Original Research Article

Chen Software Reliability Growth Model

Abstract

Software reliability analysis is very vital in software development. Software manufacturers assess the quality of their developed software through this analysis. This has triggered the development of reliability models. Software reliability growth models have been used extensively to examine the quality of manufactured software before they are sent to the market. This study presents a new software reliability growth model using Chen distribution. The Chen software reliability growth model is then used to establish sequential probability ratio test limits for determining whether a manufactured software is reliable or unreliable. The applications of the proposed model revealed that it performs better than some of the existing software reliability growth models for the given datasets.
Keywords: Software reliability, Growth model, Chen, SPRT, Quality control.

1 Introduction

Quality control in the production of goods and services come up handy on regular bases due to the fact that the world is evolving into a global village (Schonberger, 2008). More producers have products and goods they need to send to the market with each passing day. Thus, when customers realize these products are not of high quality, demand of products decrease (Stalk, 1988). Quality control is one of the most important ways of examining quality for products before they are sent to the market (Feigenbaum, 1991). Products come in the form of tangible and intangible goods (Levitt, 1981). Ranging from tangible products such as clothing, food , building materials and so on, to intangible products such as software. Software are a set of instructions that teach a computer what to do (Rodrigues, 1985). In today's economy, software quality is becoming increasingly critical. This is due to the fact that the advantages of its usage outweighs the detrimental aspect of it (Michael and Miller, 2013). Spanning from performing basic mathematics to performing complex computations, software make living easier. According to ANSI Standard (ANSI/ASQC A3/1978): "Quality is the totality of features and characteristics of a product or a service that thrives on its ability to satisfy the given needs". Users of software are aware of software as a tool for assisting them in their endeavors whiles producers see software as a means yielding profits (Jamwal et al., 2009). Software reliability growth models (SRGM) are implementations of statistical methods as a quality control technique, in verifying the quality and durability of a software. These models are able to model various forms of failure data, by collecting information of the behavior of software over a period. It stems from employing techniques such as the Non-homogeneous Poisson Process (NHPP), the Sequential Probability Ratio testing (SPRT) and so on. During the production of software, the monitoring process is the phase where the performance of the software is tracked. The SPRT provides a rule for stopping or continuing the monitoring process of a software during production. It gives a general perspective of whether a software should be deployed to the market or not. This is done by comparing the limits of the SPRT against cumulative number of failures at a given time (this could be number of hours, days, weeks, months or years). This study seeks to propose a Chen SRGM that incorporates environmental factors and the limits of its SPRT. The remainder of the paper is organized as follows: Section two presents the SPRT for the NHPP, Section three presents the proposed Chen SRGM, Section four presents the application of the proposed model and lastly Section five presents the conclusion of the study.

2 Sequential Probability Ratio Test for NHPP

Sequential sampling is a sampling technique where items are drawn one at a time and the consequent result at every stage determines if sampling or testing should stop. Hence, any sampling procedure where the number of observations is a random variable can be regarded as sequential sampling. The term sequential test has its name of origin from the fact that the sample size is not determined in advance but allowed to "float" with a decision as to whether it should be accepted, rejected, or the test must continue after each trial or data point (Anderson, 1960).
SPRT is a ratio test method that follows the sequential sampling process. The SPRT is similar to the Maximum Likelihood Estimation used for constructing tests. It is formulated by taking the ratio of the sample densities under H_1 over H_0. Consider the NHPP $\{N(t), t \in [0, \infty)\}$. Suppose we are interested in testing the hypothesis in terms of the mean value function $m(t)$:
$H_0 : m(t) = m_0(t)$ versus $H_1 : m(t) = m_1(t) > m_0(t)$. The probability distribution when H_0 is true can be formulated as,

$$P_0 = \frac{[m_0(t)]^{N(t)}}{N(t)!} e^{-m_0(t)}. \tag{1}$$

Then, when H_1 is true, the probability distribution can be formulated as,

$$P_1 = \frac{[m_1(t)]^{N(t)}}{N(t)!} e^{-m_1(t)}. \tag{2}$$

Suppose X and Y are two positive constants where $X > Y$. Then at every stage of the SPRT, the likelihood ratio (LR), $LR = \frac{P_0}{P_1}$ is estimated. Thus, based on the value of the likelihood ratio, the following decisions are arrived at:

1. Stop sampling and reject H_0 as soon as $LR \geq X$.
2. Stop sampling and accept H_0 as soon as $LR \leq Y$.
3. Continue sampling as long as $Y < LR < X$.

