C. Wohlin, P. Runeson and J-E. Johansson, "When Is the Software Reliability Estimate Reliable?", Proceedings Bellcore/KPN/Purdue Workshop on Issues in Software Reliability", Leidschendam, The Netherlands, 1995.

When Is the Software Reliability Estimate Reliable?

Claes Wohlin[1] Per Runeson[2] Jan-Eric Johansson[3]

1) Dept. of Communication Systems[*], Lund Institute of Technology,
Lund University, Box 118, S-221 00 Lund, Sweden,
E-mail: claesw@tts.lth.se

2) Q-Labs AB, Ideon Research Park, S-223 70 Lund, Sweden,
E-mail: pr@q-labs.se

3) Telia AB, S-205 21 Malmö, Sweden,
E-mail: Jan-Eric.Johansson@materials.telia.se

Abstract

It is of paramount importance that software reliability requirements not only can be formulated, but also enforced and evaluated before accepting a software product for release. This requires sound acceptance criteria for software, which implies that methods for evaluating the software reliability are needed during the testing phase. Thus, theoretical models which are practically applicable are needed. These models should support software procurers in their difficult task of objectively accepting software products prior to releasing them for operation. This paper addresses this issue through studying two possible models for when to stop testing and accepting the software. The models are discussed with one particular software reliability growth model in mind, but the approach can be enlarged to take other software reliability growth models into account. The proposed stopping rules are illustrated on a set of failure data. It is concluded that it is indeed possible to evaluate and hence accept software products based on a software reliability estimate, including confidence in the estimate.

1. Introduction

A major problem in software reliability engineering is the problem of when to accept a software product based on some estimation of the current reliability level. Software reliability growth models are applied to estimate the current reliability level and to predict the forthcoming reliability growth. These models must, however, be combined with acceptance criteria of the software. It is not enough that the growth models

[*] This work is supported by National Board for Industrial and Technical Development (NUTEK), Sweden, reference Dnr: 93-2850.

estimate that the reliability exceeds a certain required value, we must know with a certain confidence that the software fulfils whatever reliability requirement set on it. Thus the following question must be answered: When is the software reliability estimate reliable? This is often referred to as the stopping rule problem in software reliability engineering, as it is concerned with the problem of when to stop testing and release the software.

The stopping rule problem has been discussed in several articles, see for example [Krten80, Wohlin90]. The problem is here studied further based on a particular software reliability model, although a similar approach can be taken for other models, and with the objective to provide a practical useful acceptance rule. This implies that the emphasis is on a method practical applicable, which however should have a theoretical basis. Two different stopping rules are described and then exemplified on a data set.

The software reliability model used throughout this paper is the model presented in [Currit86]. This model is the model normally referenced in the Cleanroom literature, therefore henceforth referred to as the Cleanroom software reliability growth model. The model is briefly introduced in Section 2, before introducing the two stopping rules in Section 3. The two rules are illustrated on real failure data in Section 4 and then finally some conclusions are presented in Section 5.

2. The Cleanroom software reliability growth model

The software reliability model discussed in connection with Cleanroom is as follows:

$$MTBF_k = A * B^k \text{, with } k = 0 .... \quad (1)$$

This form is supposed to describe the change in MTBF (Mean Time Between Failures) when faults are corrected. The model is discussed in detail in [Currit86]. The parameters A and B are estimated from the collected failure data by taking the logarithm of the equation and then applying the minimum square method. This gives estimates of A and B and in particular it is possible from the equation to predict future failure occurrences. From the value of MTBF, it is possible to calculate the reliability of the software. Thus meaning that based on a required reliability, it is possible to evaluate the software against this requirement. The problem is, of course, even easier if the requirement is formulated in terms of MTBF.

It can be worth noting that A can be seen as the estimated value of $MTBF_0$ based on the failures observed. B is a measure of the growth in MTBF between two consecutive failures. This can be interpreted as follows, if B is 1.1 we have a 10% growth in the MTBF between two consecutive failures.

The model is in particular more appealing than some other proposed models, for example, the models of Jelinski-Moranda [Jelinski72] and Goel-Okumoto [Goel79]. The reason being that these model use the maximum likelihood estimation method. This clearly cause problems if the number of data points is small. The objective is to develop high quality software, which indicates that the model to be used should be able to handle software with few failures. The faults made should have been detected before the testing phase, or even better never made in the first place. This is the actual meaning of

certification as it is discussed within Cleanroom, i.e. the high quality should be proven and testing should be aimed at certification, i.e. the objective is not fault detection. It should, however, be noted that if we succeed in developing software, which is almost fault free prior to the testing phase, which is the objective with Cleanroom, then software reliability growth models are not suited at all.

The model proposed in Cleanroom is simple and easily understood. The calculation of the parameters is also simple, although from a theoretical point of view a number of correction factors should be used due to that we take the logarithm of Eq. 1 before determining the model parameters. The correction factors are discussed in [Currit86]. They are, however, not used in this study as it is believed that from a practical viewpoint there are more critical problems, for example, data collection and criteria for acceptance. A major question remaining is the ability of the model to predict future failure behaviour as well as the reliability in the reliability estimate. These aspects are discussed further below.

