STUK-YTO-TR 119
JANUARY 1997

Statistical reliability assessment of software-based systems

J. Korhonen
VTT Electronics
U. Pulkkinen, P. Haapanen
VTT Automation

In the Finnish Centre for Radiation and Nuclear Safety
the study was supervised by
Harri Heimbürger

This study was conducted by order of
the Finnish Centre for Radiation and Nuclear Safety

The conclusions presented in the report are those of the authors and do not represent the official position of the Finnish Centre for Radiation and Nuclear Safety.

FINNISH CENTRE FOR RADIATION AND NUCLEAR SAFETY
P.O.BOX 14, FIN-00881 HELSINKI, FINLAND
Tel. +358-9-759881
Fax +358-9-75988382

ISBN 951-712-164-4
ISSN 0785-9325

Oy Edita Ab
Helsinki 1997

KORHONEN, Jukka (VTT Electronics), PULKKINEN, Urho, HAAPANEN, Pentti (VTT Automation). Statistical reliability assessment of software-based systems. STUK-YTO-TR 119. Helsinki 1997. 31 pp.

ISBN 951-712-164-4
ISSN 0785-9325

Keywords: Safety, safety analysis, reliability analysis, automation, programmable systems, reactor protection systems, nuclear reactor safety, dynamic testing, statistical analysis of test data

ABSTRACT

Plant vendors nowadays propose software-based systems even for the most critical safety functions. The reliability estimation of safety critical software-based systems is difficult since the conventional modeling techniques do not necessarily apply to the analysis of these systems, and the quantification seems to be impossible. Due to lack of operational experience and due to the nature of software faults, the conventional reliability estimation methods can not be applied.

New methods are therefore needed for the safety assessment of software-based systems. In the research project “Programmable automation systems in nuclear power plants (OHA)”, financed together by the Finnish Centre for Radiation and Nuclear Safety, the Ministry of Trade and Industry and the Technical Research Centre of Finland, various safety assessment methods and tools for software based systems are developed and evaluated.

This volume in the OHA-report series deals with the statistical reliability assessment of software based systems on the basis of dynamic test results and qualitative evidence from the system design process. Other reports to be published later on in OHA-report series will handle the diversity requirements in safety critical software-based systems, generation of test data from operational profiles and handling of programmable automation in plant PSA-studies.

KORHONEN, Jukka (VTT Elektroniikka), PULKKINEN, Urho, HAAPANEN, Pentti (VTT Automaatio). Ohjelmistopohjaisten järjestelmien luotettavuuden tilastollinen arviointi. STUK-YTO-TR 119. Helsinki 1997. 31 s.

ISBN 951-712-164-4
ISSN 0785-9325

Avainsanat: Luotettavuus, ohjelmoitavat digitaalijärjestelmät, reaktoriturvallisuus, dynaaminen testaus, luotettavuusanalyysit

TIIVISTELMÄ

Ydinvoimalaitosten toimittajat tarjoavat nykyään ohjelmoitavaa tekniikkaa kaikkein turvallisuuskriittisimpiinkin turvatoimintoihin. Turvallisuuskriittisten ohjelmoitavien järjestelmien luotettavuuden arviointi on vaikeaa koska tavanomaiset mallinnusmenetelmät eivät välttämättä sovellu näiden analyysiin, ja luotettavuuden kvantifiontia pidetään mahdottomana. Käyttökokemusten puute ja ohjelmistovikojen luonne aiheuttavat sen, että tavanomaisia luotettavuuden arviointimenetelmiä ei voida soveltaa.

Uusia menetelmiä tarvitaan näinollen ohjelmoitavien järjestelmien turvallisuuden arviointiin. ”Ydinvoimalaitosten ohjelmoitavat automaatiojärjestelmät (OHA)” -tutkimushankkeessa kehitetään ja arvioidaan erilaisia ohjelmoitavien järjestelmien turvallisuuden arviointimenetelmiä. Hanketta rahoittavat Säteilyturvakeskus (STUK), Kauppa- ja teollisuusministeriö (KTM) sekä Valtion teknillinen tutkimuskeskus (VTT).

OHA-projektin raporttisarjan tämä osa käsittelee ohjelmoitavien järjestelmien luotettavuuden tilastollista arviointia järjestelmän dynaamisten testien tulosten ja sen suunnitteluprosessin laatua kuvaavan kvalitatiivisen evidenssin avulla. Myöhemmin julkaistavissa sarjan muissa raporteissa käsitellään turvallisuuskriittisten ohjelmoitavien järjestelmien diversiteettivaatimuksia, testidatan generointia käyttöprofiileista sekä ohjelmoitavien järjestelmien käsittelyä laitoksen PSA-tutkimuksissa.

CONTENTS

1 INTRODUCTION

Plant specific PSAs are required by the safety authorities in many countries, and quantitative safety goals are set on the core melt frequency and possibly also on the reliability of the most critical safety functions, e.g. the reactor scram. For example the FinnishYVL-requirements state that the "*failure probability of the reactor scram shall be less than 10^{-5} per demand with a good confidence*".

Plant vendors nowadays propose software-based systems even for the most critical safety functions. The reliability modeling of safety critical software-based systems is difficult since the conventional modeling techniques do not necessarily apply to the analysis of these systems, and the quantification seems to be impossible. Due to lack of operational experience and due to the nature of software faults, the conventional reliability estimation methods can not be applied.

The reliability of software-based systems is a property of the operation environment as well as that of the system itself. Although there may be errors in the software, these errors can cause a loss of safety function only when inputs occurring with very low probability are introduced to the system. In other words, the reliability of programmable systems depends on the operational profile, which as the probability distribution of input sequences, varies from one environment to another. This restricts the use of generic operational experience in determination of reliability parameters.

Quantitative reliability estimates are always based on certain evidence, which is most often operational experience statistics. For software-based systems this evidence is either very limited or not applicable due to differences between the operational profiles of the data source and the actual system. Another source of evidence is obtained from the dynamic testing of the system. If high reliability is required, the number of tests is very large, and it may be practically impossible to test the system extensively enough. Thus additional evidence from other sources is needed to make the reliability estimation practicable.

To obtain better estimates for reliability of programmable systems, all possible evidence should be applied in the analysis. This requires extensive applications of experts opinions about the weight of various pieces of evidence. A most suitable approach for using analyzing experts judgements is based on Bayesian models. Part of the factors may be directly observable and measurable, part of the may be unobservable and qualitative characterization of the system and its development process.

