	225.63	160.5720	124.1377	96.73
	Inflection S-shaped	<i>a</i>	417.3858	262.3664	228.911	229.9413	216.4770	179.195	78.4079	78.2116	70.122	56.0866	51.5912	41.2073
		<i>b</i>	0.1025	0.1615	0.2184	0.2446	0.3227	0.4571	0.1417	0.1724	0.3377	0.1054	0.1272	0.175
		<i>β</i>	9.9600	8.19	9.66	5.7300	9.9400	0.46	1.3600	2.24	9.99	9.9400	9.5800	9.9
	Musa Exponential	<i>a</i>	531.4072	283.7191	288.04	235.2945	259.0608	271.5965	78.0416	77.4571	71.057	56.5782	52.4353	40.622
		<i>b</i>	0.0519	0.1064	0.10427	0.1954	0.1758	0.17441	0.1839	0.1906	0.2288	0.0907	0.1051	0.1477
	Yamada	<i>a</i>	766.2404	418.559	2194	276.4678	981.4864	571.6172	78.21	79.967	75.5562	58.7899	74.2837	67.4602
		<i>b</i>	0.01	0.0259	0.004	0.0682	0.0139	0.029	0.0893	0.0864	0.0965	0.035	0.0256	0.0276
		<i>α</i>	0.0088	0.000026	0.00009	0.0001	0.00002	0.000001	0.000124	0.000014	0.000015	0.0016	0.000022	0.00002
	Pham Nordmann Zhang (PNZ)	<i>a</i>	568.5847	172.6774	277.0184	230.3869	131.9857	296.1905	78.0073	62.8146	71.1162	55.8397	41.2670	25.3371
		<i>b</i>	0.001	0.0372	0.01	0.2062	0.0196	0.11	0.2490	0.0734	0.2129	0.1239	0.0497	0.001
		<i>d</i>	0.0136	-0.0491	-0.0307	-0.01	-0.0938	-0.0287	0.0646	-0.0519	0.01	0.0336	-0.0355	-0.05