2. The stopping rule problem

2.1 Introduction

It is not enough that the model estimates that the reliability exceeds the required reliability one time, there has to be a better criterion for when to stop the certification. The criterion ought to be based on statistics, i.e. how certain is the estimate of the reliability. The main reason for needing a stopping criterion is that the model only estimates the mean time between failures and it does only take the stochastic variations into consideration in the estimation process. Even if the model may predict an MTBF which is larger than the requirement, it is not certain that the reliability constraint is fulfilled. The model may predict a high value due to some high values in the beginning, which are a result of the stochastic variations of failure occurrences. It may also be the case that although the prediction is correct, i.e. the right mean value is predicted, the large stochastic variations may mean that several failure occurrences will be worse than the required reliability. That is, the failures will occur more frequently than required. This is quite natural and it has to be accepted. The procurer may, however, want to know how the stochastic variations influence the estimated MTBF and to know with a certain confidence that the estimated MTBF fulfils the reliability requirement. This is not explicitly discussed in the presentation of the Cleanroom software reliability growth model.

The Cleanroom software reliability growth model estimates an MTBF value, but does not say anything about the variations. Two different types of probabilities that are of interest can be identified:

- the probability that the estimated MTBF is higher than a predefined value. This probability covers the stochastic variations.
- the degree of confidence that the estimated MTBF is within certain limits. This probability covers the statistical uncertainty in the estimate itself.

The first probability gives the probability for a specific value, while the second is a measure of the uncertainty around the value. There is, however, a coupling between these two, which is further discussed after studying them separately. The two different approaches mentioned above are investigated: the stochastic variations and the statistical confidence in the estimate. First the stochastic variations are taken into consideration (Section 2.2-2.5) and then the certainty in the estimate itself is discussed (Section 2.6). In Section 2.7 the coupling between the two approaches is described. Section 2.8 describes a stepwise method of how to apply the two different stopping rules.

2.2 Distribution of the stochastic variations

The user of the model would like to know how probable it is that the MTBF actually fulfils the requirement, i.e. that the MTBF is stable above the required value. This knowledge can only be obtained if a distribution is assumed. The failures occur randomly based on the usage profile. It is therefore natural to assume that the actual times for failures are distributed according to an exponential distribution, with the mean value as predicted by the Cleanroom model. Mean value according to Cleanroom:

$$MTBF_k = A * B^k \qquad (2)$$

A required mean time between failures is the basis for stopping the testing, i.e. to accept the software as fulfilling the reliability requirement. This requirement is denoted $MTBF_C$.

The probability density function of the assumed exponential distribution can be written as:

$$f_k(x) = \mu_k \ e^{-\mu_k x} \text{ , with } \mu_k = \frac{1}{A*B^k} \quad (3)$$

The probability distribution is simply:

$$F_k(x) = 1 - e^{-\mu_k x} \quad (4)$$

2.3 Stochastic basis

The objective is to find a way to determine if the software can be accepted, not only based on the estimation of a mean value but with a certain confidence, i.e. in other words with a calculated risk. The aim is to formulate a method that estimates the risk of accepting a product not fulfilling the requirements. The method should be based on both the risk that the mean value by chance is high and the actual trend of the reliability estimate.

Based on the formulas in the previous section, it is possible to formulate a stochastic basis for acceptance of software products. The confidence that the actual MTBF is above the requirement can be estimated through the estimated MTBF and a fictitious new outcome. This is determined so that the estimate after the fictitious value equals the requirement. Based on the fictitious value that fulfils this relationship, it is possible from the exponential distribution to determine the probability that the next estimated MTBF equals the requirement. If the estimate is high above the requirement the fictitious value can be quite low, which results in a low probability (i.e. low risk) that the next estimation of the MTBF will be as low as the requirement. This reasoning can be applied when the estimate is above the requirement. The proposed way of evaluating the confidence is better explained through some equations and figures.

Let us assume that the number of failures that have occurred is k and that $MTBF_{k+1}$ is higher than the required MTBF. $MTBF_{k+1}$ is the next expected time to a failure and it is estimated as

$$MTBF_{k+1} = A_k * B_k^{k+1} \quad (5)$$

The next step is to determine the fictitious outcome, i.e. outcome k+1 (denoted Outcome1(k+1)), that makes the next estimated MTBF equal to the requirement,

$$MTBF_{k+2} = A_{k+1} * B_{k+1}^{k+2} = MTBF_C \quad (6)$$

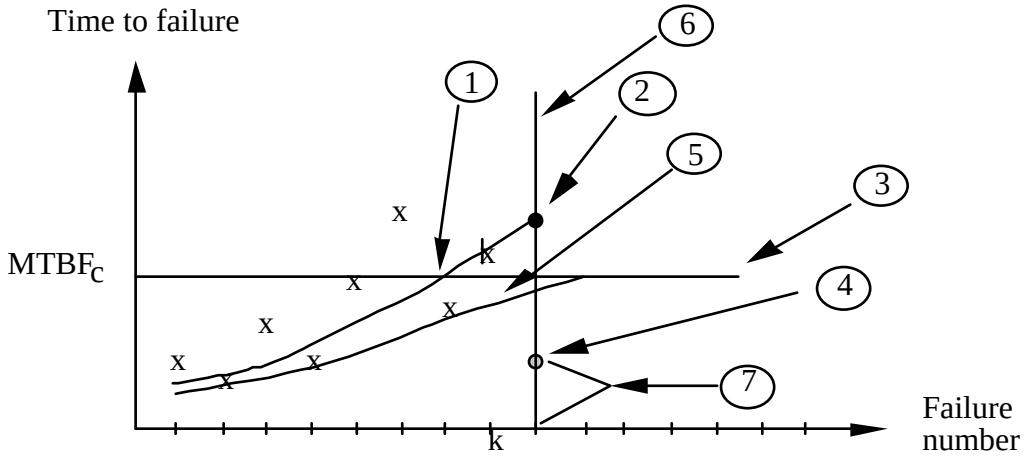
The probability that $MTBF_{k+2}$ equals or is less than $MTBF_C$ can be calculated based on $MTBF_{k+1}$ and the fictitious outcome k+1 as