The development process of the system follows certain quality assurance and quality control principles, which are based on applicable standards, but which may vary from one developer to another. More strict principles are believed to result to more reliable products. Thus the quality assurance process provides evidence on reliability. Same may apply to other tools and principles followed during the development process. To use this kind of evidence in determining the reliability estimates requires models and ability to weight the evidence.

In this study, which is a part of the research project "Programmable automation systems in nuclear power plants (OHA)", various approaches to quantify the reliability of software-based systems using dynamic test results and other qualitative evidence are discussed. The OHA-project is financed together by the Finnish Centre for Radiation and Nuclear Safety, the Ministry of Trade and Industry and the Technical Research Centre of Finland.

1.1 Dynamic testing as a process

The technical or mathematical meaning of random testing refers to an explicit lack of "system" in the choice of test data. Statistical testing, as a special case of random testing, uses random sampling of what is considered the usage distribution of system inputs. This distribution describes the eventual usage of the system in its intented environment.

Statistical testing involves various tasks and activities, which are outlined in Fig. 1. The most essential activities of the testing process include generation of test cases, testing itself and calculation of system reliability on the bases of test results. This report will mostly deal with the selection or development of the method to assess reliability. The most promising alternatives of different assessment approaches will be discussed. Such issues that may influence on the selection of the assessment method (like required reliability and confidence level) will also be studied in the report.

Once the assessment method is known and test results are available, application of the method (see Fig. 1) is in principle a straightforward and easy task. Statistical testing has at this phase of the process been performed and the test results are available. The rest of the process is to feed the results into the method, write the estimated reliability into the test report and draw the inevitable conclusions regarding the required reliability and estimated reliability. In practice, however, if errors have been detected, things are not usually this simple.

The first problem that is encountered in the case of software failure, is the interpretation of the terms *fault* and *failure*. As the system is in dynamic testing tested against its specification, all the anomalies that in general can be found, are between the system and its specification. Lets assume that a deviation has been detected. When things have been sorted out, the cause of the error is traced to the system specification that in fact specifies the desired behavior in an imperfect manner. So actually there is no fault in the software itself, but in its specification. Should this kind of a fault be counted into those faults that are used in reliability assessment? Probably yes, as the procedure has revealed an inconsist-

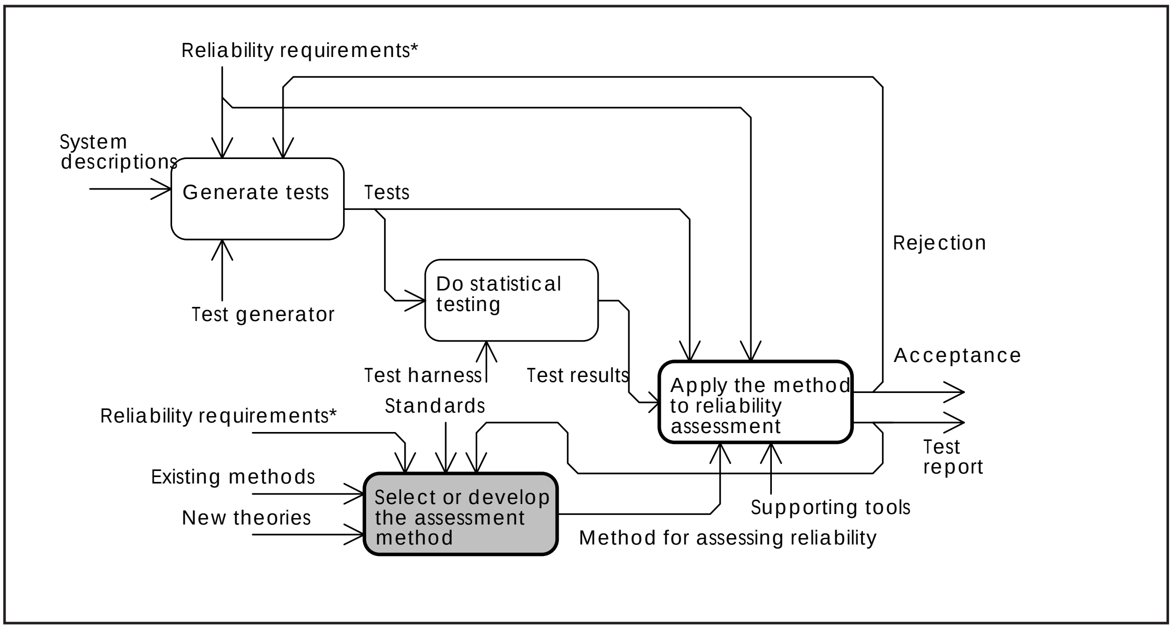

Figure 1. *A general model of the dynamic testing process.*

ency in the system specification, which raises suspicions towards the development process. Similar examples, which certainly need further analysis, are for instance:

What is the cause of the error? Is it likely that similar errors exist elsewhere in the system?

Did the error prevent the system from executing its main functions (i.e. how much the error did affect the functionality of the system)?

Is the nature of the error deterministic (i.e. if the test case is repeated, does the error manifest itself in the same manner)?

These kind of issues usually lead to intervene human judgment into the assessment. Though the mixture of human judgment and quantitative assessment is not always a desirable combination, it may be the only option in this case. Certainly an important goal of the assessment is to minimize the possible sources of subjectivity, which can be achieved for instance by providing such information that will support quantitative reliability judgment. This is a noteworthy point also in selecting the assessment method, i.e. to examine the data that the assessment method requires and the possibilities to provide such data in an objective manner.

The "frame" which includes all the elements used in the evaluation of test results, has thus an obvious tendency to expand much wider than the original, purely quantitative results suggest. It is important to be aware of this, and study and define the frame before the startup of testing (see for instance [IEEE94] for error classification).

Once the necessary definitions of terms have been agreed and the target values for the reliability and confidence level been set, testing can started. Thus prior to testing it is possible to estimate the number of test cases that are needed if the system operates according to its specification. But if errors are found during testing, how can we calculate the new amount of test cases? Certainly the number of tests can not be the same as originally estimated. What kind of methods can be used for this? Are we able to utilize those positive experiences that we got about the system before it failed or should we make a fresh start? These and other interesting reliability assessment themes are discussed in this report.

