Approach to Predict the Software Reliability with different Methods

Ossmane Krini[1] and Josef Börcsök[2]

[1]Computer Architecture and System Programming, University of Kassel, Germany
o.krini@uni-kassel.de
[2]Computer Architecture and System Programming, University of Kassel, Germany
j.boercsoe@uni-kassel.de

Abstract

This particular essay expounds upon how one can foresee and predict software reliability. There are two major components that exist within a computer system: hardware and software. The reliabilities between the two are comparable because both are stochastic processes, which can be described by probability distributions. With this said, software reliability is the probability that will function without failure in a given software and in a given environment during a specified period of time. Thus, this is why software reliability is a major and key factor in software developmental processes and quality. However, one can spot the difference between software reliability and hardware reliability where it concerns the quality duration and the fact that software reliability does not decrease its reliability over time.

1. Introduction

Hardware and software have their faults, but they are different in what these faults actually are. Hardware consists of physical faults whereas the faults in software lay in the design of the actual software itself. This makes it more complicated to diagnose, classify, and or detect software faults within the system. This is because, as a major characteristic of software reliability, it tends to continuously change throughout and during test periods. Software vendors need to be ensured that their products are reliable before they are introduced to the market. Software-Reliability-Stochastic- Models (SRSM) help provide that information. SRSMs are designed to estimate or predict the number of failures. By looking at the requirements of program operation, one can see that 'failure' is the departure of external results. The term failure is associated to the behavioral aspects of the program.

In this particular industry, and even more specifically in software critical system, it is very important to produce highly reliable software, i.e. software with a low proportion of faults. A long testing and fault correction process is required to be able to produce reliable software. It is useful and time saving to use Software Reliability Stochastic Models to predict the software testing time, because this process can consume a large period of time and a substantial amount of resources to achieve the desired reliability results. The goal of this research summarizes and explains a detailed mathematical investigation of Software-Reliability-Stochastic- Models. This exert also presents how a stochastic approach based on non- homogeneous Poisson Process (NHPP) processes. A collection of Software Reliability Stochastic Models is also described throughout the essay. It is required, that to be able to use such model reliability prediction, that certain model parameters are using the failure data during the initial testing period. Below are the techniques used and described in this essay.

2. MATHEMATICAL BACKGROUND

2.1. Weibull-distribution

First, the Weibull-distribution will be explained. This distribution is one of the most commonly used because of its engineered reliability success of attaining various values of β (shape parameter). A great variety of data and life characteristics can therefore be modeled, [1]. The flexibility of the Weibull distribution is provided by the shape parameter. The Weibull distribution can also model a wider variety of data, when the value of the shape parameter is changed. The representation of these time dependent failure probabilities F(t) and their component are then made possible by this common distribution. One can find that for this to able to happen, one must find it necessary to posses the determined function parameters from observed data. These also have a technical important meaning in principle. It is then possible to determine whether one is dealing with early, random, or aging failures from the provided data. Failure frequency, number of all components and failure times of the components is the preferred required data. The independence of single component failures from each other can be assumed by The Weibull distribution because of its application of simple assumptions. Thus, the Weibull distribution is just a simplification of the Exponential distribution. Weibull distribution defines the probability of failure as

$$F(t) = W(t;\beta;t_0;T) = 1 - e^{-\left(\frac{t-t_0}{T-t_0}\right)^{\beta}} \qquad (1)$$

where t is the time, β is a shape parameter.
Weibull distribution's probability density function is given by

$$f(t) = W(t;\beta;t_0;T) = \frac{\beta}{T-t_0}\cdot\left(\frac{t-t_0}{T-t_0}\right)^{\beta-1}\cdot e^{-\left(\frac{t-t_0}{T-t_0}\right)^{\beta}} \qquad (2)$$

If $\beta = 1$, then the Weibull distribution is identical to the exponential distribution, if $\beta = 2$, the Weibull distribution is identical to the Rayleigh distribution; if β is between 3 and 4 the Weibull distribution approximates the normal distribution.