$$P1 = P(MTBF_{k+2} \leq MTBF_c) = 1 - e^{-Outcome1(k+1) / MTBF_{k+1}} \quad (7)$$

Another possible application of this is to determine a second fictitious outcome k+1 (denoted Outcome2(k+1)) that makes $MTBF_{k+2}$ less or equal than $MTBF_{k+1}$. This can be used to determine the trend, since $MTBF_{k+2}$ ought to be larger than $MTBF_{k+1}$, i.e. if B is larger than 1. This is true if we have a reliability growth, which should be the case. This probability is calculated as

$$P2 = P(MTBF_{k+2} \leq MTBF_{k+1}) = 1 - e^{-Outcome2(k+1) / MTBF_{k+1}} \quad (8)$$

The equation including $MTBF_c$ is illustrated in Figure 1. The other equation can be illustrated in a similar way.



1. The estimated curve based on k outcomes
2. Estimation of $MTBF_{k+1}$
3. The level of the required MTBF, i.e. $MTBF_c$
4. Fictitious outcome k+1, which makes the new estimated curve at the point k+2 to be equal to the requirement, see also 5.
5. The new curve is estimated based on the k actual outcomes and the fictitious outcome k+1. The curve is estimated to reach the requirement after k+2 outcomes.
6. The possible outcome is assumed to be distributed according to an exponential distribution, i.e. between zero and infinity.
7. The probability space that the estimated curve actually will equal or be below the requirement after k+2 outcomes.

The calculated probability is a measure of certainty that the MTBF will be above the requirement in the future as well.

Figure 1: An explanation of the proposed probability calculations.

This illustrates the way we obtain the probabilities that should be used as a stochastic basis to determine whether we are prepared to accept the estimated reliability or not. We

will return to these two probabilities below, but first we have to deal with the problem of determining the fictitious outcome.

2.4 Determining a fictitious outcome

The basic problem is that the MTBF is a function of the parameters A and B, which are estimated based on the failure data. The problem encountered is to determine a particular outcome that, when the parameters are estimated, gives the wanted MTBF. It is not possible to invert the function, but by introducing a search algorithm it is possible to achieve the objective.

The fictitious outcome is determined with the following algorithm, which also is illustrated in Figure 2:

1. Determine which of the two probabilities to find the fictitious outcome for, i.e. Outcome1(k+1) or Outcome2(k+1). The rest of the algorithm is described assuming that we are interested in Outcome1(k+1). The algorithm is exactly the same for Outcome2(k+1).
2. Two variables High and Low are introduced as a help to locate the right fictitious value. High is initially given a value that is higher than the estimated MTBF, i.e. $MTBF_{k+1}$. Low is set to 0 (zero).
3. Outcome1(k+1) is first set to $MTBF_{k+1}$. This should result in that $MTBF_{k+2}$ is higher than $MTBF_C$, since B should be higher than 1.
4. The new curve is calculated, i.e. $MTBF_{k+2} = A_{k+1} * B_{k+1}^{k+2}$.
5. If $MTBF_{k+2} > MTBF_C$ then
   High = Outcome1(k+1)
   else
   Low = Outcome1(k+1);
6. The fictitious outcome, Outcome1(k+1) is given a new value, i.e. Outcome1(k+1) = (High-Low)/2 + Low.
7. If ($MTBF_{k+2} = MTBF_C$) or (Outcome1(k+1) = High or Low) then
   The search is ready and the right Outcome1(k+1) has been found.
   else
   Goto 4.

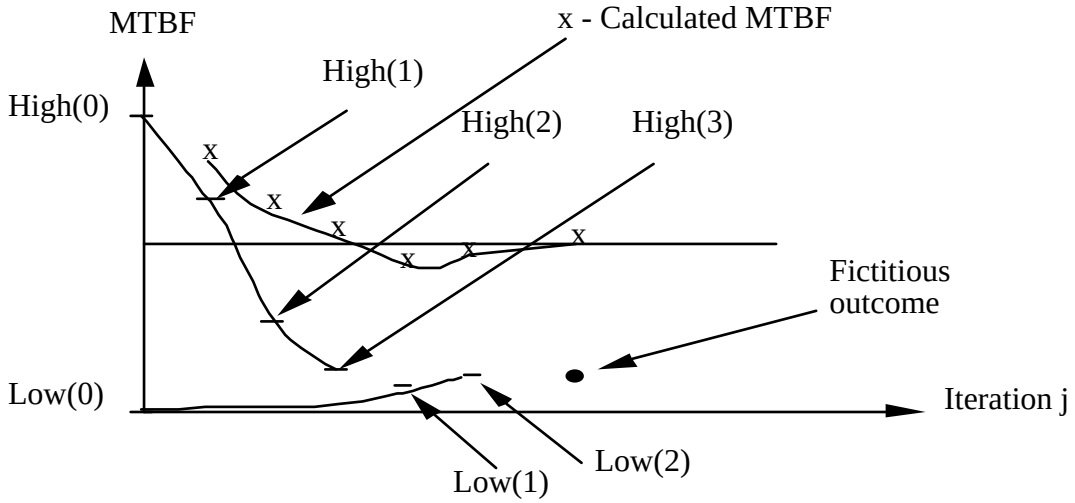


Figure 2: Algorithm for determination of a particular MTBF.