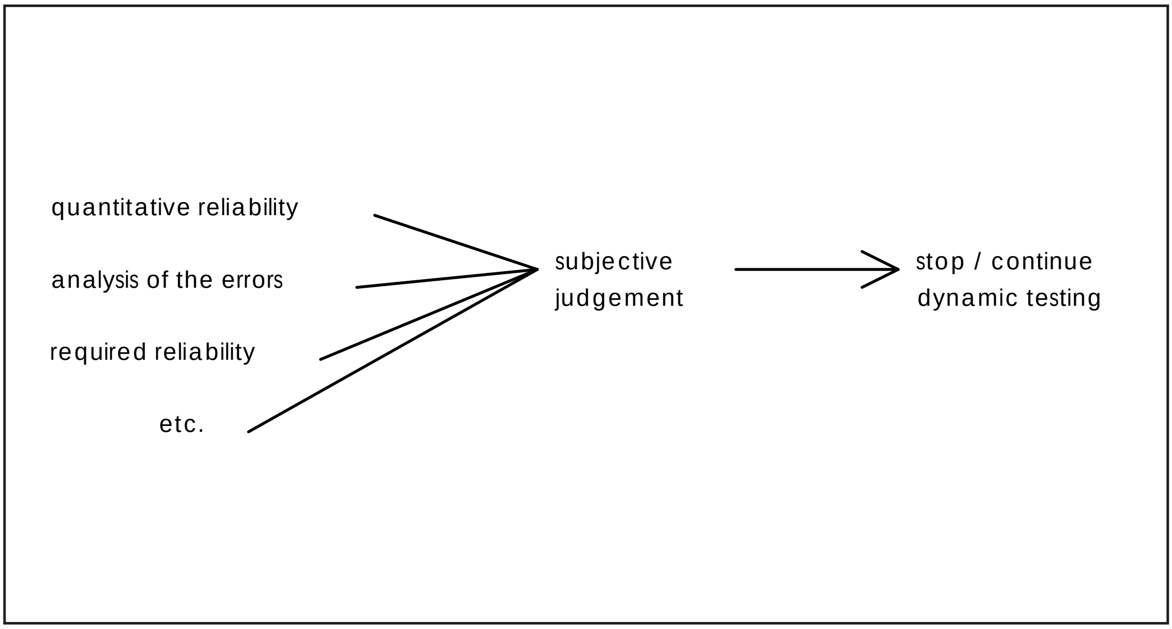

Figure 2. *Human judgement in assessing the results of dynamic testing.*

2 THEORETICAL BACKGROUND

2.1 Definitions of software reliability

The reliability of a software product can be defined in several ways. Generally, software reliability means the probability of failure-free operation of a computer program in a specified environment for a specified time [see Musa et al, 1987]. The IEEE standards define software reliability more specifically [IEEE 199x]. One measure for software reliability given by the standard [IEEE 199x] is the **probability of successful run**, which is defined as the sum of the probabilities of successful runs divided by the sum of probabilities of all runs. Statistically it may be estimated for example from the results of a test by dividing the number of successful runs by the number of all runs. The **run reliability**, R_k, is defined as the probability that randomly selected *k* runs give correct results.

Software reliability may be defined also in terms of time. Measures as mean time to failure, **MTTF,** or mean time between failures, **MTBF**, can be used. These measures can be defined in continuous or discrete time. Closely related to these measures are various failure rates, the definition of which corresponds to those of hardware reliability [see Musa et al, 1987].

In the case of emergency systems, it is important that the system operates correctly when it is demanded. The systems must generate the emergency function when a certain input occurs. For example, the reactor protection signal should occur in certain situations. The failures, in which the system does not produce the emergency function are most important for the safety of the plant; these failures are the ones which are modelled in PSAs. The reliability in this case is measured by the **probability of failure on demand**. The safe failures, in which the emergency function occurs although they are not demanded, are not as important. However, the existence of safe failures may be seen as evidence on the existence of failures in general.

In addition to the above measures, the software reliability may be characterized by the number of errors in the program. In this case one may be interested to determine the **probability distribution of the number of errors**. The reliability is then measured by the **probability that there are less the *n* failures**. Usually this type of reliability measures are applied when the software reliability growth during the testing and development process is considered.

Many of the above mentioned software reliability measures are not purely describing the target system but they depend essentially on its operational profile. This mean leads to certain requirements for the software testing, e.g. the test cases should correspond to the actual user profile.