Table 4 Performance Values for Each NHPP Model

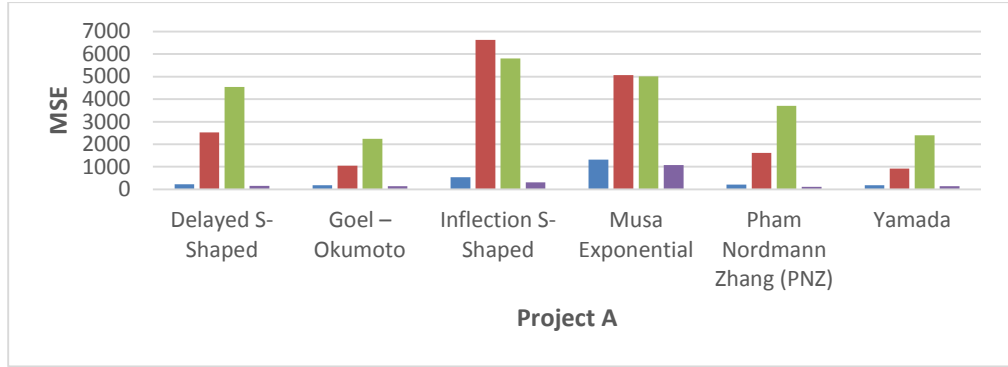
		Performance Methods	Project A			Project B			Project C			Project D		
			100%	70%	50%	100%	70%	50%	100%	70%	50%	100%	70%	50%
NHPP Model	Goel – Okumoto	MSE	314.2668 ²	2236.8 ¹	1045.504 ²	113.1534 ³	1527.8 ⁵	4056.722 ⁶	48.2321 ⁴	2.7 ⁵	6.5195 ¹	39.4855 ²	5.9 ⁵	28.8866 ²
		MMRE	0.1152 ¹	0.1085 ¹	0.1007 ³	0.0984 ²	0.1638 ⁵	0.2677 ⁶	0.2673 ³	0.0195 ⁵	0.0301 ¹	0.1877 ²	0.0396 ⁵	0.1053 ²
		PRED	0.9608 ¹	1 ¹	1 ²	0.9167 ²	1 ⁵	0.583 ⁶	0.8889 ³	1 ⁵	1 ¹	0.6833 ²	1 ⁵	1 ²
		A.BPRE	0.0081 ³	-0.0929 ¹	0.0895 ³	0.0305 ²	0.1367 ⁵	0.2024 ⁶	0.0460 ⁴	0.0191 ⁶	-0.0291 ¹	-0.0040 ¹	0.0366 ⁵	-0.0940 ²
	Delayed S-shaped	MSE	4753 ⁶	5011.4 ⁵	5069.251 ⁵	575.9913 ⁶	27.8 ²	360.863 ¹	47.7816 ³	1.07 ²	20.61 ³	47.7816 ³	1.6 ²	95.043 ⁴
		MMRE	0.5444 ⁶	0.1691 ⁵	0.1638 ⁵	0.2422 ⁶	0.0208 ²	0.0817 ¹	0.275 ⁴	0.0124 ²	0.0582 ³	0.2246 ³	0.0199 ²	0.1878 ⁴
		PRED	0.2157 ⁶	0.7333 ⁵	0.76 ⁵	0.5833 ⁶	1 ²	1 ¹	0.8889 ⁴	1 ²	1 ³	0.6667 ⁴	1 ²	0.967 ⁴
		A.BPRE	0.2972 ⁶	-0.1380 ⁵	-0.1322 ⁵	0.1573 ⁶	0.0201 ²	-0.0621 ¹	0.0538 ⁶	0.0121 ²	-0.0549 ³	0.0678 ⁵	-0.0141 ²	-0.1565 ⁵
	Inflection S-shaped	MSE	1726.6 ⁵	5809 ⁶	6639.237 ⁶	127.4596 ⁴	10.2 ¹	797.818 ³	45.0623 ²	1.1 ³	42.883 ⁶	67.1216 ⁵	1.9 ³	85.576 ³
		MMRE	0.1964 ⁴	0.1823 ⁶	0.1926 ⁶	0.0919 ¹	0.0101 ¹	0.1163 ³	0.2148 ²	0.0135 ³	0.0838 ⁶	0.2339 ⁴	0.0226 ³	0.1773 ³
		PRED	0.7255 ⁴	0.7333 ⁶	0.68 ⁶	0.9583 ¹	1 ¹	1 ³	0.8889 ¹	1 ³	1 ⁶	0.6833 ³	1 ³	0.967 ³