The choice of selection of the estimates of X and Y are chosen such that the SPRT has the needed strength (α, β). Thus, with the above test, suggested by Wald (1945) X and

Y can be expressed as follows: $Y = \frac{\beta}{1-\alpha}$ and $X = \frac{1-\beta}{\alpha}$. The basis for α and β respectively are, therefore,

$$P[LR > X|H_0] = \alpha \tag{3}$$

and

$$P[LR < X|H_0] = \beta, \tag{4}$$

where α and β are the type I and type II errors respectively. To make computation more simplified the logarithm form of the likelihood ratio is used often, that is $\log Y < \log \frac{P_0}{P_1} < \log X$. Hence, using the SPRT in reliability analysis of software that follows the NHPP, the following decisions may be arrived at: the given software is reliable, the software is unreliable or the testing process should be continued by adding more observations in the failure data.

The likelihood ratio $\frac{P_0}{P_1}$ at a given time t is considered as an estimate for a decision to take with regards to $m_0(t)$ or $m_1(t)$, for a sequence of time $t_1 < t_2 \cdots < t_n$ with their corresponding realizations $N(t_1), N(t_2), ..., N(t_n)$. By using the likelihood ratio, a software failure data which follows a NHPP is reliable if,

$$\frac{P_0}{P_1} = \frac{e^{-m_1(t)}[m_1(t)]^{N(t)}}{e^{-m_0(t)}[m_0(t)]^{N(t)}} \leq Y. \tag{5}$$

Implying that,

$$N(t) \leq \frac{\log(\frac{\nu}{1-\alpha}) + m_1(t) - m_0(t)}{\log(\frac{m_1(t)}{m_0(t)})}. \tag{6}$$

A software failure data which follows a NHPP is unreliable if,

$$\frac{P_0}{P_1} = \frac{e^{-m_1(t)}[m_1(t)]^{N(t)}}{e^{-m_0(t)}[m_0(t)]^{N(t)}} \geq X. \tag{7}$$

Implying that,

$$N(t) \geq \frac{\log(\frac{1-\beta}{\alpha}) + m_1(t) - m_0(t)}{\log(\frac{m_1(t)}{m_0(t)})}. \tag{8}$$

The testing process is continued as long as,

$$\frac{\log(\frac{\beta}{1-\alpha}) + m_1(t) - m_0(t)}{\log(\frac{m_1(t)}{m_0(t)})} < N < \frac{\log(\frac{1-\beta}{\alpha}) + m_1(t) - m_0(t)}{\log(\frac{m_1(t)}{m_0(t)})}. \tag{9}$$

3 The Proposed Chen SRGM

Software developers often times ignore the effects environmental conditions could have on their software. When the location of the software is changed from its original environment, this could result in a type I or II error (Basili and Weiss, 1984).

Pham (2014) proposed a generalized NHPP incorporating the uncertainty of environmental conditions with all existing assumptions of the NHPP in place and the following assumptions;

i. The occurrence of software failures follows an NHPP.

ii. Software can fail during execution, caused by faults in the software.

iii. The software failure detection rate at any time is proportional to the number of remaining faults in the software at that time.

iv. When a software failure occurs, a debugging effort removes the faults immediately.

v. For each debugging effort, regardless of whether the faults are successfully removed, some new faults may be introduced into the software system.

vi. The environment affects the unit failure detection rate, $c(t)$, by multiplicative factor ω.

Thus a generalized NHPP SRGM can be obtained as,

$$\frac{dm(t)}{dt} = \omega[c(t)(N - m(t))] \tag{10}$$

where ω represents the uncertainty of the system fault detection rate in the operating environments with a probability density function f, N is the expected number of faults that exists in the software before testing, $c(t)$ is the fault detection rate function, which also represents the average failure rate of a fault, and $m(t)$ is the mean value function (the expected number of errors detected by time t). Solving the differential equation from equation (10) and setting the initial condition of $m(0) = 0$ results in,

$$m(t) = \int_{\omega} N(1 - e^{-\gamma \int_0^t b(t)dt})dg(\omega). \tag{11}$$

The mean value function can be derived from equation (11) given that the random variable ω has a generalized probability density function g with two parameters ρ and τ, and is given by,

$$m(t) = N\left(1 - \frac{\tau}{\tau + \int_0^t b(s)ds}\right)^{\rho}, \tag{12}$$

where $b(s)$ is the fault detection rate (Pham, 2003).
Now, assuming the underlying distribution of the failure time data is the Chen distribution and its fault rate is given as,

$$b(t) = \gamma\nu s^{\nu-1}e^{t^{\nu}}, t > 0, \theta > 0, \gamma > 0, \nu > 0.$$

Let the expected number of failures in the software be N, then the mean value function when the environmental uncertainties are factored in can be given as,

$$N\left(1 - \frac{\tau}{\tau + \gamma(e^{t^{\nu}} - 1)}\right)^{\rho}, \tag{13}$$

where γ, τ, ρ and ν are positive parameters of the mean value function.