2.2 Some concepts of statistics

As already discussed software reliability is measured in terms of probability. The reliability measures are thus statistical concepts, and their interpretation must be considered carefully. The operationalization and the practical use of the quantitative reliability characteristics depend on the interpretation of statistical concepts. In the interpretation of statistical concepts one must be aware of the fact that probability is very abstract notion formulated as an measure theoretic model for uncertainty.

```
660 -2750 (F) 9.999756 0.274612 51 0 0 A1`6
711 -2750 (ir) 9.999756 0.274612 56 0 1 A1`6
767 -2750 (st,) 9.999756 0.274612 85 0 2 A1`6
852 -2750 ( one m) 9.999756 0.274612 271 2 5 A1`6
1124 -2750 (ust mak) 9.999756 0.274612 300 1 6 A1`6
1423 -2750 (e a dif) 9.999756 0.274612 250 2 6 A1`6
1671 -2750 (f) 9.999756 0.274612 30 0 0 A1`6
1702 -2750 (er) 9.999756 0.274612 71 0 1 A1`6
1774 -2750 (ence betw) 9.999756 0.274612 382 1 8 A1`6
2156 -2750 (een the) 9.999756 0.274612 274 1 6 A1`6
660 -2640 (inter) 11.49971 0.643097 170 0 4 A1`6
831 -2640 (pr) 11.49971 0.643097 77 0 1 A1`6
909 -2640 (eta) 11.49971 0.643097 108 0 2 A1`6
1017 -2640 (tions of pr) 11.49971 0.643097 406 2 10 A1`6
1424 -2640 (oba) 11.49971 0.643097 134 0 2 A1`6
1559 -2640 (bility) 11.49971 0.643097 197 0 5 A1`6
1749 -2640 (. ) 11.49971 0.643097 47 1 1 A1`6
1802 -2640 (As pur) 11.49971 0.643097 262 1 5 A1`6
2065 -2640 (el) 11.49971 0.643097 67 0 1 A1`6
2132 -2640 (y ma) 11.49971 0.643097 194 1 3 A1`6
2327 -2640 (th-) 11.49971 0.643097 103 0 2 A1`6
660 -2530 (ematical formulation, probability is an additive) 6.166504 0.274811
1770 5 47 A1`6
660 -2420 (measur) 1.964233 0.274612 266 0 5 A1`6
927 -2420 (e def) 1.964233 0.274612 184 1 4 A1`6
1109 -2420 (ined in a measur) 1.964233 0.274612 614 3 15 A1`6
1723 -2420 (e space) 1.964233 0.274612 272 1 6 A1`6
1995 -2420 (. ) 1.964233 0.274612 46 1 1 A1`6
2041 -2420 (T) 1.964233 0.274612 56 0 0 A1`6
2097 -2420 (he meas-) 1.964233 0.274612 333 1 7 A1`6
660 -2310 (ur) 11.49971 0.38887 77 0 1 A1`6
737 -2310 (e theor) 11.49971 0.38887 266 1 6 A1`6
1003 -2310 (etic f) 11.49971 0.38887 199 1 5 A1`6
1203 -2310 (or) 11.49971 0.38887 77 0 1 A1`6
1280 -2310 (m) 11.49971 0.38887 71 0 0 A1`6
1352 -2310 (ula) 11.49971 0.38887 113 0 2 A1`6
1465 -2310 (tion doesn\222) 11.49971 0.38887 425 1 10 A1`6
1889 -2310 (t sa) 11.49971 0.38887 137 1 3 A1`6
2027 -2310 (y an) 11.49971 0.38887 168 1 3 A1`6
2194 -2310 (ything) 11.49971 0.38887 236 0 5 A1`6
660 -2200 (a) -5.714157 0.096146 41 0 0 A1`6
701 -2200 (bout the inter) -5.714157 0.096146 479 2 13 A1`6
1180 -2200 (pr) -5.714157 0.096146 77 0 1 A1`6
1256 -2200 (eta) -5.714157 0.096146 107 0 2 A1`6
1364 -2200 (tion of pr) -5.714157 0.096146 330 2 9 A1`6
1694 -2200 (oba) -5.714157 0.096146 132 0 2 A1`6
1827 -2200 (bility) -5.714157 0.096146 194 0 5 A1`6
2014 -2200 (,) -5.714157 0.096146 23 0 0 A1`6
2037 -2200 ( b) -5.714157 0.096146 63 1 1 A1`6
2098 -2200 (ut it f) -5.714157 0.096146 187 2 6 A1`6
2283 -2200 (ix) -5.714157 0.096146 72 0 1 A1`6
2353 -2200 (es) -5.714157 0.096146 77 0 1 A1`6
660 -2090 (the r) -5.694321 -0.416656 158 1 4 A1`6
817 -2090 (ules of pr) -5.694321 -0.416656 331 2 9 A1`6
1148 -2090 (oba) -5.694321 -0.416656 132 0 2 A1`6
1280 -2090 (bility calculus. ) -5.694321 -0.416656 550 2 16 A1`6
1822 -2090 (T) -5.694321 -0.416656 56 0 0 A1`6
1878 -2090 (he ) -5.694321 -0.416656 109 1 2 A1`6
```

the confidence intervals are not applicable, and one must define confidence regions. Often these regions are defined as ellipsoids covering the true value of the parameter with fixed confidence (confidence ellipsoids).

Various "confidence" concepts may be defined also based on the Bayesian interpretation. To define these concept we must take a look at the Bayesian estimation. The Bayesian estimation follows the likelihood principle, and it uses the likelihood function defined earlier. The basic idea of the Bayesian inference is to express the uncertainty of all the unknown variables of the model by probability distributions. This means that the parameter, which is unknown a priori is modelled as a random variable.

The observable variables, or the evidence $\boldsymbol{E} = (x_1,...,x_n)$, are modelled by their joint distribution, i.e. the likelihood function. Before observations are made, the uncertainty about the value of the parameter (θ) is modelled by a probability distribution, **the prior distribution**, which we denote by $g(\theta)$. The evidence $\boldsymbol{E}$ provides additional information about θ, and the distribution of θ is updated by using the Bayes' rule. The updated distribution, **the posterior distribution**, is the conditional distribution of θ given the evidence, and we denote it by $g(\theta|\boldsymbol{E})$. The posterior distribution is obtained from

$$g(\theta|E) = \frac{l(x_1,\ldots,x_n|\theta)g(\theta)}{\int\limits_{\theta\in\Theta} l(x_1,\ldots,x_n|\theta)g(\theta)d\theta}. \tag{5}$$

The posterior distribution describes the uncertainty about the parameter when the information from the observed sample is taken into account. In the predictive Bayesian framework we are interested in predicting the next observation, x_{n+1}, on the basis of the evidence $\boldsymbol{E} = (x_1,...,x_n)$. The prediction is given in terms of **predictive distribution,** defined by

$$p(x_{n+1}|E) = \int\limits_{\theta\in\Theta} p(x_{n+1}|\theta)g(\theta|E)d\theta. \tag{6}$$

The predictive distribution expresses the uncertainty on the next observation, given the earlier observations. The uncertainty on the parameter is taken into account by integrating the conditional distribution of x_{n+1}, over the posterior distribution of θ, or using the rule of total probability.

In many cases we are interested to evaluate certain posterior probability intervals for θ. One possibility is to apply so called Highest Posterior Probability Density (HPD) regions [see Box & Tiao, 1973, or Tanner, 1991]. A region $\boldsymbol{R} \subset \boldsymbol{\Theta}$ is HPD region of content γ (i.e. at confidence level γ), if

a) $P(\theta \in R|E) = \gamma$ (7)

b) for $\theta_1 \in \mathbf{R},\ \theta_2 \notin \mathbf{R},\ p(\theta_1|E) \geq p(\theta_2|E)$. (8)

In other word, HPD region is a kind of central posterior confidence interval. If θ is one-dimensional, another possibility is to define upper and lower posterior uncertainty bounds, θ_U and θ_L simply as the posterior γ-fractiles

$$\theta_U : \int_{-\infty}^{\theta_U} g(\theta|E)d\theta = \gamma, \tag{9}$$

and

$$\theta_L : \int_{\theta_L}^{\infty} g(\theta|E)d\theta = \gamma. \tag{10}$$

The problems of Bayesian approaches are connected with the selection of the prior distribution. The specification of the prior is basically a subjective expression of the uncertainty about the parameter prior to any observations. If the number of observations is large the likelihood determines almost totally the posterior distribution, and "the subjectivity of the prior" vanishes. However, we should notice that also the likelihood model is based on subjective judgements. This is true also for frequentistic approach, which also requires specification of statistical model a priori.