		A.BPRE	0.0022 ²	-0.1466 ⁶	-0.1501 ⁶	0.0019 ¹	-0.0006 ¹	-0.086 ²	0.0494 ⁵	0.0132 ³	-0.0772 ⁶	0.0916 ⁶	-0.0153 ³	-0.1457 ⁴
	Musa Exponential	MSE	936.5059 ⁴	4543 ⁴	2528.719 ⁴	83.4796 ¹	236.0 ³	494.7153 ²	42.2765 ¹	1 ¹	33.226 ⁵	64.593 ⁴	1.5 ¹	98.245 ⁵
		MMRE	0.2191 ⁵	0.1594 ⁴	0.095 ²	0.1067 ⁴	0.0665 ³	0.1037 ²	0.1713 ¹	0.0042 ¹	0.0739 ⁵	0.2398 ⁵	0.0194 ¹	0.1904 ⁵
		PRED	0.6667 ⁵	0.7333 ⁴	0.84 ⁴	0.8750 ⁴	1 ³	1 ²	0.8519 ⁵	1 ¹	1 ⁵	0.65 ⁵	1 ¹	0.967 ⁵
		A.BPRE	-0.0751 ⁵	-0.1308 ⁴	-0.0747 ¹	-0.0344 ⁴	0.0618 ³	0.0930 ³	0.0131 ¹	0.0032 ¹	-0.0686 ⁵	-0.0105 ²	-0.0130 ¹	-0.1582 ⁶
	Yamada	MSE	311.9062 ¹	2403.5 ²	920.3878 ¹	131.4286 ⁵	1502 ⁴	1557.2 ⁵	49.2921 ⁵	2.2616 ⁴	9.0438 ²	38.7966 ¹	10.674 ⁶	8.7466 ¹
		MMRE	0.1145 ³	0.1129 ²	0.0939 ¹	0.1110 ⁵	0.1620 ⁴	0.1694 ⁴	0.2680 ⁵	0.018 ⁴	0.0372 ²	0.185 ¹	0.0535 ⁶	0.0494 ¹
		PRED	0.9412 ³	1 ²	1 ¹	0.8750 ⁵	0.8571 ⁴	0.75 ⁵	0.8889 ³	1 ⁴	1 ²	0.7 ¹	1 ⁶	1 ¹
		A.BPRE	-0.0005 ¹	-0.0958 ²	0.0837 ²	0.0502 ⁵	0.1355 ⁴	0.1408 ⁴	0.0455 ³	0.0176 ⁴	-0.0357 ²	-0.0188 ³	0.0498 ⁶	-0.0348 ¹
Pham Nordmann Zhang (PNZ)	MSE	325.314 ³	3701.8 ³	1618.9 ³	91.0526 ²	1780.7 ⁶	1423.8 ⁴	53.3199 ⁶	3.8920 ⁶	32.6652 ⁴	97.4237 ⁶	2.5197 ⁴	133.9964 ⁶	
	MMRE	0.1154 ²	0.1415 ³	0.13 ⁴	0.1042 ³	0.1767 ⁶	0.1779 ⁵	0.1831 ⁶	0.0197 ⁶	0.0733 ⁴	0.2884 ⁵	0.0263 ⁴	0.1808 ⁶	
	PRED	0.9608 ²	0.8 ³	1 ³	0.9167 ³	0.7143 ⁶	0.9167 ⁴	0.8333 ⁶	1 ⁶	1 ⁴	0.65 ⁶	1 ⁴	0.7 ⁶	
	A.BPRE	0.0093 ⁴	-0.1174 ³	0.1129 ⁴	0.0316 ³	0.1458 ⁶	0.1496 ⁵	-0.0214 ²	-0.0190 ⁵	-0.0681 ⁴	-0.0379 ⁴	-0.0195 ⁴	0.1193 ³	