2.2. Exponential-distribution

In many applications as such in engineering, one can find the exponential distribution to be very useful. For example, it can be used to describe the life-time X of a transistor. The exponential distribution is perhaps the most well known and probably the most favorite probability for a reliability analysis of safety systems. It is then possible with this distribution to represent the time dependent probability F(t) of components, when it is necessary to obtain observed data to determine X. The exponential distribution defines the failure probability as

$$F(t) = 1 - e^{-\lambda \cdot t} \tag{3}$$

respectively with failure density

$$f(t) = \begin{cases} \lambda \cdot e^{-\lambda \cdot t} & for\ \ t \geq 0 \\ 0 & otherwise \end{cases} \tag{4}$$

2.3. Poisson-distribution

The Poisson distribution is a special case of a Binomial distribution, if the probability of occurrence p is very small and the number of experiments n is very large. The conditions under which the Poisson distribution holds are: Counts of rare events, all events are independent and the average rate does not change over the period of interest. The Poisson probability P(x) is given by

$$P(X = x) = \frac{\mu^x}{x!} \cdot e^{-\mu} \tag{5}$$

where μ is the mean rate of occurrence and x is the observed number of failure. Only one parameter μ in a Poisson distribution is needed to determine the probability of an event. The Expected number $E(x)$ is defined by the Poisson distribution as

$$E(x) = \sum_{x=0}^{\infty} x \cdot \frac{\mu^x}{x!} \cdot e^{-\mu} = \mu \cdot \sum_{k=1}^{\infty} \frac{\mu^{k-1}}{(k-1)!} \cdot e^{-\mu} = \mu \tag{6}$$

The variance σ^2 of the Poisson distribution is given by the formula below, If μ is the mean rate of successes occurring in a given time interval or region in the Poisson distribution, [2]:

$$\sigma^2 = \int_{-\infty}^{\infty} (x - E(x))^2 \cdot f(x)\ dx = \mu \tag{7}$$

2.4. Binomial-distribution

The binomial distribution belongs to the discrete distribution. It is explained with the following equation:

$$P(X \leq k) = \sum_{i=0}^{k} \binom{n}{i} \cdot p^i \cdot (1-p)^{n-i} \tag{8}$$

where X is the random variable, n is the number of experiments, k is the number of considered events or results and p is the probability of occurrence. The following equations define the variance σ^2 and the expected value $E(n,p)$ for the Binomial-distribution.

$$\sigma^2(n,p) = \sum_{x=0}^{n} (x - E(x))^2 \cdot f(x) = n \cdot p \cdot (1-p) \tag{9}$$

$$E(n,p) = \sum_{x=0}^{n} x \cdot f(x) = \sum_{x=0}^{n} x \cdot \binom{n}{x} \cdot p^x \cdot q^{n-x} = n \cdot p \tag{10}$$

A Specific feature of this distribution is that it can only accept two values, success or failure. That means the component failure is present $(X = 0)$ or is not present $(X = 1)$.

3. Stochastic Methods to Predict the Software-Reliability

The two most important types of distribution are Poisson and binomial distribution. They can also be called “macro”-distribution. Depending on how the failure quantity distribution is specified, the approaches can then be classified into types. Poisson type is the first type of the macro-distribution, [3]. The Poisson types have a Poisson process over time. Good approximation of occurrences in many real world events such as telephone calls, orders to factory, breakdown of machinery and arrivals on a queue are provided by the processes of the Poisson. However, there are a number of random failures with the Poisson-type models. Therefore each failure is considered a random variable when removed. The Poisson type has some important assumptions. Where the cumulative number of failure experienced at time t is $M(t)$. At time $t = 0$, $M(0) = 0$, there are no failures experienced. The probability that a failure will occur during $(t, t + \Delta t)$ is:

$$P(t, t + \Delta t) = {}^{s}\lambda(t)\Delta t + o(\Delta t) \tag{11}$$

where ${}^{s}\lambda(t)$ is the failure intensity function in the software and $o(\Delta t)$ is the probability that more than one failure is occurred . The probability $o(\Delta t)$ is defined as:

$$\lim_{\Delta t \to \infty} \frac{o(\Delta t)}{\Delta t} = 0 \tag{12}$$

The probability of exactly m failures that is occurring during the time interval $(0, t)$ for the Poisson type is:

$$P(M(t)=m)=\frac{[^{s}\mu(t)]^{m}}{m!}e^{(-^{s}\mu(t))} \qquad (13)$$

where ${}^{s}\mu(t)$ is mean value function in the software. The equation below shows the relationship between the failure rate ${}^{s}\lambda(t)$ and the mean value function ${}^{s}\mu(t)$, [4]:

$$E[M(t)=m]={}^{s}\mu(t)=\int_{0}^{t}{}^{s}\lambda(s)\,ds \qquad (14)$$

where $E[M(t)=m]$ is the expected value. The binomial type is the second important macro-distribution. The binomial type based off of several important assumptions. The fault that is caused will be removed instantly whenever a software failure occurs. There are u_0 inherent faults within the program. The hazard rates ${}^{s}z(t)$ for all faults are the same. The distribution of the number of failures experienced by time t is given by the binomial type as:

$$P(m)=\binom{u_0}{m}\cdot(F_a(t))^{m}\cdot(1-F_a(t))^{u_0-m} \qquad (15)$$

where u_0 is a fixed number of faults and F_a is the failure probability function. The mean value function ${}^{s}\mu(t)$ is defined as:

$$E[M(t)=m]={}^{s}\mu(t)=u_0F_a(t) \qquad (16)$$

The failure probability function is given as:

$$F_a(t)=1-e^{-\int_0^t z_a(\tau)\,d\tau} \qquad (17)$$

where $z_a(\tau)$ is the hazard rate.

3.1. Stochastic Approach based on Weibull model

The Weibull-model is one of the most widely used models for hard- and software reliability. It is based on binomial type (macro-distribution). There is also a special feature to this model. Because of the great flexibility that expressed throughout the model parameters, one can tell how each failure density can be positive, negative, or even remain constant. From the start of the observation time of the software, there are a fixed number (N) of faults. The time to failure of fault a, is distributed as a Weibull distribution with parameter β and δ. The density function $f_a(t)$ for the Weibull-model is defined as:

$$f_a(t)=\beta\cdot\delta\cdot t^{\beta-1}\cdot e^{\left(-\delta\cdot t^{\beta}\right)} \qquad (18)$$

where $\beta,\delta>0$ and $t\geq 0$. The per-fault hazard rate $z_a(t)$ is given by:

$$z_a(t)=\beta\cdot\delta\cdot t^{\beta-1} \qquad (19)$$

With the assistance of equation (19), it can be seen that if $0<\beta<1$, the per-fault hazard rate $z_a(t)$ is decreasing with respect to time. If the shape parameter equals 1, the per-fault hazard rate $z_a(t)$ is constant. $z_a(t)$ may also increase if the shape parameter $\beta>1$. The conditional hazard rate $z(t|t_{i-1})=(N-i+1)\cdot\beta\cdot\delta\cdot(t+t_{i-1})^{\beta-1}$, with $0<\beta<1$ is presented in Fig. 1. Because of the power function component, the effect on the hazard rate decreases with time.