3 STATISTICAL MODELLING APPROACHES

In the dynamic testing of software-based systems it is important to be able to draw statistical conclusions from test results. If, for instance, the target system has been tested with 1000 independent test cases, and no erroneous behaviour of the system has been detected, what can be said about the reliability of system or the probability of failure during next 1000 demands? An interesting question will also be confronted, if an error or several errors have been detected during dynamic testing. What kind of attitude should be taken towards system reliability and especially, towards the general fitness of the system for its purpose?

The quantitative software reliability estimation is necessary when the quantitative safety criteria are set on the reliability of emergency systems at nuclear power plants. The objectivity of reliability depends on the nature and amount of the evidence behind the estimates. It can be argued that the most objective evidence originates from systematic random test of the system. In the following, the strength of that evidence is evaluated on the basis of statistical analysis.

3.1 Success in dynamic testing

Parnas et.al. [1990] argue that the validation of system safety and trustworthiness should rest on a tripod made up of testing, mathematical review and certification of personnel and production process. So roughly one third of the assessment result should be based on quantitative assessment, the rest being more or less qualitative. Leveson [1986] proposes that the influence of quantitative methods should be even smaller in safety evaluation.

The approach for quantitative reliability assessment proposed by Parnas et.al. [1990] is simple. They state that in most safety-critical applications it is not necessary to know the actual probability of failure; it is enough if it can be shown that the failure probability is below a specified upper limit. Before the testing is started, the probability for the correctness of the result should be set. That is, we have to specify how sure we want to be that the result is correct. From the statistical point of view this means that a confidence level (see section 2.2) must be set for the assessment process.

The choice of the confidence level is not trivial. From frequentistic point of view the upper confidence bound for system failure probability (p_U) at confidence level γmeans that if a large number of test sets, each including several test cases, is performed, then *100γ%* of tests sets are such that the true failure probability is covered by the interval [$0, p_U$] and that in *100(1-γ)%* of test cases the true failure probability is larger than the upper confidence bound. Thus there is a *100(1-γ)%* chance to make an erroneous judgement on the failure probability. In some cases 90% confidence level is high enough, while in some other case we'll need to be 99,99% sure that the probability estimate is correct. Actually the choice of confidence level depends on the possible consequences of the erroneous judgement, which depend on the consequences of systems failure. Further, the confidence level should correspond somehow to the decision makers risk attitude. We discuss the choice of confidence level from decision theoretic point of view later in this report.

A notable feature of the procedure discussed by Parnas [1990] is the repetition of testing in case of a failure: if a failure is detected during testing, the fault is corrected and whole testing process is started again from the beginning. This is justified by the fact that if a software fault is detected and corrected, we cannot know the real consequences of the repair. The "new" reliability may be slightly better or it may even be much worse than before the repair. Thus the previous, successful test results should not be used in favour of the system after a failure.

Test cases in this method should be statistically independent and taken from a distribution that represents the actual usage of the target system. This may in many cases be a tough job that requires careful analysis of operational environment of the system. The methods for the analysis depend on the type of the system: if there is a user interface in the system, then Markov chains are perhaps the best means to describe the usage information [see Whittaker & Thomason, 1994, and Whittaker & Poore, 1993]. If the system is connected to other systems or sensors, the frequencies of different stimuli can provide some help for the analysis.

Time trajectories are only one example of a more general issue with programmable systems: the concern about the determinism of software. This is expressed by Parnas, for instance. Unfortunately time is not the only thing that creates nondeterminism into programmable systems. Concurrent task handling is the most common cause of nondeterminism, as concurrent event handling lead in many cases to nondeterminism in the implementation of software. Here nondeterminism is deliberate and implemented by means of an operating system to schedule concurrent processes.

Time trajectories and other causes of state behaviour should be considered before dynamic testing. A study of these issues may provide a great deal of help in determining the correct length for test cases. Though the state behaviour of a target system gets its final outlook during the implementation, the things that influence in it can already be seen in specifications. Detailed knowledge of system implementation is therefore not necessary, but an analysis of specifications is generally enough to determine the duration of test cases.

In the following analyses that can be done on the bases of a *succesful* test, we first assume that the usage model describes the actual usage of the target system. Later we (in section 3.3) shortly discuss the case in which the usage model is approximative.

We denote by $p = 1/h$ the probability that a test case which is randomly selected from the operational profile leads to a failure. Thus $1\text{-}p = 1\text{-}1/h$ represents the probability that the target system operates according to its specifications, and it may be interpreted as the reliability of the software. The series of independent test cases can be modelled as Bernoulli trials and thus the probability that N independent random tests are successful (given that the failure probability is p) is given by

$$P(\text{“}N \text{ tests succesful”}|\,\mathrm{p}) \;=\; (1\text{-}\mathrm{p})^{\mathrm{N}} = (1-\frac{1}{h})^{N}. \tag{11}$$

From (11) we may easily determine the classical upper confidence bound for p at any confidence level

$$p_U = 1-(1-\gamma)^{1/N}, \tag{12}$$

where γ is the confidence level. If the reliability requirement is that the failure probability is at most p_U at confidence level γ, then the number of successful test to show that is given by

$$N = \frac{\ln(1-\gamma)}{\ln(1-p_U)}. \tag{13}$$

Here we must emphasize the probabilistic interpretation of the upper confidence bound p_U. According to the definition discussed in section 2.2, p_U (at confidence level γ) is the largest value of the parameter p which leads to N successful tests at most with probability *(1-γ)*. We notice that $p = p_U$, then there is *(1*-γ) change to obtain N successful tests, i.e. the upper confidence bound gives the probability of observing a successful test result, although the failure probability is unacceptable. Further, we notice that the probability statement leading to the upper confidence bound doesn’t say anything about the parameter *(p)*, but it is about the test sample (=“N successful tests”).