Received : 15.03.2016

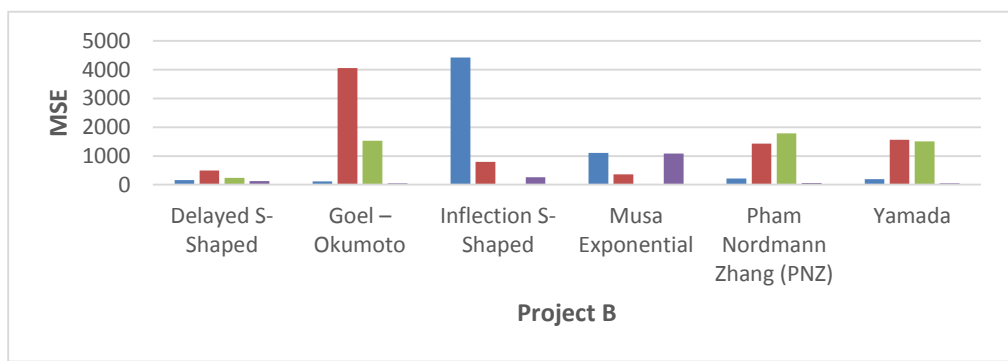
Editorial Process Begin: 10.04.2016

Published: 02.06.2016

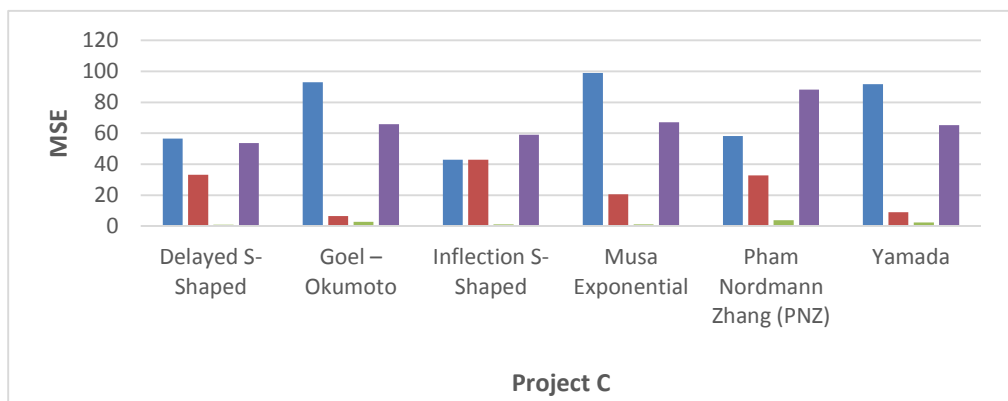
MSE of %70
(observed)



MSE of %30
(predicted)



MSE of %50
(observed)



MSE of %50
(predicted)