3.1 SPRT Limits for Chen SRGM

This section presents the limits of the SPRT of the Chen SRGM. Assuming our interest is in testing the null hypothesis $H_0 : m(t) = m_0(t)$ against the alternative hypothesis $H_1 : m(t) = m_1(t)(> m_0(t))$. Suppose the focus parameters here is γ, then for a Chen

SRGM which follows the NHPP having a mean value function given in equation (13), the probability distribution when H_0 is true and when H_1 is true are respectively given as,

$$P_0 = \frac{exp\left(-N\left(1-\frac{\tau}{\tau+\gamma_0(e^{t^\nu}-1)}\right)^\rho\right)\left(\left(1-\frac{\tau}{\tau+\gamma_0(e^{t^\nu}-1)}\right)^\rho\right)^{N(t)}}{N!}, \tag{14}$$

and

$$P_1 = \frac{exp\left(-N\left(1-\frac{\tau}{\tau+\gamma_1(e^{t^\nu}-1)}\right)^\rho\right)\left(\left(1-\frac{\tau}{\tau+\gamma_1(e^{t^\nu}-1)}\right)^\rho\right)^{N(t)}}{N!}, \tag{15}$$

where the parameter specifications are; for γ is γ_1 and γ_0. Therefore, the acceptance region for a reliable software with the Chen mean value function is established as,

$$N(t) \leq \frac{\log(\frac{\beta}{1-\alpha}) + N\left(\left(1-\frac{\tau}{\tau+\gamma_1(e^{t^\nu}-1)}\right)^\rho - \left(1-\frac{\tau}{\tau+\gamma_0(e^{t^\nu}-1)}\right)^\rho\right)}{\log\frac{\left(1-\frac{\tau}{\tau+\gamma_1(e^{t^\nu}-1)}\right)^\rho}{\left(1-\frac{\tau}{\tau+\gamma_0(e^{t^\nu}-1)}\right)^\rho}}.$$

Similarly, for unreliable software, the rejection region is given by,

$$N(t) \geq \frac{\log(\frac{1-\beta}{\alpha}) + N\left(\left(1-\frac{\tau}{\tau+\gamma_1(e^{t^\nu}-1)}\right)^\rho - \left(1-\frac{\tau}{\tau+\gamma_0(e^{t^\nu}-1)}\right)^\rho\right)}{\log\frac{\left(1-\frac{\tau}{\tau+\gamma_1(e^{t^\nu}-1)}\right)^\rho}{\left(1-\frac{\tau}{\tau+\gamma_0(e^{t^\nu}-1)}\right)^\rho}}. \tag{16}$$

The continuation region is given by,

$$\frac{\log(\frac{\beta}{1-\alpha}) + N\left(\left(1-\frac{\tau}{\tau+\gamma_1(e^{t^\nu}-1)}\right)^\rho - \left(1-\frac{\tau}{\tau+\gamma_0(e^{t^\nu}-1)}\right)^\rho\right)}{\log\frac{\left(1-\frac{\tau}{\tau+\gamma_1(e^{t^\nu}-1)}\right)^\rho}{\left(1-\frac{\tau}{\tau+\gamma_0(e^{t^\nu}-1)}\right)^\rho}} < N(t) < \frac{\log(\frac{1-\beta}{\alpha}) + N\left(\left(1-\frac{\tau}{\tau+\gamma_1(e^{t^\nu}-1)}\right)^\rho - \left(1-\frac{\tau}{\tau+\gamma_0(e^{t^\nu}-1)}\right)^\rho\right)}{\log\frac{\left(1-\frac{\tau}{\tau+\gamma_1(e^{t^\nu}-1)}\right)^\rho}{\left(1-\frac{\tau}{\tau+\gamma_0(e^{t^\nu}-1)}\right)^\rho}}.$$

4 Application of the Chen SRGM

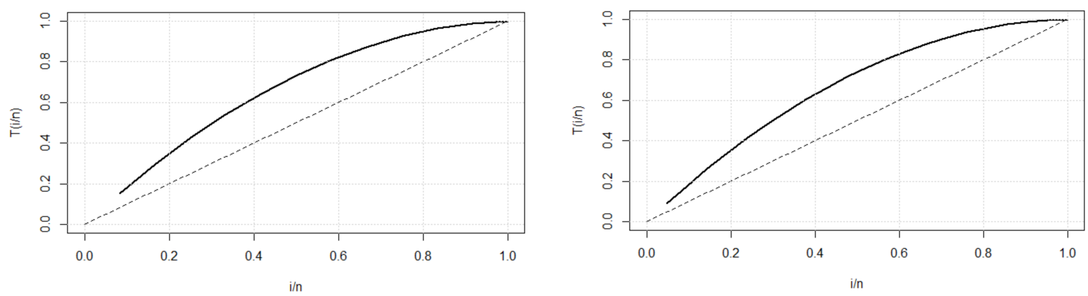
This section presents the application of the Chen SRGM in real life using two data sets. The first data set was reported by Zhang and Pham (2002) based on system test data for a telecommunication system data (TSD). System test data consisting of two releases, that is Phase 1 and Phase 2. In both tests, automated test and human involved tests are executed on multiple test beds. Phase two was considered in this study.
The second data set employed yields from a modest on-line data entry software program available in Japan since 1980 (Ohba, 1984). According to the International Business Machines (IBM) journal, the software was about 40,000 lines of code long. The duration