Fig. 3 gives the numbers of successful tests required to demonstrate different failure probabilities at different confidence levels. Increase in the confidence level does not effect the number of test cases much. If the required confidence level is 99,9%, the number of test cases is 6 904. And if a still higher confidence is needed, 99,99% for instance, the number of test cases is 9 205. Thus the error marginal of the result can be squeezed very low with a reasonable increase in the number of test cases. However, in many life-critical systems the reliability requirements are higher - failure probabilities of 10^{-6} or 10^{-7} per demand or per mission are not unusual.

From the Bayesian point of view the above statistical inference gets another form. The likelihood describing the evidence (***E*** = "no failures in *N* tests") is given by

$$l(E|p) = (1-p)^N . \tag{14}$$

If the prior distribution is *g(p)* then the posterior distribution is

$$g(p|E) = \frac{(1-p)^N g(p)}{\int_0^1 (1-p)^N g(p)dp}, \tag{15}$$

which must be, in the general case, evaluated numerically. If we apply a conjugate beta-prior distribution, having density function

$$g(p|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1}, \tag{16}$$

in which $\alpha > 0$, $\beta > 0$ fixed parameters and $B(\cdot,\cdot)$ is the beta function we obtain the posterior

$$g(p|E,\alpha,\beta) = \frac{1}{B(\alpha,\beta+n)} p^{\alpha-1}(1-p)^{\beta+n-1}. \tag{17}$$

If $\alpha = 1$ and $\beta = 1$, i.e. the prior distribution is uniform, the posterior is

$$g(p|E,\alpha,\beta) = (n+1)(1-p)^n. \tag{18}$$

The upper posterior γ-fractile corresponding to (18) is given by

$$p_U = 1-(1-\gamma)^{1/N+1}, \tag{19}$$

and the number of successful test required to show that $p \le p_U$ with probability γ is

$$N = \frac{\ln(1-\gamma)}{\ln(1-p_U)} - 1. \tag{20}$$

Opposite to the classical confidence bounds, the Bayesian "confidence bound" is based on a probability statement on the value of the parameter. This is due to the Bayesian setting, in which the parameter is assumed to be a random variable. However, we notice that when uniform prior distribution is applied, the classical and Bayesian upper bound are close to each other. If other pri-

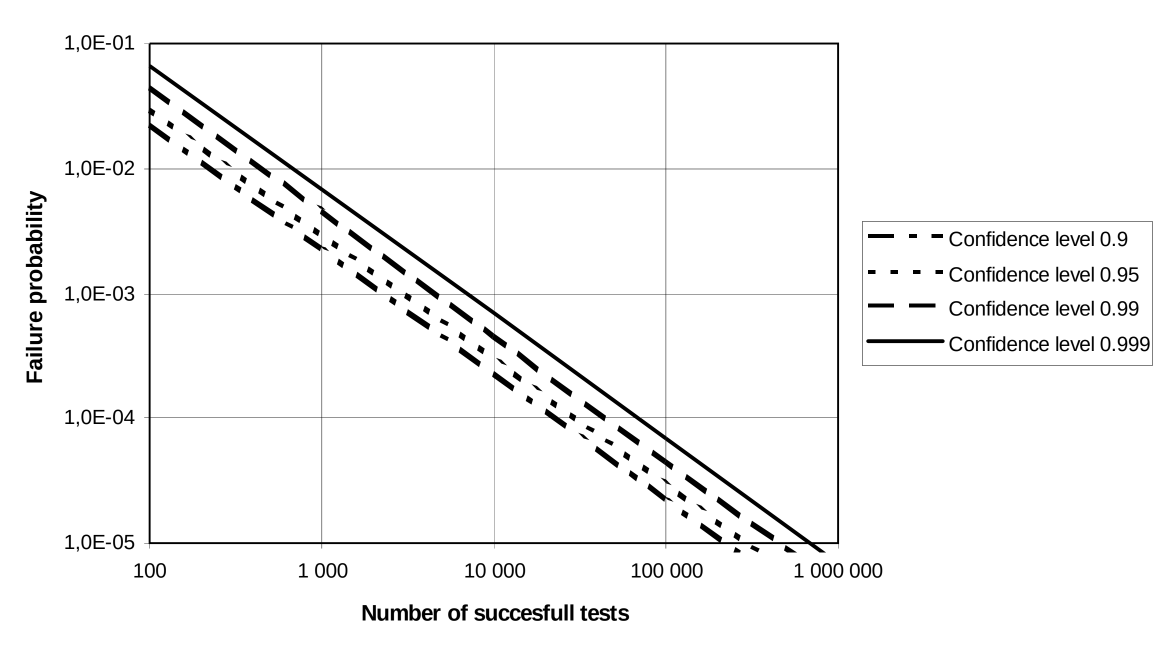

Figure 3. *Number of tests required to accept failure probabilities at selected confidence levels.*

ors are applied, this it not true [see e.g. Martz & Waller, 1982, Box & Tiao, 1973, Kapur & Lamberson, 1977].

The selection of the prior distribution is one of the most important issues of Bayesian statistical inference. The prior distribution should express *a priori* knowledge on the parameter. If little is known about the parameter, then the prior should be flat. However, if the decision maker wants to give the greatest weight to the sample, the so called noninformative priors should be applied (see Box & Tiao, 1973, for details). In the case of binomial of Bernoulli sampling, which is the setting here, the approximately noninformative and data translated prior distribution is the beta distribution with parameters α = ½ and β = ½. The Bayesian upper bound for the parameter based on the noninformative prior is close to the one given in equation (19).

If there is prior evidence (in the form of earlier operational experiences or tests, expert judgements, etc.) which is powerful enough to give reasons to apply informative priors, then the number of tests can be smaller. When noninformative priors are applied their use must be justified and the arguments behind the prior must be documented clearly. To demonstrate the impact of prior distributions we present Fig. 4, where the Bayesian upper bounds are given for certain beta prior distributions as the function of the number of successful tests. We notice that if very high reliability is required, the prior distribution doesn't have much effect on the required number of successful tests, unless $\alpha > 1$ (α is the parameter of the prior distribution, see equation (16)). If α is small, but β is large, then smaller number of tests are needed to justify that p is smaller that some fixed limit with a high probability.