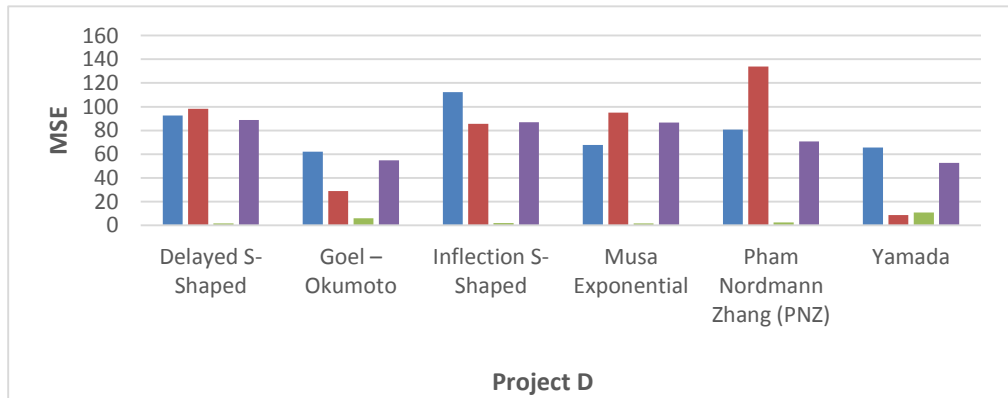


Figure 6 MSE Values for Studied NHPP Models over Full and Partial Data

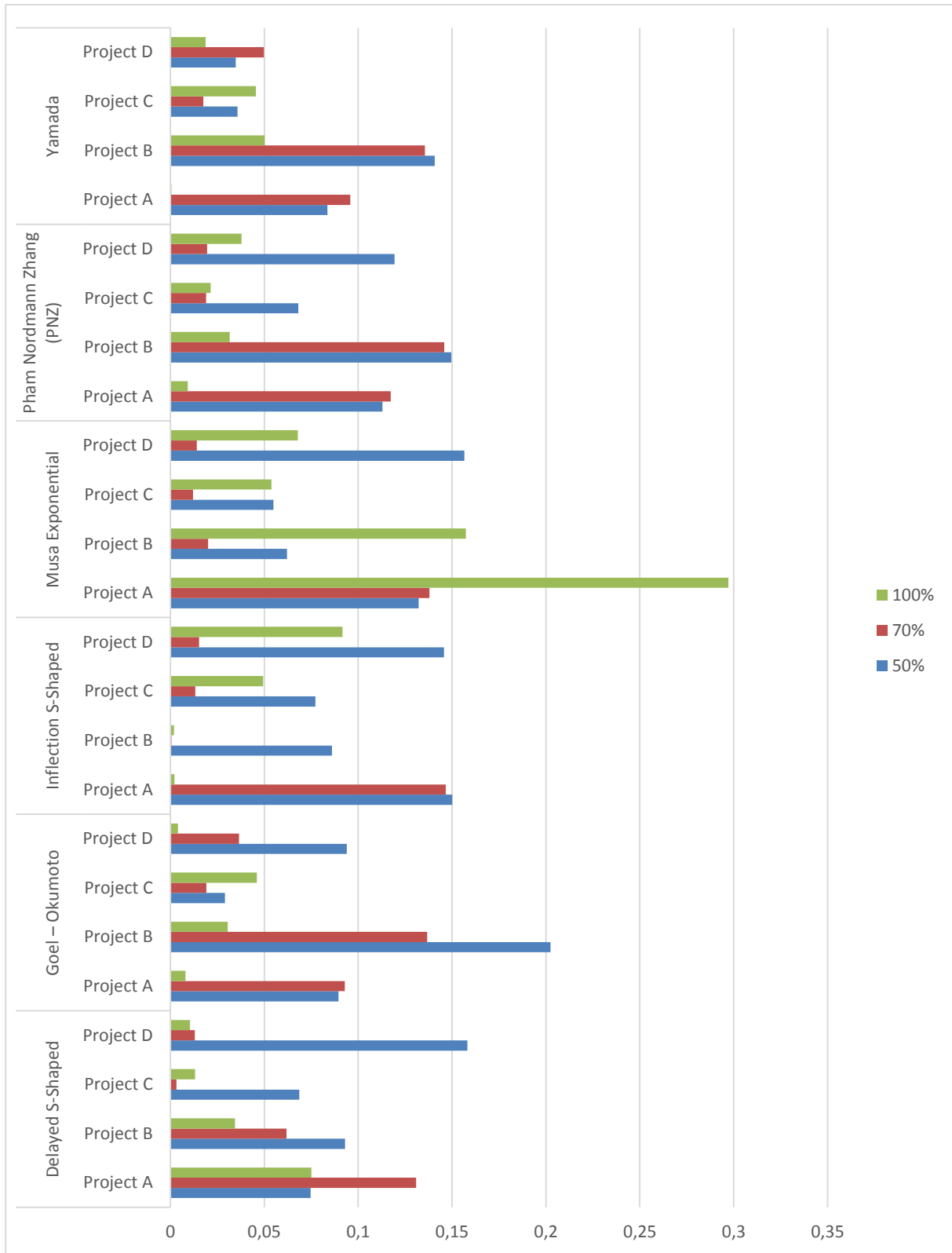


Figure 7 Absolute A.BPRE Values of Each NHPP Model

Table 5 Summary of the NHPP Model Performance Results (Best and Second Best Cases)

			MSE	MRE	PRED	A.BPRE	Total	Grand Total
NHPP Model	Delayed S-shaped	Best	B _{100%} , C _{100%} , C _{70%} , D _{70%}	C _{100%} , C _{70%} , D _{70%}	C _{70%} , D _{70%}	C _{100%} , C _{70%} , D _{70%} , A _{50%}	13	18
		Second Best	B _{50%}	B _{50%} , A _{50%}	B _{50%}	D _{100%}	5	
	Yamada	Best	A _{100%} , A _{50%} , D _{100%} , D _{50%}	A _{50%} , D _{100%} , D _{50%}	A _{50%} , D _{100%} , D _{50%}	A _{100%} , D _{50%}	12	21
		Second Best	A _{70%} , C _{50%}	A _{70%} , C _{50%}	A _{70%} , C _{50%}	A _{70%} , C _{50%} , A _{50%}	9	
	Goel – Okumoto	Best	A _{70%} , C _{50%}	A _{100%} , A _{70%} , C _{50%}	A _{100%} , A _{70%} , C _{50%}	D _{100%} , A _{70%} , C _{50%}	11	24
		Second Best	A _{100%} , D _{100%} , A _{50%} , D _{50%}	B _{100%} , D _{100%} , D _{50%}	B _{100%} , D _{100%} , D _{50%} , A _{50%}	B _{100%} , D _{50%}	13	
	Inflection S-shaped	Best	B _{70%}	B _{100%} , B _{70%}	B _{100%} , B _{70%} , C _{100%}	B _{100%} , B _{70%}	8	13
		Second Best	C _{100%}	C _{100%}	C _{100%}	A _{100%} , B _{50%}	5	
	Musa Exponential	Best	B _{50%}	B _{50%}	B _{50%}	B _{50%}	4	16
		Second Best	B _{70%} , C _{70%} , D _{70%}	B _{70%} , C _{70%} , D _{70%}	B _{70%} , C _{70%} , D _{70%}	B _{70%} , C _{70%} , D _{70%}	12	
	Pham Nordmann Zhang (PNZ)	Best	-	-	-	-	0	4
		Second Best	B _{100%}	A _{100%}	A _{100%}	C _{100%}	4	