of the testing was calculated using the amount of shifts spent executing test cases and evaluating results of the matched couples
It is very important to explore the behavior of the failure rate of the data sets employed in this study. Table 1 shows descriptive statistics of the failure rate of the TSD-Phase two and IBM data sets. They both have a minimum of 1 and a maximum of 21 weeks. They both have negative excess kurtosis that indicates the distribution of the failure rate is flat with thin tails. They also both have a skewness of 0 that indicates the failure rate is perfectly symmetrical.

Table 1: **Descriptive Statistics of the TSD-Phase Two and IBM data**

Statistic	**TSD-Phase two data**	**IBM data**
Minimum	1.00	1.00
Maximum	21.00	21.00
Mean	11.00	11.00
Median	11.00	11.00
Standard deviation	6.205	6.205
Skewness	0	0
Kurtosis	-1.37	-1.37

The total test on time (TTT) plot is employed to investigate the empirical behavior of the data sets. Figure 1 presents the failure rate of both data sets. It shows that they both have an increasing failure since the curve is concave above the 45^0 line.



TSD-Phase Two **IBM**

Figure 1: Plot of TTT of the TSD-Phase Two and IBM failure rate

Tables 2 and 3 summarizes the least square estimates of parameters for the TSD-Phase two and the IBM data sets respectively for the Chen SRGM and the other existing competing SRGMs. The estimated parameters of the Chen SRGM are all significant at 5%. This can be verified by using the *p-values* of the parameters.

Table 2: **Estimates of parameters of fitted models for TSD-Phase two data set**

SRGM	**Parameters**	**Estimates**	**Standard error**	Z value	**p-value**
GO	$\hat{a}$	3.2305×10^{2}	9.5361×10^{-10}	3.38771×10^{11}	$2.2 \times 10^{-16*}$
	$\hat{b}$	6.8853×10^{-3}	4.2424×10^{-5}	1.6230×10^{2}	$2.2 \times 10^{-16*}$
DS	$\hat{a}$	67.2835630	2.0357850	33.050	$2.2 \times 10^{-16*}$
	$\hat{b}$	0.1089428	0.0032053	33.988	$2.2 \times 10^{-16*}$
IS	$\hat{a}$	44.357603	0.693023	64.006	$2.2 \times 10^{-16*}$
	$\hat{b}$	0.296139	0.013283	22.294	$2.2 \times 10^{-16*}$
	$\hat{\beta}$	22.674540	2.911473	7.788	$6.808 \times 10^{-15*}$
YID	$\hat{a}$	0.214698	0.026390	8.1355	$4.102 \times 10^{-16*}$
	$\hat{b}$	0.397253	0.063547	6.2513	4.070×10^{-16}
	$\hat{\alpha}$	11.611068	1.424200	8.1527	$3.559 \times 10^{-16*}$
DPM	$\hat{\rho}$	444.36	1.3159×10^{-8}	3.3768×10^{10}	$2.2 \times 10^{-16*}$
	$\hat{\gamma}$	-3.4406	2.1050	-163.45	$2.2 \times 10^{-16*}$
Ohba	$\hat{a}$	44.355318	0.692953	64.0092	$2.2 \times 10^{-16*}$
	$\hat{\beta}$	22.689046	2.915186	7.7831	$7.08 \times 10^{-15*}$
	$\hat{b}$	0.296201	0.013291	22.2862	$2.2 \times 10^{-16*}$
VTS	$\hat{\rho}$	3.4988×10^{-1}	9.3259×10^{-2}	3.7517	0.0001757
	$\hat{a}$	1.0011×10^{-2}	2.378×10^{-4}	4.2086×10^{-5}	$2.2 \times 10^{-16*}$
	$\hat{b}$	5.0000	5.9218×10^{-17}	8.4433×10^{16}	$2 \times 10^{-16*}$
	$\hat{\nu}$	99.886	2.3604×10^{-4}	4.2317×10^{5}	2×10^{-16}
TC	$\hat{\rho}$	6.8986×10^{-1}	3.3148×10^{-5}	2.0811×10^{4}	$2.2 \times 10^{-16*}$
	$\hat{a}$	2.4868×10^{-3}	1.3072×10^{-4}	1.9023×10^{1}	$2.2 \times 10^{-16*}$
	$\hat{b}$	1.1893	1.8847×10^{-2}	6.3101×10^{1}	$2 \times 10^{-16*}$
	$\hat{\nu}$	10.838	3.2230×10^{-6}	3.3626×10^{6}	2×10^{-16}
	$\hat{N}$	7.9735×10^{3}	7.9101×10^{-9}	1.0080×10^{12}	$2 \times 10^{-16*}$
Cheng	$\hat{\rho}$	77.554	7.3665×10^{-5}	1.0528×10^{6}	$2.2 \times 10^{-16*}$
	$\hat{\beta}$	24.119	6.9397×10^{-6}	3.4756×10^{6}	$2.2 \times 10^{-16*}$
	$\hat{b}$	2.5562	1.9032×10^{-1}	13.431	$2 \times 10^{-16*}$
	$\hat{a}$	0.10795	2.6385×10^{-3}	40.914	2×10^{-16}
Chen SRGM	$\hat{\rho}$	0.48073	4.9985×10^{-2}	9.6175	$2.2 \times 10^{-16*}$
	$\hat{\gamma}$	9.6184×10^{-2}	4.5404×10^{-2}	2.1184	0.03414
	$\hat{\nu}$	0.77355	2.0186×10^{-2}	38.3208	$2.2 \times 10^{-7*}$
	$\hat{\tau}$	1.6402×10^{2}	2.5666×10^{-3}	63907.0738	$2.2 \times 10^{-16*}$
	$\hat{N}$	4.2713	0.6325	67.5342	$2.2 \times 10^{-16*}$