The per fault distribution $f_1, f_2,..., f_n$ are the number of faults which are detected in each of the respective intervals $[(t_0,t_1),(t_1,t_2),..., (t_{i-1},t_i),..., (t_{n-1},t_n)]$ and therefore are not dependent for any finite collection of times. One needs the fault counts in each of testing intervals f_i's for the determination of the probability of failure. The completion time of each period is also needed while the software is under observation f_i's. The failure intensity function ${}^{s}\lambda(t)$ for the Weibull model is given by:

$${}^{s}\lambda(t)=N\cdot\beta\cdot\delta\cdot t^{\beta-1}\cdot e^{-\delta\cdot t^{\beta}} \qquad (20)$$

where β is shape parameter and N is a total number of faults in the system by time $t=0$. The distribution f_a becomes the exponential if the shape parameter is $\beta\leq 1$. The failure intensity is Rayleigh distributed if the shape parameter is. Fig. 2 shows the failure intensity function with different shape parameters.

With the help of Equation (16), the mean value function ${}^{s}\mu(t)$ for the Weibull-model is defined as:

$${}^{s}\mu(t)=N\cdot(1-e^{-\delta\cdot t^{\beta}}) \qquad (21)$$

where ${}^{s}\mu(t)$ is the mean value function for the software.

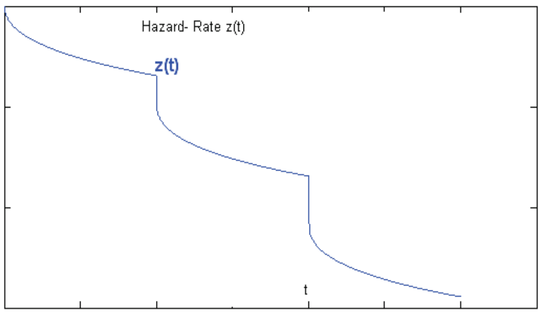

Fig. 1. Hazard rate function for the Weibull model

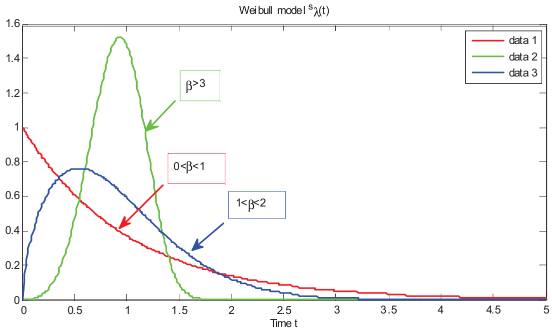

Fig. 2. Failure intensity function ${}^{s}\lambda(t)$ for the Weibull model with different shape parameters

Provided in the following formulas is the relationship between the failure probability ${}^{s}F(t)$ and the software reliability ${}^{s}R(t)$:

$$
\begin{aligned}
&{}^{s}R(t) + {}^{s}F(t) = 1 \\
&{}^{s}R(t) = 1 - {}^{s}F(t) = e^{-\delta \cdot t^{\beta}}
\end{aligned}
\tag{22}
$$

Fig. 3 and 4 shows the mean value function ${}^{s}\mu(t)$ and the software reliability ${}^{s}R(t)$. The mean value function is non-decreasing. The Weibull-model is a finite model, because ${}^{s}\mu(t \to \infty) = N$.

3.2. Stochastic Approach based on Poisson model

Real-world situations can be efficiently modeled by the Poisson process. There is not a fixed number N of total faults which are considered binomial for the Poisson type. However, for the sake of using the Poisson type, the total failure is a random variable with mean ω_0. Because it is more reflective of the actual stress induced on the software system, this model is practically based off of execution time τ. A Poisson random variable with a mean of ω_0 is the total number of faults remaining in the program at $\tau = 0$, [5]. So whenever a software failure occurs, the fault that caused it will be removed instantaneously. The hazard rate $z_a(\tau) = \Phi$ for a single fault is constant, see Fig. 5. Failures are independent. With the expected number of failures experienced, one can see the failure intensity function ${}^{s}\lambda(\tau)$ decreases exponentially. Following the Poisson process, he cumulative number of failure by time τ, $M(\tau)$. The mean value function ${}^{s}\mu(\tau)$ increases and reaches a finite value , therefore making the exponential-model a finite model. The expected number of failure occurrences for any time period is proportional to the expected number of undetected faults at that time is the mean value function. By assuming that the fault correction rate is proportional to the hazard rate, the fault removal process is then characterized on an average basis.