In Figure 2, it can be seen how the algorithm changes High respectively Low depending on the estimated $MTBF_{k+2}$ so that it finally reaches the requirement, which can be either $MTBF_c$ or $MTBF_{k+1}$ depending on the probability to be calculated.

The proposed way of calculating the fictitious outcome, as well as the determination of the probabilities will become clearer in Section 3 where an example is presented.

2.5 Evaluation criteria

The interpretation of the probabilities has to be discussed a little further. The first probability (based on Outcome1(k+1)) is a measure of the confidence. The second one is a little more difficult to interpret, but it is possible to see it as a probability for the trend. A low value of the second probability means that there is a little probability that $MTBF_{k+2}$ is lower than $MTBF_{k+1}$. This means that there is only a small risk that the curve will have a downward tendency in comparison. Suppose that there should only be a risk of 5% that the $MTBF_{k+2}$ is less than $MTBF_c$ and that we would like it to be twice as probable that the curve has an upward tendency as a downward. Theis means that the value of the second probability should be 0.333. Other values of this probability is calculated from: 1 / (1 + No. of times the upward tendency should be larger than the downward tendency), i.e. in our small example 1 / (1 + 2) = 0.333.

One problem remains, however, if the risk (see 5% above) is much smaller than the acceptance level, then the tendency requirement can be loosened. The reason is, of course, that if our estimated MTBF is high above the requirement, then the tendency is not particular interesting. But for values close to the requirement, it is reasonable to assume that we want be pretty sure that the tendency is upwards.

Proposal:

- Determine the two probability requirements, P1(req.) and P2(req.), see Eqs. 7 and 8 for explanation of P1 and P2.
- Evaluate P1 separately, i.e. is P1 < P1(req.). (Example: Check if P1 < 0.05).

- Evaluate the product of P1 and P2. The first evaluation has to succeed before this second evaluation is performed. The product should be less than the product of the required values, i.e. P1 * P2 < P1(req.) * P2(req.). (Example: P1 is smaller than 0.05. Check if P1 * P2 < 0.0167).

The proposed stochastic stopping rule based on the reliability model suggested in Cleanroom will be applied on an example in Section 3.2 to explain the practical use of the proposed scheme.

2.6 The confidence in the estimated MTBF

In the previous sections, it has been shown that it is possible to determining a probability that the estimated MTBF is higher than a predefined value. The statistical confidence in the estimate has, however, not been treated. This problem will be addressed in this section.

The parameters, in the Cleanroom software reliability model, are estimated after taking the logarithm of Eq. 1, and then these are put into the unlinear equation and through this an approximation of the curve is obtained. The confidence of the estimate is obtained in a similar way, i.e. from the linear curve and then transformed to the unlinear case. The confidence interval for the estimated MTBF can be calculated as follows.

For simplicity let x = ln(MTBF).

First some help variables are introduced to simplify the rest of the calculations, where n is the number of data points. Let:

$$S_{xx} = \sum_{i=0}^{n-1} (x_i)^2 - \frac{1}{n} * \left(\sum_{i=0}^{n-1} x_i \right)^2 \quad (10)$$

$$S_{kk} = \sum_{i=0}^{n-1} (k_i)^2 - \frac{1}{n} * \left(\sum_{i=0}^{n-1} k_i \right)^2 \quad (11)$$

$$S_{kx} = \sum_{i=0}^{n-1} (x_i * k_i) - \frac{1}{n} * \left(\sum_{i=0}^{n-1} k_i \right) * \left(\sum_{i=0}^{n-1} x_i \right) \quad (12)$$

From these we obtain

$$Q_0 = S_{xx} - \frac{(S_{kx})^2}{S_{kk}} \quad (13)$$

The estimated standard deviation of the one value becomes

$$s = \sqrt{Q_0 / (n-2)} \quad (14)$$

Letting k_e be the value of k at which the confidence interval should be calculated, we obtain the mean error in the estimate as

$$d = s * \sqrt{\frac{1}{n} + \frac{(k_e - k_{mean})^2}{\sum_{i=0}^{n-1} (k_i - k_{mean})^2}} \quad (15)$$

where

$$k_{mean} = \left(\sum_{i=0}^{n-1} k_i \right) / n \quad (16)$$

The confidence interval at k_e becomes

$$I_x = (x_{mean} - t_{a/2}(f) * d, x_{mean} + t_{a/2}(f) * d) \quad (17)$$

with

f = n-1 (i.e. the number of degrees of freedom)

and

$$x_{mean} = \left(\sum_{i=0}^{n-1} x_i \right) / n \quad (19)$$

This interval is a two-sided confidence interval with 1-a as confidence degree. The t-value is found in tables of the t-distribution, which is a function of f and the confidence degree.