*: means significant at the 5% significance level

Table 3: **Estimates of parameters of fitted models for IBM data**

SRGM	Parameters	Estimates	Standard error	Z value	p-value
GO	$\hat{a}$	23.190	1.543×10^{-1}	1.5029×10^{2}	$2.2 \times 10^{-16*}$
	$\hat{b}$	2.1101×10^{2}	2.6257×10^{-18}	8.0363×10^{19}	$2.2 \times 10^{-16*}$
DS	$\hat{a}$	23.190	1.5430×10^{-1}	1.5029×10^{2}	$2.2 \times 10^{-16*}$
	$\hat{b}$	1.4040×10^{2}	1.9559×10^{19}	7.1784×10^{19}	$2.2 \times 10^{-16*}$
IS	$\hat{a}$	23.190	0.15430	150.29	$2.2 \times 10^{-16*}$
	$\hat{b}$	100.70	5.6227×10^{-19}	1.7910×10^{20}	$2.2 \times 10^{-16*}$
	$\hat{\beta}$	10.00	$5.6620 times 10^{-18}$	7.788×10^{18}	$2.2 \times 10^{-16*}$
YID	$\hat{a}$	0.21873	0.02963	7.3820	$1.559 \times 10^{-13*}$
	$\hat{b}$	0.34298	0.07076	4.8471	1.253×10^{-6}
	$\hat{\alpha}$	12.14272	1.55468	7.8104	$5.699 \times 10^{-15*}$
DPM	$\hat{\rho}$	486.00	1.0010×10^{-8}	4.8551×10^{10}	$2.2 \times 10^{-16*}$
	$\hat{\gamma}$	-3.2227×10^{-2}	1.7530×10^{-4}	-183.84	$2.2 \times 10^{-16*}$
Ohba	$\hat{a}$	23.190	0.15430	150.29	$2.2 \times 10^{-16*}$
	$\hat{\beta}$	1.0000×10^{-3}	3.9086×10^{-13}	2.5584×10^{9}	$7.08 \times 10^{-15*}$
	$\hat{b}$	1.0046×10^{2}	3.8906×10^{-18}	$2.5822 times 10^{19}$	$2.2 \times 10^{-16*}$
VTS	$\hat{\rho}$	3.4988×10^{-1}	9.3259×10^{-2}	3.7517	0.0001757
	$\hat{a}$	1.0011×10^{-2}	2.378×10^{-4}	4.2086×10^{-5}	$2.2 \times 10^{-16*}$
	$\hat{b}$	5.0000	5.9218×10^{-17}	8.4433×10^{16}	$2 \times 10^{-16*}$
	$\hat{\nu}$	99.886	2.3604×10^{-4}	4.2317×10^{5}	2×10^{-16}
TC	$\hat{\rho}$	6.4619×10^{-2}	1.2030×10^{-2}	5.3716	$7.805 \times 10^{-8*}$
	$\hat{a}$	1.5643×10^{-1}	1.3342×10^{-2}	1.1725×10^{1}	$2.2 \times 10^{-16*}$
	$\hat{b}$	1.6590	8.3058×10^{-2}	$1.9974e \times 10^{1}$	$2 \times 10^{-16*}$
	$\hat{\nu}$	1.4149	1.4184×10^{-4}	9.9755×10^{4}	2×10^{-16}
	$\hat{N}$	7.6813×10^{2}	1.6073×10^{-5}	4.7790×10^{7}	$2 \times 10^{-16*}$
Cheng et al	$\hat{\rho}$	55.386	1.5073×10^{-4}	3.6745×10^{5}	$2.2 \times 10^{-16*}$
	$\hat{\nu}$	1.7364×10^{-1}	7.7813×10^{-6}	2.2316×10^{6}	$2.2 \times 10^{-16*}$
	$\hat{b}$	2.6857	1.9466×10^{-1}	1.3797	$2 \times 10^{-16*}$
	$\hat{a}$	0.12041	3.5473×10^{-3}	33.944	2×10^{-16}
Chen SRGM	$\hat{\rho}$	3.0728	4.9905×10^{-2}	61.574	$2.2 \times 10^{-16*}$
	$\hat{\gamma}$	20.688	4.5292×10^{-3}	4.5678×10^{3}	0.03414
	$\hat{\nu}$	0.38282	2.4930×10^{-3}	1.5356×10^{-2}	$2.2 \times 10^{-7*}$
	$\hat{\tau}$	1.6284×10^{2}	5.7543×10^{-4}	2.8299×10^{5}	$2.2 \times 10^{-16*}$
	$\hat{N}$	1.1604×10^{2}	1.9592×10^{-5}	5.9226×10^{6}	$2.2 \times 10^{-16*}$