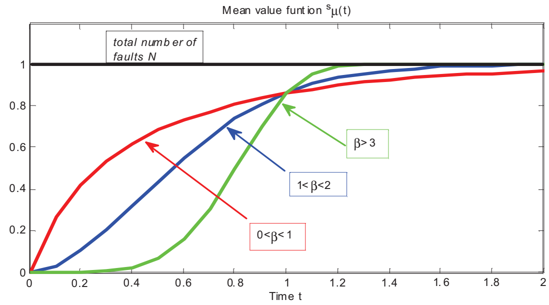

Fig. 3. Mean value function ${}^{s}\mu(t)$ for the Weibull model with different shape parameters

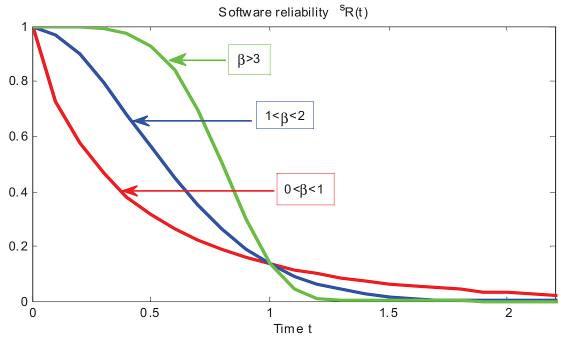

Fig. 4. Software reliability function ${}^{s}R(t)$ for the Weibull model with different shape parameters

This proportionality constant called a fault reduction factor B.

The failure rate ${}^{s}\lambda(\tau)$ is defined as

$$
{}^{s}\lambda(\tau) = \upsilon_0 \cdot \Phi \cdot B \cdot e^{-\Phi \cdot B \cdot \tau} \tag{23}
$$

where υ_0 is the number of failures, Φ is the hazard rate and B is the fault reduction factor. With help of equation (14), it follows the mean value function ${}^{s}\mu(\tau)$ according to:

$$
{}^{s}\mu(\tau) = \upsilon_0 \cdot (1 - e^{-\Phi \cdot B \cdot \tau}) \tag{24}
$$

Figures 6 and 7 illustrate the relationship. The failure intensity function ${}^{s}\lambda(\tau)$ is decreasing with execution time. Because ${}^{s}\mu(\tau)$ is a cumulative function, The mean value function is non-decreasing.

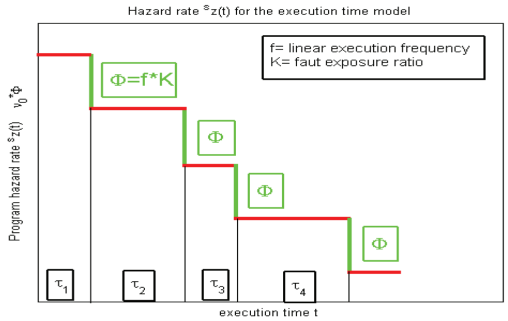

Fig. 5. Program Hazard rate ${}^{S}z(t)$ and the per-fault hazard rate for single fault ${}^{S}z_a(t) = \Phi$

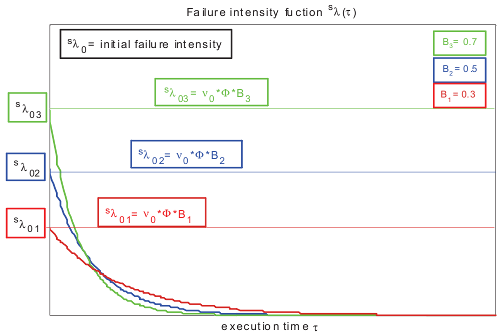

Fig. 6. Failure intensity functions ${}^{S}\lambda(t)$ in the software with different reduction factor.