These calculations should now be formulated as a stopping rule. This is done by determining the degree of confidence wanted. It should be noted that a 95% confidence means that a 90% confidence degree should be used in the calculations. The reason is that only the lower limit is of interest, i.e. 5% is below the lower limit. The confidence interval is calculated from the estimated MTBF, the estimated standard deviation of the estimate and a value from the t-distribution. The value depends on the confidence degree wanted and the number of observations.

A simple example: Assume that the confidence interval for ln($MTBF_k$) is (7.3303, 7.9195) with a mean value of 7.6249. The confidence interval for $MTBF_k$ is obtained by applying the exponential function, i.e. (exp(7.3303), exp(7.9195)) with a mean value exp(7.6249), i.e. (1526, 2750) with mean value 2049. It can be observed that the symmetric confidence interval in the linear case becomes asymmetric when applying the exponential function. The stochastic variations give the outcomes that is the input to the estimation of the curve. The uncertainty in the estimate is shown by the confidence interval, see Figure 3. The asymmetry in the confidence interval indicates that further

theoretical work is needed, although the proposed scheme may be a suitable starting point from a practical viewpoint.

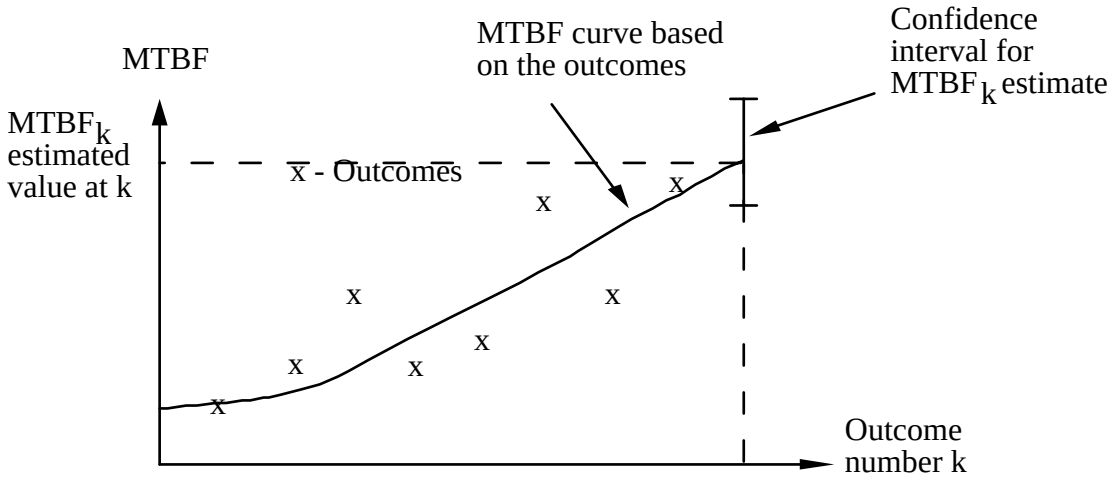


Figure 3: Outcomes, estimated curve and confidence interval

The stopping rule is simply that when the lower limit of the defined confidence interval is over the requirement, then the software is accepted. This means that with the confidence degree chosen the estimate of the MTBF is over the required MTBF. The stopping rule is based on the fact that the MTBF ought to be over the requirement if the whole confidence interval is over it.

The application of the statistical stopping rule is shown in an example in Section 3.2.

The trend is missing for this rule. In the future it ought to be possible to extend the rule to capture this aspect as well. The confidence interval can be determined, which also ought to mean that it is possible to determine the probability that the value is above a specific value assuming the t-distribution (normal distribution with unknown standard deviation).

2.7 Relationship between the two stopping rules

The stochastic stopping rule implies that the estimate of $MTBF_{k+2}$ is over the requirement at k+2 with the probability level chosen as a suitable acceptance confidence. The estimate is based on the real outcomes and the fictitious outcome at k+1, where the value of the fictitious outcome is determined from the confidence required. This means that it is possible to choose a probability that says that for example in 95% of the cases the $MTBF_{k+2}$ is above a specific value, i.e. for example the requirement.

The statistical stopping rule on the other hand gives a confidence interval in which the estimated MTBF is with the specified confidence degree. This means that if the assumptions of the models were perfect, the models should be close to each other at the lower limit of the confidence interval. The stochastic rule should give the point which equals the lower limit of the confidence interval, see Figure 4.