*: means significant at the 5% significance level

Three model selection criteria are used to compare the Chen SRGM and other existing SRGM for the data sets. This is to help identify which model fits the data sets better. For the three criteria, the smaller the value, the closer the model fits relative to other models (Sharma et al., 2010).

1. **The Akaike information criterion (AIC)**
   The AIC is a model selection criterion test used to evaluate how well a model fits the data it is meant to describe. The test statistic is given by,
   $$\text{AIC} = -2\log L(\hat{\theta}) + 2k,$$
   where k is the number of estimated parameters for the model.

2. **Bayesian Information Criterion (BIC)**
   The BIC is a model selection criterion that employs the Bayes factor assumption. It is defined as,
   $$\text{BIC} = -2\log L(\hat{\theta}) + k\log(n),$$
   where n is the sample size.

3. **Predictive Ratio Risk (PRR)**
   The PRR indicates the distance of model estimates from actual data with respect to the model estimates. Its test statistic is given as,
   $$\text{PRR} = \sum_{i=1}^{n}\left(\frac{\hat{m}(t_i) - y_i}{\hat{m}(t_i)}\right)^2.$$

Tables 4 and 5 shows the AIC, BIC, and PRR of the fitted models for the TSD-Phase two and the IBM data set respectively. Since the Chen SRGM has the smallest AIC, BIC and PRR values for both data sets, it means that the Chen SRGM performs better than the other existing models.

Table 4: **Model selection criteria for fitted models for TSD-Phase two data set**

Model	**AIC**	**BIC**	**PRR**
Goel Okumoto	599.229	601.318	762.2693
Delayed Shaped	314.666	316.7551	14106.45
Infection S-shaped	213.5258	216.6594	1382.98
Yamada imperfect debugging	395.0934	398.227	203.7502
Testing coverage	439.8727	445.09534	113.5076
Dependent paramter model	247.4809	249.57	232.6096
Ohba	213.5258	216.6594	192.7347
Vtub-shaped fault detection rate	11640	11644.18	353.0258
Cheng	10923.01	10927.19	7960.853
Chen SRGM	197.1243	202.3469	77.22079

4.1 Sensitivity Analysis

In understanding the effect of the model parameters on the model, sensitivity analyses were carried out on the two data sets. Tables 5 and 6 shows that the parameters of Chen SRGM which vary the most are τ and γ for the TSD-Phase two and IBM data sets respectively. This means that τ and γ have a higher effect on the model than the other parameters.

Table 5: **Sensitivity analysis of the model paramters of the TDS-Phase two data set**

Parameter	-30%	-20%	-10%	0%	10%	20%	30%
ρ	-0.2578838	-0.1584435	-0.06342719	0	0.1365728	0.2415565	-0.2578838
γ	0.0114692	0.009285574	0.004400278	0	0.2044003	0.4092856	0.6114692
ν	5.16919	-1.241054	0.007564877	0	0.2075649	-0.8410543	5.76919
τ	-0.2078706	0.006207353	0.2075649	0	0.2063613	2.233445	0.3921294
N	7.971515×10^{-6}	5.314485×10^{-6}	0.007564877	0	0.153671	0.4000053	0.600008