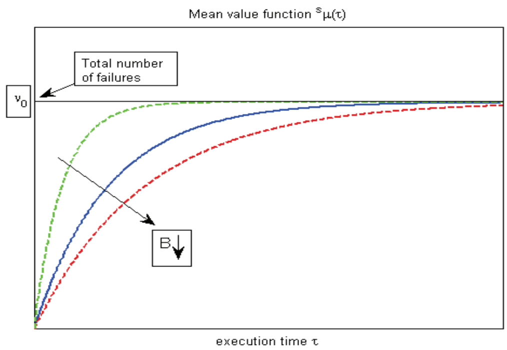

Fig. 7. Mean value function ${}^{S}\mu(t)$ in the software

By substituting Equation (24) into Equation (13), it gives the cumulative probability distribution of time to the i th failure:

$$P[T_i \le \tau] = e^{\left\{-\upsilon_0\left[1-e^{(-\Phi\cdot B\cdot\tau)}\right]\right\}} \cdot \sum_{j=i}^{\infty} \frac{\upsilon_0^j\left[1-e^{(-\Phi\cdot B\cdot\tau)}\right]^j}{j!} \qquad (25)$$

Therefore, the conditional reliability function ${}^{S}R(\tau_i|\tau_{i-1})$ after $(i-1)$ failures are given by the following formula:

$${}^{S}R(\Delta\tau|\tau_{i-1}) = e^{\left\{-\left[\upsilon_0\cdot e^{(-\Phi\cdot B\cdot\tau_{i-1})}\right]\cdot\left[1-e^{(-\Phi\cdot B\cdot\Delta\tau)}\right]\right\}} \qquad (26)$$

4. Theory of Estimation

It should be taken into account that the parameters of reliability models should be estimated. The primary importance in software reliability prediction is the parameter estimation. Maximum likelihood estimation (MLE) is one of the more popular techniques for point estimation. We must obtain the most probable values of the parameters for a given distribution, because this is the basic idea behind MLE. This will best describe the data that is provided. When the underlying distributions of data are known or specified, then the general technique that is used is the estimation method, [6]. The product of the probability density function that is evaluated at each sample point is the maximum likelihood function, $L(X;\psi)$. By maximizing $L(X;\psi)$ with the respect to ψ, the maximum likelihood estimator ψ can then be found. The log likelihood function is given by the following equation:

$$\ln L(X;\psi) = \sum_{i=1}^{n} \ln f(X_i;\psi) \qquad (27)$$

5. Conclusions

In this paper, detailed overviews of two software reliability approaches were provided. Different types, such as the Weibull type which is based on binomial and the exponential typed which is based on the Poisson model, were described, illustrated, and explained. The paper also described the assumptions of the different approaches. Different distributions, which are also important for reliability analysis, were explained.

6. References

[1] J. Börcsök, 2004. Electronic Safety Systems, Hardware Concepts, Models, and Calculation. Heidelberg, Germany 2004.

[2] J.D. Musa, Software Reliability Engineering 2004. More Reliable Software Faster and Cheaper. United States of America 2004.

[3] N. F. Schneidewind, Software Reliability Engineering 1992. Minimizing risk in Applying Metrics on Multiple Projects, Proc. IEEE Int. Symp, pp. 173-179.

[4] J. Börcsök, O. Krini, 2009. Principle Software Reliability Analysis with different Failure Rate Models. IEEE Int. Symp. ICAT 2009

[5] B. Lindqvist, A. Doksum, 2003. Mathematical and Statistical Method in Reliability. Covent Garden, London 2003.

[6] M. Takahashi and Y. Kamayachi, IEEE Computer Society, 1991. An Empirical Study of a Model for Program Error Prediction, in Software Reliability Models. pp. 71-77.