3.2 Errors found during testing

The assessment of software reliability during dynamic testing is more difficult when failures occur during test. Littlewood & Wright, [1995], argue that the earlier failure information needs to be taken into account when new tests are planned for a system that has once failed and then

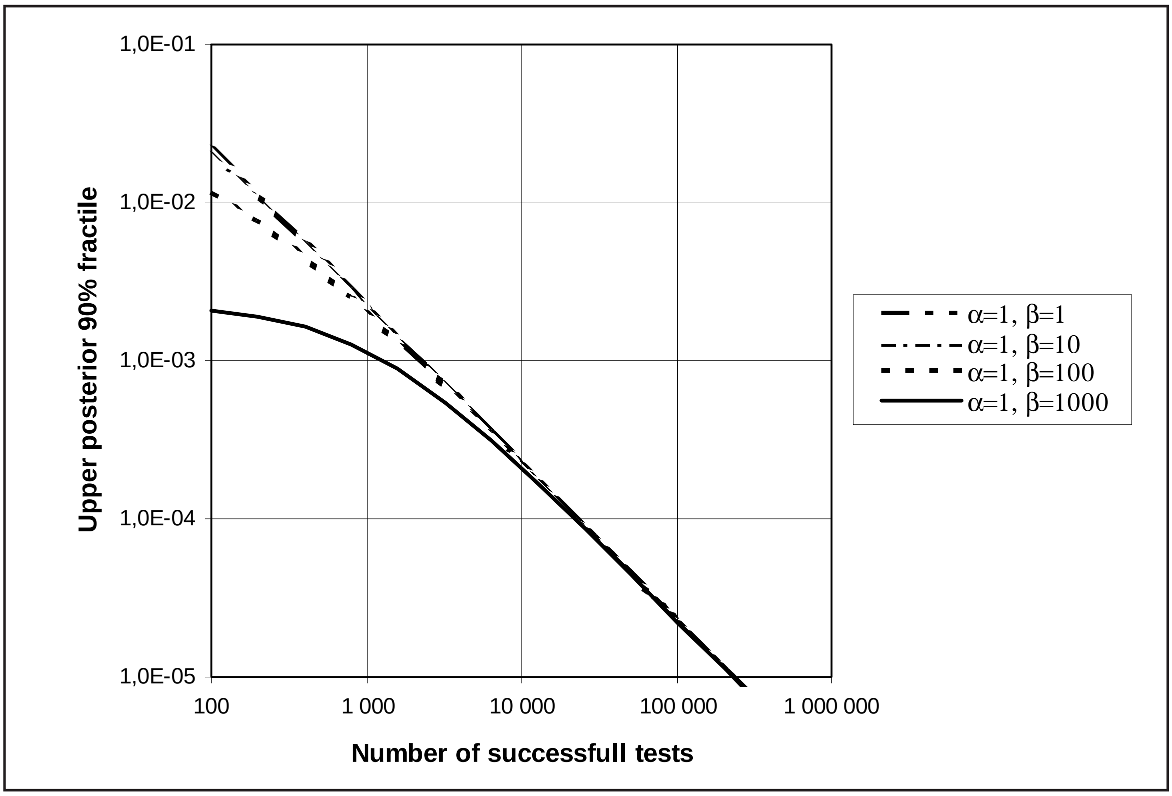

Figure 4. *Bayesian acceptance testing: upper posterior fractiles for selected beta-prior distributions.*

been modified. The method suggests that if a failure is detected, the number of test cases in the next dynamic testing should increase, i.e. reliability requirements should be more stringent.

The problem with dynamic testing is how to deal with found errors. Parnas [1990] recommends on the basis of a classical statistical model that the test procedure should start again from the very beginning with same initial requirements. In this case the criteria for accepting the dynamic test does not change; after software has been updated testing will be redone with the same confidence level and reliability requirement (see section 3.1).

However, one could ask if our prior beliefs towards the system have changed due to found errors. It may well be that the regulator considers the errors as an evidence of low software quality or poor production process, and therefore requires a more stringent testing. Should the requirements be changed and if so, how much, are questions that need to be answered in this situation.

Littlewood & Wright [1995] propose the following Bayesian approach for resolving the reliability estimation problem:

At the start of the test, the number, n_0, of demands that must be executed failure-free for the test to succeed, is computed.

The system is put on test and either successfully executes the n_0 demands, in which case the test stops and the system is declared to have achieved its *pfd* (probability of failure upon demand) requirement, or a failure is observed on demand s_1 $(<n_0)$, in which case the test is stopped.

In the light of the evidence of one failure in s_1 demands, we compute the number, n_1, of further demands that must be executed failure-free for the next test to succeed and stop.

The system is put on test again and either successfully executes the n_1 demands, in which case the test stops and the system is declared to have achieved its *pfd* requirement, or a failure is observed on demand s_1+s_2 $(s_2<n_1)$, in which case this test is stopped.

In graphical form (Fig. 5), the initial amount of demands is n_0, the failure is observed on demand s_1 $(<n_0)$, and a new number of demands after the failure is n_1.

It is important to notice that between steps 2 and 3 the fault has been removed. The other alternative is naturally to continue testing regardless of the failure, and hope that the system will finally reach the required reliability level. However, in this approach it is assumed that the cause of the failure is removed, and that the assumed reliability of the system remains the same. The latter assumption simply means that we do not want to change our a priori knowledge about the system.

For solving the stopping rule in *pfd*-based testing, the Bayesian framework is used. Before calculations can be done, some assumptions have to be made:

As in section 3.1, the successive demands are assumed to be statistically independent Bernoulli trials; *p* is the probability of failure per demand. Thus the basic random variable, number of failures *(K)* in *n* demands follows the Binomial distribution (given p), i.e.

$$P(K=k|\,p,n)=\binom{n}{k}p^k(1-p)^{n-k} \tag{21}$$

If $k=0$ we have the situation already discussed in section 3.1.