Table 6: **Model selection criteria for fitted models for IBM data set**

Model	**AIC**	**BIC**	**PRR**
Goel Okumoto	9546.476	9548.565	335.0073
Delayed Shaped	9546.476	9548.565	109.8387
Infection S-shaped	9548.476	9551.61	109.5235
Yamada imperfect debugging	75.97479	79.10836	199.6663
Testing coverage	71.23634	76.45895	4714.083
Dependent paramter model	105.0307	107.1197	98.73821
Ohba	9548.476	9551.61	47758.21
Vtub-shaped fault detection rate	11640	11644.18	168.4894
Cheng et al	12070.82	12075	3518.143
Chen SRGM	71.2169	76.43951	84.2606

Table 7: **Sensitivity analysis of the model paramters of the IBM data set**

Parameter	−30%	−20%	−10%	0%	10%	20%	30%
ρ	0.006366132	0.002814267	0.0006997361	0	0.2006997	0.4028143	0.6063661
γ	0.002500126	0.00166693	0.0008339586	0	0.0008339586	0.4016669	0.6025001
ν	0.6985742	4.199024	−0.1091852	0	0.09081482	4.599024	1.298574
τ	0.2933698	0.1934233	0.09364173	0	0.2936417	0.5934233	0.8933698
N	-4.442251×10^{-5}	-6.145704×10^{-5}	0.0004685412	0	0.2004685	0.3999385	-4.442251×10^{-5}

4.2 Application of the SPRT

After conducting the sensitivity analysis, it showed that τ and γ were the most sensitive parameters for the TSD-Phase two and IBM data sets respectively. Thus, the focus of the SPRT are on these two parameters. Table 8 shows the specifications of the cases for the application of the SPRT. Hence, the paramters of interest are set as, $\tau_0 = \tau - \delta$, $\tau_1 = \tau + \delta$ and $\gamma_1 = \gamma + \delta$, $\gamma_0 = \gamma - \delta$. The values of α and β are kept low. This is due to the fact that these values increase the effectiveness of the SPRT (Erixon et al., 2003).

Table 8: **Case for applying SPRT**

Case	$\boldsymbol{\delta}$	$\boldsymbol{\alpha}$ (Type I Error)	$\boldsymbol{\beta}$ (Type II Error)
Case 1 (for parameter τ)	0.03	0.1	0.1
Case 2 (for parameter γ)	0.9	0.1	0.1

Table 10 presents the SPRT for the parameter τ for the TSD-Phase two data set. From 10 cumulative failure for serial number 15 indicates we should terminate the experiment and conclude that the associated software the data was extracted from is not reliable. Likewise from 9 which presents the SPRT for parameter τ for the IBM data set. The cumulative failure for serial number 16 indicates the experiment should be terminated and the conclusion drawn that the software is not reliable.

Table 9: **SPRT results for the IBM data (Case 1, parameter τ)**

T	$N(t)$	**Acceptance region**	**Rejection region**	**Decision**
1	2	-36.77727	36.79657	continue
2	3	-36.65325	36.78118	continue
3	4	-36.623666	36.77414	continue
4	5	-36.61433	36.77027	continue
5	7	-36.61109	36.77174	continue
6	9	-36.60999	36.77396	continue
7	11	-36.60962	36.77666	continue
8	12	-36.60949	36.77626	continue
9	19	-36.60944	36.79635	continue
10	22	-36.60943	36.80281	continue
11	23	-36.60942	36.80283	continue
12	25	-36.60942	36.80626	continue
13	27	-36.60942	36.80977	continue
14	31	-36.60942	36.81955	continue
15	32	-36.60942	36.82004	continue
16	38	-36.60942	36.83579	continue
17	39	-36.60942	36.83632	continue
18	42	-36.60942	36.84289	continue
19	43	-36.60942	36.84355	continue
20	46	-36.609423	36.85011	continue
21	47	-36.60942	36.85087	continue

Table 10: **SPRT results for the TSD-Phase two (Case 1, parameter τ)**

T	$N(t)$	**Acceptance region**	**Rejection region**	**Decision**
1	3	-36.77244	36.80139	continue
2	4	−36.65195	36.7855	continue
3	4	−36.62366	36.77414	continue
4	7	−36.61402	36.77797	continue
5	9	−36.61098	36.77916	continue
6	9	−36.60999	36.77396	continue
7	10	−36.60962	36.77316	continue
8	13	−36.60949	36.77969	continue
9	17	−36.60944	36.78964	continue
10	23	−36.60943	36.8061	continue
11	25	−36.60942	36.80931	continue
12	30	−36.60942	36.8222	continue
13	32	−36.60942	36.82549	continue
14	36	−36.60942	36.83506	continue
15	37	−36.60942	36.83536	continue
16	39	−36.60942	36.83882	continue
17	39	−36.60942	36.83632	continue
18	39	−36.60942	36.83399	continue
19	39	−36.60942	36.83181	continue
20	42	−36.60942	36.83847	continue
21	43	−36.60942	36.83933	continue

Tables 11 and 12 presents the SPRT for the case where the focused parameter is $\gamma$for the IBM and TSD-phase two data set respectively. Based on the reliability, using the Chen SRGM, it can be concluded that the software is reliable as the SPRT gave an early decision.