Conclusion

In this paper we have evaluated six known NHPP SRMs on four software projects within the Web software context, developed in a Turkish software company. By using this unique setting and focusing on this scope three main research questions were set: In RQ1 we tried to examine whether the NHPP SRMs are applicable to Web applications and we discovered that NHPP models can be used with Web software defect data as the project and defect inflow characteristics are in accordance with the NHPP model characteristics. The results of our analysis and the feedback collected by project members ratify our assumption. In RQ2, the performance of each one of the assessed NHPP models with respect to Web applications was analyzed. A total of 48 measurements were compared and ranked and it is observed that Yamada and Goel-Okumoto models give the best results in the examined Web projects.

Finally, for RQ3 we tried to assess the release readiness of a Web application by trying to understand how good predictors of the future testing stages are the NHPP model parameters that are estimated by using the 70% and 50% of the defect data. Overall, Yamada and Delayed S-Shaped models performed better with respect to their A.BPRE values and therefore we argue that these models are better for the assessment of the release readiness of a Web application.

It is important to address the validity threats of this study and we do that by using the approach presented by Runeson and Höst (2009). The reliability of the study is concerned with to what extent the data and the analysis are dependent on the specific researchers. The defect data used was collected directly from the defect databases of the same organization. With the use of semi-structured interviews, it was assessed that all members of the software teams have the same understanding regarding the term “defect”, which was done in order to address the issues of construct validity. All formulas, computations and intermediate steps are given clearly, thus if the study is repeated the same results can be obtained. On the other hand, internal validity is related to causal relations. This study tries to assess how NHPP models perform with respect to Web applications, however factors like development lifecycle, development environment, used programming language and deployment platform may affect the results. In order to minimize these effects, we tried to select projects with different characteristics. Another threat to internal validity arises from using MLPE in order to estimate the SRM parameters. However, as stated by Zhao and Xie (1996) MLPE is the most common approach used while estimating the parameters of NHPP models. In order to minimize the internal validity threat regarding the performance assessment, four different performance criteria, namely MSE, MMRE, PRED and A.BPRE were used and the results from all four of them were used in the evaluation process. As the data used in this study comes from projects that belong to a single software development company, this is also a threat to the validity of the research. Therefore, in order to assess the generalizability of the obtained results with respect to external validity, it is important to conduct further similar case studies in other software organizations that develop Web software, which is also within the scope of planned future studies.

In this study the applicability of the NHPP SRMs in Web projects was analyzed with a practical focus. Further studies in similar direction, that may address different aspects of these models in the specific domain of Web applications, would allow further and wider adoption of reliability approaches, as there is a significant need for more efficient reliability solutions in Web software projects. Combining NHPP SRMs with time-series analysis and modelling separate life-cycle stages of a project with different SRMs are two of the initially planned future studies. We do believe that this research constitutes a necessary and important initial step towards the wider and more detailed use of SRMs, and specifically NHPP models, in the Web application development processes.