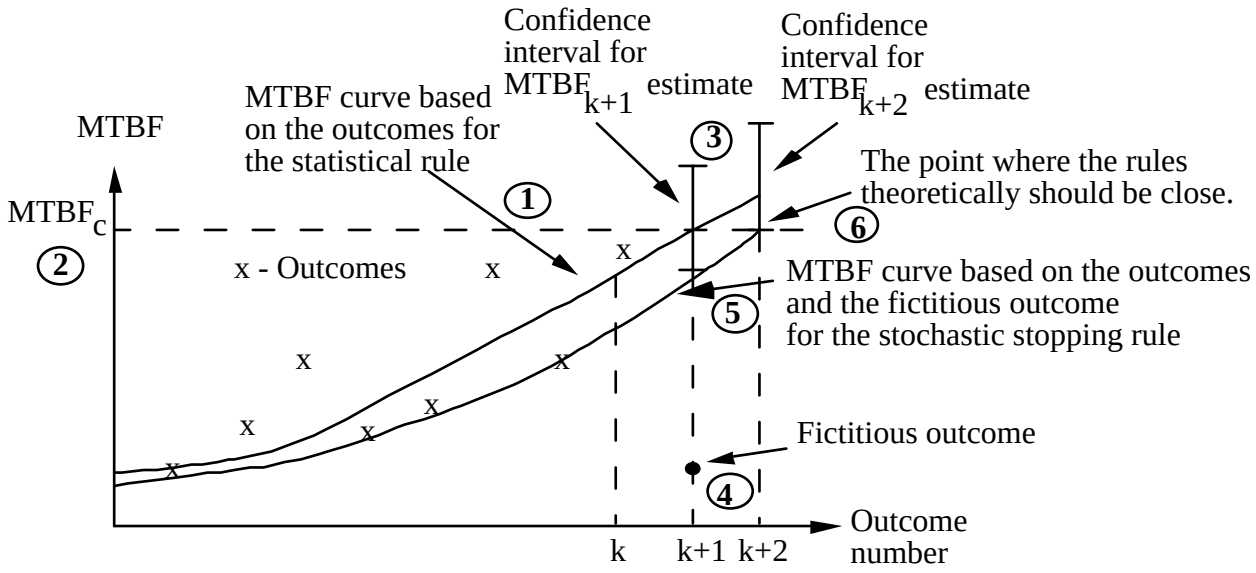


Figure 4: Relationship between the stopping rules.

The numbering in the figure can briefly be described as:

1. The curve is calculated based on k outcomes and its predicted behaviour is shown for k+1 and k+2.
2. The requirement is shown with a horizontal dashed line.
3. The confidence intervals for the predicted values of $MTBF_{k+1}$ and $MTBF_{k+2}$ are shown in the figure. The lower limit of the confidence interval for $MTBF_{k+2}$ is in this case exactly on the requirement. This is not always the case, it might be a bit above the requirement since the interval is only calculated at discrete points in time, i.e. k, k+1 and k+2.
4. A fictitious outcome is determined for the stochastic stopping rule.
5. Based on the k outcomes and the fictitious one, a new line is calculated. This line reaches the requirement at k+2.
6. This means that the line calculated from the stochastic rule and the lower limit of the confidence interval for the statistical rule is close to each other at the requirement.

It must be noted that the line calculated for the stochastic stopping rule always will meet the requirement exactly, while the interval calculated from the statistical stopping rule will be equal or larger than the requirement. The reason for this being that the calculations are performed at discrete points in time.

The two stopping rules should theoretically be close to each other at the point indicated in Figure 4. This is, however, not the case practically, since the assumptions are not completely fulfilled. The stochastic rule assumes that the time between failures is exponentially distributed, while the statistical rule is based on the normal distribution around the line. Another aspect making the results slightly different, when applying the rules practically, is the fictitious outcome. This outcome actually means that the calculations for the stochastic rule is made with one more outcome. In particular, it is made with the knowledge of a bad outcome, i.e. a short time between failures. The picture is further blurred because of the unlinear relationship describing the MTBF.

The main advantage with the stochastic rule is the opportunity to study both the requirement as well as the trend. The disadvantage is mainly its complexity in both understanding and computation effort. The statistical stopping rule is simpler theoretically, which includes understanding. This is clearly an advantage. The major disadvantage is its inability to capture the trend. This disadvantage may be possible to remove, which has to be further examined in the future.

Although the models should give similar results, it is recommended to use both models until more experience is gained.

2.8 Method for applying the stopping rule

The techniques described above in Sections 2.2-2.6 can be formulated in a method for applying the two stopping rules. The following method is proposed, where A is for the stochastic stopping rule and B for the statistic stopping rule:

1. Determine the reliability requirement, preferably in terms of an MTBF requirement, or so that the requirement can be recalculated to an MTBF requirement.
2A. Determine the probability requirements, i.e. the risk and the trend probabilities, see section 2.5.
2B. Determine the confidence degree wanted in the estimate, see section 2.6.
3. Estimate the curve as proposed in the Cleanroom literature, i.e. make a linear regression and then transform it to the unlinear case, see Section 2.6.
4A. Determine the fictitious outcome that makes the next MTBF exactly equal to the requirement, see Sections 2.2 and 2.3.
4B. Based on the results from the previous item a confidence interval is calculated for the unlinear case, see Section 2.6. An evaluation is then carried out.
5A. Calculate the risk probability, P1, based on the fictitious one, see Section 2.2.
6A. Evaluate the obtained value of P1 compared to the risk probability determined in item 1, see Section 2.5.
7A. Determine a new fictitious outcome that makes the next MTBF equal to the previous prediction, see Section 2.2 and 2.3.
8A. Calculate the trend probability, P2, based on the second fictitious outcome, see Section 2.2.
9A. Evaluate the obtained value of P2 compared to the required product of the risk probability and the trend probability, see Section 2.5.

Every item of the proposed method should, of course, not be carried out at all times. For example, if the evaluation of P1 says that the software should not be accepted then the rest of the items have not to be carried out.

3. Example

In Appendix A, a table with failure data is presented. The data are used to illustrate and, to some extent, evaluate the ideas presented above.