Table 11: **SPRT results for the IBM data (Case 2, parameter γ)**

T	$N(t)$	**Acceptance region**	**Rejection region**	**Decision**
1	2	4415.771	–	Accept

Table 12: **SPRT results for the TSD-Phase Two data (Case 2, parameter γ)**

T	$N(t)$	**Acceptance region**	**Rejection region**	**Decision**
1	3	4297.679	–	Accept

5 Conclusion

Quality control techniques come in different forms. Software reliability growth model as a quality control technique is one of the most crucial elements in software production. A newly developed Chen SRGM is proposed incorporating environmental uncertainties. We developed the limits of its SPRT and showed the application of these limits. This was done by examining which parameters were the most sensitive ones. The results revealed that the variation of the parameters τ and γ were the largest. Thus the SPRT was performed based on these parameters. The first case of the SPRT showed the software was not reliable and hence the experiment was terminated, whiles the second case showed that software was reliable and could be deployed to the market. The results from the model selection criteria showed that the Chen SRGM is suitable for reliability analysis than the other competing models. The Chen SRGM can therefore assist software engineers model reliability data which follow the Chen distribution. The limits of the SPRT developed can be used as a quality control tool in software production during the monitoring phase.

List of Abbreviations

- **NHPP** Non-Homogeneous Poisson Process

- **SRGM** Software Reliability Growth Model
- **G-O** Goel Okumoto
- **SPC** Statistical Process Control
- **ANFIS** Adaptive Neuro Fuzzy Inference System
- **PHM** Proportional Hazard Model
- **PRR** Predictive Ratio Risk
- **SPRT** Sequential Probability Ratio Test

BIBLIOGRAPHY

Anderson, T. W. (1960). A modification of the sequential probability ratio test to reduce the sample size. *The Annals of Mathematical Statistics*, pages 165–197.

Basili, V. R. and Weiss, D. M. (1984). A methodology for collecting valid software engineering data. *IEEE Transactions on software engineering*, (6):728–738.

Erixon, P., Svennblad, B., Britton, T., and Oxelman, B. (2003). Reliability of Bayesian posterior probabilities and bootstrap frequencies in phylogenetics. *Systematic biology*, 52(5):665–673.

Feigenbaum, A. V. (1991). Total quality control. *New York*.

Jamwal, R. S., Jamwal, D., and Padha, D. (2009). Comparative analysis of different software quality models. In *3rd National Conference*.

Levitt, T. (1981). Marketing intangible products and product intangibles. *Cornell Hotel and Restaurant Administration Quarterly*, 22(2):37–44.

Michael, K. and Miller, K. W. (2013). Big data: New opportunities and new challenges [guest editors' introduction]. *Computer*, 46(6):22–24.

Ohba, M. (1984). Software reliability analysis models. *IBM Journal of research and Development*, 28(4):428–443.

Pham, H. (2003). Software reliability and cost models: Perspectives, comparison, and practice. *European Journal of Operational Research*, 149(3):475–489.

Pham, H. (2014). A new software reliability model with vtub-shaped fault-detection rate and the uncertainty of operating environments. *Optimization*, 63(10):1481–1490.

Rodrigues, D. (1985). Computers and basic writers. *College Composition and Communication*, 36(3):336–339.

Schonberger, R. J. (2008). *World class manufacturing*. Simon and Schuster.

Sharma, K., Garg, R., Nagpal, C., and Garg, R. K. (2010). Selection of optimal software reliability growth models using a distance based approach. *IEEE Transactions on Reliability*, 59(2):266–276.

Stalk, G. (1988). Time–the next source of competitive advantage.

Wald, A. (1945). Sequential tests of statistical hypotheses. *The annals of mathematical statistics*, 16(2):117–186.

Zhang, X. and Pham, H. (2002). Predicting operational software availability and its applications to telecommunication systems. *international journal of systems science*, 33(11):923–930.

Table 13: **IBM data**

Failure rate (weeks)	**Failures**	**Cumulative failures**
1	2	2
2	1	3
3	1	4
4	1	5
5	2	7
6	2	9
7	2	11
8	1	12
9	7	19
10	3	22
11	1	23
12	2	25
13	2	27
14	4	31
15	1	32
16	6	38
17	1	39
18	3	42
19	1	43
20	3	46
21	1	47

Table 14: **TSD-Phase two data**

Failure rate (weeks)	**Failures**	**Cumulative failures**
1	3	3
2	1	4
3	0	4
4	3	7
5	2	9
6	0	9
7	1	10
8	3	13
9	4	17
10	2	19
11	4	23
12	2	25
13	5	30
14	2	32
15	4	36
16	1	37
17	2	39
18	0	39
19	0	39
20	3	42
21	1	43