References

- Ahmad, N., Khan, M., & Rafi, L. (2011). Analysis of an Inflection S-shaped Software Reliability Model Considering Log-logistic Testing-Effort and Imperfect Debugging. *International Journal of Computer Science and Network Security*, 11(1), 161-171.
- Aydın, A., & Tarhan, A. (2014). Investigating defect prediction models for iterative software development when phase data is not recorded lessons learned. *2014 International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)* (pp. 1-11). IEEE.
- Chouseinoglou, O., & Aydın, Ö. (2013). A Fuzzy Model of Software Project Effort Estimation. *Turkish Journal of Fuzzy Systems*, 4(2), 68-76.
- Di Lucca, G., & Fasolino, A. (2006). Testing Web-based Applications: The State of Art and Future Trends. *Information and Software Technology*, 48(12), 1172-1186.
- Doğan, S., Betin-Can, A., & Garousi, V. (2014). Web application testing: A systematic literature review. *Journal of Systems and Software*, 174-201.
- Fasolino, A., Amalfitano, D., & Tramontana, P. (2013). Web Application Testing in Fifteen Years of WSE. *15th IEEE International Symposium on Web Systems Evolution (WSE)* (pp. 35-38). IEEE.
- Ferrara, E., De Meo, P., Fiumara, G., & Baumgartner, R. (2014). Web data extraction, applications and techniques: A survey. *Knowledge-Based Systems*, 70, 301-323.
- Garousi, V., Mesbah, A., Betin-Can, A., & Mirshokraie, S. (2013). A systematic mapping study of web application testing. *Information and Software Technology*, 55(8), 1374-1396.
- Hieatt, E., & Mee, R. (2002). Going faster: Testing the web application. *Software, IEEE*, 19(2), 60-65.
- IEEE Reliability Society. (2008). *IEEE Recommended Practice on Software Reliability*.
- Lai, R., & Garg, M. (2012). A Detailed Study of NHPP Software Reliability Models. *Journal of Software*, 7(6), 1296-1306.
- Li, Y.-F., Das, P., & Dowe, D. (2014). Two decades of Web application testing - A survey. *Information Systems*, 43, 20-54.
- Mendes, E. (2014). Web Development Versus Software Development. In *Practitioner's Knowledge Representation* (pp. 13-25). Springer Berlin Heidelberg.
- Murugesan, S., Deshpande, Y., Hansen, S., & Ginige, A. (2001). Web engineering: A new discipline for development of web-based systems. In *Web Engineering* (pp. 3-13). Springer Berling Heidelberg.
- Offutt, J. (2002). Web Software Applications Quality Attributes. *quality engineering in Software Technology (CONQUEST 2002)*, 187-198.
- Öztürk, M., Çavuşoğlu, Ü., & Zengin, A. (2015). A novel defect prediction method for web pages using k-means++. *Expert Systems with Applications*, 42, 6496-6506.
- Pham, H. (2003). Software Reliability and Cost Models: Perspectives, Comparison, and Practice. *European Journal of Operational Research*, 149(3), 475-489.
- Pham, H. (2006). *System Software Reliability*. Springer-Verlag London.
- Pham, H. (2007). An imperfect-debugging fault-detection dependent-parameter software. *International Journal of Automation and Computing*, 4(4), 325-328.
- Qian, Z., & Miao, H. (2011). Towards Testing Web Applications: A PFSM-Based Approach. *Advanced Materials Research*, 204, 220-224.
- Rana, R., Staron, M., Berger, C., Hansson, J., Nilsson, M., Törner, F., . . . Höglund, C. (2014). Selecting software reliability growth models and improving their predictive accuracy using historical projects data. *Journal of Systems and Software*, 98, 59-78.
- Robson, C. (2002). *Real World Research: A Resource for Social Scientists and Practitioner-Researchers* (Vol. 2). Oxford: Blackwell.

- Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2), 131-164.
- Torchiano, M., Ricca, F., & Marchetto, A. (2011). Are web applications more defect-prone than desktop applications? *International journal on software tools for technology transfer*, 13(2), 151-166.
- Ullah, N., Morisio, M., & Vetro, A. (2012). A comparative analysis of software reliability growth models using defects data of closed and open source software. *35th Annual IEEE Software Engineering Workshop (SEW)* (pp. 187-192). IEEE.
- Xie, M., Hong, G., & Wohlin, C. (2003). Modeling and Analysis of Software System Reliability. In W. Blischke, & D. Murthy (Eds.), *Case Studies in Reliability and Maintenance* (pp. 233-249). Wiley-Interscience.
- Yamada, S. (2014). *Software Reliability Modelling: Fundamentals and Applications*. Tokyo: Springer.
- Zhao, M., & Xie, M. (1996). On Maximum Likelihood Estimation for a General Non-homogeneous Poisson Process. *Scandinavian Journal of Statistics*, 597-607.