3.1 Stochastic stopping rule

The requirements are as follows:

$MTBF_C = 1500$
P1(req.) = 0.05
P2(req.) = 1/3 ≈ 0.333

It should be observed that k in the table is calculated from 0 and upwards, since the first MTBF normally is denoted $MTBF_0$. The actual numbering of the failures is of minor interest. The stopping rule should be applied after 4, 5 and 19 failures, since the predicted MTBF is larger or equal than 1500. These three cases have been investigated, but they are not shown here. The software was not accepted after 4, 5 or 19 failures have occurred. The next time the estimated MTBF is larger than 1500 is for k = 20, i.e. 21 failures have occurred.

In particular, it must be noted that if the mean value was the only criterion, then the software should have been accepted after 4 failures (k = 3). It is from the rest of the failure data obvious that this would have been too early, and that the high estimate is simply obtained based on a number of high values in the beginning. This indicates the importance of confidence in the estimate.

First calculation:

k = 20, see table in Appendix A. The calculation gives:

$MTBF_{21} = A_{20} * B_{20}^{21} = 189.4097 * 1.1133^{21} = 1804$ (see table).

Let High = 10000, Low = 0 and Outcome1(21) = 1804 => $MTBF_{22} = 2009$
This means that $MTBF_{22} > MTBF_C$ => High = 1804 and Outcome1(21) = (High-Low) / 2 + Low = (1804 - 0) / 2 + 0 = 902 => $MTBF_{22} = 1771$.
=> High = 902, Outcome1(21) = 451 => $MTBF_{22} = 1563$
=> High = 451, Outcome1(21) = 226 => $MTBF_{22} = 1376$
=> Low = 226, Outcome1(21) = 339 => $MTBF_{22} = 1484$
=> Low = 339, Outcome1(21) = 395 => $MTBF_{22} = 1526$
=> High = 395, Outcome1(21) = 367 => $MTBF_{22} = 1504$
=> High = 367, Outcome1(21) = 353 => $MTBF_{22} = 1494$
=> Low = 353, Outcome1(21) = 360 => $MTBF_{22} = 1501$
=> High = 360, Outcome1(21) = 357 => $MTBF_{22} = 1496$
=> Low = 357, Outcome1(21) = 359 => $MTBF_{22} = 1498$
=> Low = 359, Outcome1(21) = 360 => $MTBF_{22} = 1501$

The value of Outcome1(21) is now equal to High and has to be accepted as the best fictitious value even if $MTBF_{22}$ is not exactly equal to $MTBF_C$. For accepting rules see above in the presented algorithm. This calculation procedure is easily implemented into a program. Based on the fictitious outcome the probability P1 can be calculated,

$P1 = 1 - e^{- Outcome1(21) / MTBF_{21}} = 1 - e^{- 360 / 1804} = 0.181$

The value of P1 is clearly above the requirement, i.e. the software is not accepted. This means that it is unnecessary to calculate P2 and consequently Outcome2(21).

Second calculation:

The next real outcome is 3798. This gives $MTBF_{22}$ = 2301. The search for the fictitious outcome results in that Outcome1(22) = 101. The calculation of P1 gives

$$P1 = 1 - e^{-\text{Outcome1}(22) / MTBF_{22}} = 1 - e^{-101 / 2301} = 0.043$$

which is better than the required maximum risk, P1(req.) = 0.05. This means that the value of Outcome2(22) has to be determined so that $MTBF_{23}$ after the next outcome is equal to $MTBF_{22}$. This results in Outcome2(22) = 1187, which gives

$$P2 = 1 - e^{-\text{Outcome2}(22) / MTBF_{22}} = 1 - e^{-1187 / 2301} = 0.403$$

The product of P1 and P2

$$P1 * P2 = 0.043 * 0.403 = 0.0173$$

but the requirement is

$$P1(\text{req.}) * P2(\text{req.}) = 0.05 * 0.333 = 0.0167$$

The software can not be accepted due to the second evaluation criterion. The upward trend is not strong enough compared to P1, which means that the software is not accepted.

Third calculation:

The next outcome is 2775 from this follows that $MTBF_{23}$ is equal to 2671. The next fictitious outcome is Outcome1(23) = 41. This lead to that

$$P1 = 1 - e^{-\text{Outcome1}(23) / MTBF_{23}} = 1 - e^{-41 / 2671} = 0.015$$

which means that the first evaluation criterion is fulfilled, i.e. P1 < P1(req.). The value of Outcome2(23) is then determined, Outcome2(23) = 1316. P2 can now be calculated

$$P2 = 1 - e^{-\text{Outcome2}(23) / MTBF_{23}} = 1 - e^{-1316 / 2671} = 0.389$$

The product between P1 and P2 is now determined

$$P1 * P2 = 0.015 * 0.389 = 0.0058$$

which is clearly below the required value 0.0167.

The software is accepted based on the Cleanroom software reliability growth model and the stochastic stopping rule introduced here after 23 failures (k = 22 in the table).

3.2 Statistical stopping rule

The statistical stopping rule can be applied when the estimate of the MTBF fulfils the requirement. The question is if this also means that the confidence interval for the estimate fulfils the requirement. The calculation of the confidence interval is carried out with well known statistics, see Section 2.6. The calculation of the confidence interval is carried out when the next estimated time between failures fulfils the requirement.

The first time the requirement is fulfilled is for k = 3 (4 failures have occurred). The value of $MTBF_4$ (from curve) = 2291, with 90% confidence interval (2291 - 1750, 2291 + 7409) = (541, 9700). Since only the lower limit is of interest the interval says that with a confidence degree of 95% is the estimate over 541. This means that the software is not accepted.

The calculation is continued until the confidence interval is over the requirement.

k = 4: $MTBF_5$ = 1587 with interval (325,7737).
k = 20: $MTBF_{21}$ = 1804 with interval (961, 3389).
k = 21: $MTBF_{22}$ = 2301 with interval (1250, 4234).
k = 22: $MTBF_{23}$ = 2671 with interval (1492, 4777).
k = 23: $MTBF_{24}$ = 2953 with interval (1698, 5128).

Finally, the interval is over the requirement on 1500, which means that the software is accepted after 24 failures based on the statistical stopping rule. The software was close to being accepted after 23 failures, but the confidence interval was just below the requirement, hence is could not be accepted.

4. Conclusions

Two ways of determining when to stop testing have been examined. The stochastic stopping rule is supposed to capture the stochastic variations in the failure process. The statistical stopping rule is formulated to capture the uncertainty in the estimated MTBF. The two rules give the following points in time when the testing should be stopped, for the example presented:

Stochastic stopping rule: 23 failures
Statistical stopping rule: 24 failures

The proposed rules show that it is possible to formulate test stopping criteria, and hence to accept the software based on a reliability requirement. The stochastic stopping rule shows that it is possible to formulate a rule that captures both the risk and the trend. The statistical stopping rule is an example of how a rule can be formulated based on the statistical uncertainty in the estimate. These rules are not supposed to be perfect, but it is believed that they can work as reasonable tools for when to stop testing and consequently accepting the quality level of the software.

Currently, it is recommended to apply both rules until more experience is gained.The results from the example indicate, however, that the choice is probably not critical since the rules accept the software very close in terms of the number of failures, at least in this particular example. It should, however, be observed that even if there is a small difference in the number of failures, the actual time until the software is accepted can

vary. The reason, of course, being that the time between failures ought to be quite long near the acceptance. The difference in time may hence be quite large although there is only one failure outcome that differs between the two rules. This is one of the reasons that it is necessary to apply both rules until more experience has been gained. It is, of course, important not to test the software unnecessarily long but on the other hand it should not be accepted until the requirement is fulfilled.

One conclusion is that the actual choice between which of the rules to use in practice has to be further investigated both theoretically and by applying the proposed rules to failure data. It is necessary to study advantages, disadvantages, similarities, differences and realism of the proposed stopping rules. In particular, the opportunity of capturing the trend with the statistical rule as well must be investigated thoroughly.

The perhaps most important conclusion is that it is possible to apply the software reliability growth model proposed in Cleanroom and in particular it is possible to determine when the test phase should be considered to be completed based on the requirements and the model, including the stopping rule.

References

[Currit86] Currit, P. Allen, Dyer, Michael and Mills, Harlan D., ”Certifying the Reliability of Software”, IEEE Transactions on Software Engineering, Vol. SE-12, No. 1, 1986, pp. 3-11.

[Jelinski72] Jeliniski, Z., and Moranda, P., ”Software Reliability Research”, Statistical Computer Performance Evaluation, 1972, pp. 465-484.

[Goel79] Goel, Amrit L., and Okumoto, Kazuhira, ”Time-dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures”, IEEE Transactions on Reliability, Vol. R-28, No. 3, 1979, pp. 206-211.

[Krten80] Krten, O. Joe and Levy, Dave, ”Software Modelling for Optimal Field Entry”, Proc. Annual Reliability and Maintainability Symposium, pp. 410-414, 1980.

[Wohlin90] Wohlin, Claes and Körner, Ulf, ”Software Faults: Spreading, Detection and Costs”, Software Engineering Journal, Vol. 5, No. 1, pp. 33-42, 1990.

Appendix A: Failure data

The failure data is presented in [Currit86]. In our context it is used to evaluate the ideas presented regarding the software reliability models and in particular to illustrate the results presented.

k	Outcome(k)	B(k)	A(k)	MTBF(k+1)
0	85	-	-	-
1	85	1	85	85
2	479	2,3739	63,7186	852,4
3	965	2,4639	62,1568	2290,8
4	469	1,7942	85,3594	1587,1
5	385	1,4655	111,7926	1107,5
6	796	1,4146	118,5845	1344,2
7	277	1,2401	154,2923	863
8	927	1,246	152,587	1104,5
9	340	1,1685	181,1194	859,5
10	405	1,1292	200,682	763,8
11	150	1,0607	247,2463	501,5
12	277	1,0401	265,6252	442,8
13	503	1,0439	261,7971	477,7
14	694	1,0537	251,422	551
15	620	1,0564	248,3864	597,6
16	4050	1,0968	205,9106	990,5
17	3064	1,1188	185,2693	1397,5
18	522	1,1015	202,3233	1269,8
19	1797	1,107	196,3941	1500
20	2329	1,1133	189,4097	1804,1
21	3798	1,1232	178,5941	2301
22	2775	1,1255	176,066	2670,7
23	2393	1,1243	177,4855	2953,5
24	909	1,1121	192,9082	2747,6
25	994	1,1025	206,7968	2614,5
26	28212	1,1235	176,6877	4099
27	14956	1,1343	162,631	5541
28	1971	1,1262	173,4047	5443,9
29	5308	1,126	173,6788	6108
30				