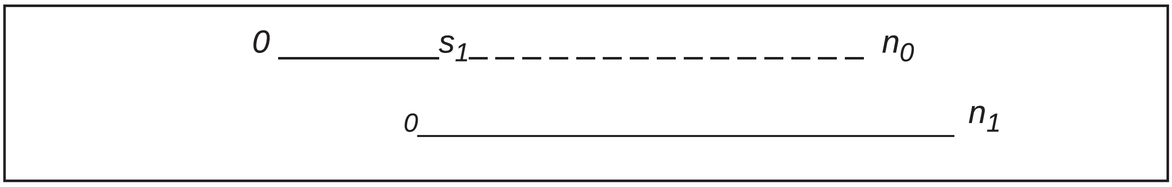

Figure 5. *Number of additional tests required.*

As in section 3.1, our a priori knowledge about the parameter of interest, *p*, must be presented. Here the prior distribution is selected from the conjugate beta-family (see equation (16))

The posterior distribution of *p* is a beta distribution, given by:

$$g(p|K=k,n,\alpha,\beta)=g(p|E,\alpha,\beta) \\ =\frac{1}{B(\alpha+k,\beta+n-k)}p^{\alpha+k-1}(1-p)^{\beta+n-k-1}. \tag{22}$$

The requirement can now be expressed as a pair *(p_U, γ)* so that

$$P(p\le p_U|E,\alpha,\beta)\ge\gamma, \tag{23}$$

where γ is the confidence level.

According to recommendations of Littlewood & Wright [1995] one should apply the uniform or ignorant prior (*$\alpha = 1$, $\beta = 1$),* which leads to the posterior

$$g(p|E,\alpha,\beta)=\frac{1}{B(1+k,1+n-k)}p^{k}(1-p)^{n-k}. \tag{24}$$

Next we assume that system has failed *j* times and that the failures have occurred at tests *s_1, (s_1+s_2), ..., (s_1+s_2+...+s_j).* After *j^th^* failure the system has been tested totally

$$n=\sum_{i=1}^{j}s_i \tag{25}$$

times, and the posterior distribution based on uniform prior is

$$g(p|E,\alpha,\beta)=\frac{1}{B(1+j,1+n-j)}p^{j}(1-p)^{n-j}. \tag{26}$$

After *n^th^* test, the system doesn't fulfill the reliability requirement, and we are interested to determine, how many additional successful tests are needed to achieve the reliability requirement (23). We denote by *n_j* the number additional successful tests, and when the additional tests have been performed the total number of tests will be *n+n_j*.

The posterior distribution after these tests is

$$g(p|E,\alpha,\beta)= \\ \frac{1}{B(1+j,1+n+n_j-j)}p^{j}(1-p)^{n+n_j-j}, \tag{27}$$

and it must, in spite of failures, fulfill the requirement. The minimum number additional successful tests is the smallest *n_j* which satisfies

$$\int_0^{p_U}\frac{1}{B(1+j,1+n+n_j-j)}p^{j}(1-p)^{n+n_j-j}\,dp\ge\gamma. \tag{28}$$

This formula in fact represent the cumulative posterior distribution, based on evidence consisting of several parts (the number of the parts is *j*+1) of failure-free behaviour of the system. Had we detected only one failure, then the posterior distribution would contain two parts: *s_1*-1 demands before the failure and then the calculated, additional number of demands after the failure, *n_1*. The fact that when the failure happens, does not change the posterior distribution, is due to the Bernoulli-trial model assumption. The calculation of *n_j* from (28) can be made numerically.

In Tab. I we present the number of tests required to demonstrate failure probability 10^{-3} /demand for some cases.

Conclusions from the model are obvious: the time of the failure in the test is not important. No matter whether the failure happens in the beginning of the test or on the last demand, the number of test cases is constant. For instance, if the number of failures is two, the pfd-based approach requires exactly 8402 test cases. According to the approach, it is totally unrelevant, if the failures did happen on the two first demands, or on the two last demands. This is due to the assumption that the test sequence is a Bernoulli trial.

The last column of the Tab. I reveals an interesting feature in the approach. The trend of the increment in the number of demands with *j* is decreasing, whereas common sense would suggest the very opposite. Common sense reasoning or expert reasoning would probably go as follows: "As

***Table I.** Total number of demands, N, in case of j failures. Required probability of failure is 10^{-3} per demand with confidence level 99%.*

Number of failures, j	Total number of demands, N	Increase in the number of demands, $N_{n+1} - N_n$
0	4 602	2 033
1	6 635	1 767
2	8 402	1 639
3	10 041	1 559
4	11 600	1 504
5	13 104	...

the number of failures grows, the mistrust towards the quality of the software is increasing. To overcome this, I propose an increasing trend in the increment of number of test cases.” The explanation for this observation is the following. In most Bayesian statistical models the increasing evidence reduces the variances of unknown variables, which means that the posterior distribution is narrower. If the mean value of posterior doesn’t essentially change with increasing evidence, then the probability that the variable is smaller than some fixed value (larger that the mean or median) increases.

In the above, we analysed the case in which failures have occurred during testing by applying Bayesian inference. Actually, the essence of the approach is in the determination of Bayesian upper probability or confidence bound for the failure probability parameter *p*. The corresponding classical confidence bounds can also be determined quite easily. Let us consider the same setting as above. Immediately after *j* failures we have tested the system *n* times, and we are interested to know, how many additional successful test must be made in order to reach the reliability requirement. In classical setting the reliability requirement corresponding to (23) is “ the upper classical confidence limit at confidence level γ is at most p_U”. The classical upper confidence level in the case of no failures was determined in equation (12). We recall that the upper classical confidence bound at confidence level γ for the parameter of binomial distribution (i.e. the parameter of Bernoulli trial model or, in our case, the failure probability), when *l* failures in *m* trials have occurred is given by

$$p_U = \frac{(l+1)F_{1-\gamma}(2l+2,2(m-l))}{(m-l)+(l+1)F_{1-\gamma}(2l+2,2(m-l))}, \qquad (29)$$

where $\boldsymbol{F}_\gamma(v_1,v_2)$ is the γ-fractile of a $\boldsymbol{F}$-distribution with v_1,v_2 degrees of freedom (see e.g. Kapur & Lamberson, 1977, Martz & Waller, 1982). In our setting, we have observed *j* failures during *n* tests, and we are interested to find n_j such that

$$p_U = \frac{(j+1)F_{1-\gamma}(2j+2,2(n+n_j-j))}{(n+n_j-j)+(j+1)F_{1-\gamma}(2j+2,2(n+n_j-j))}, \qquad (30)$$

or in other words, we like to know how many successful test are needed to assure that the upper confidence bound at level γ is at most p_U. It is worth noticing that (30) leads rather exactly to the results given by equation (28) (see also Tab. I), which is due to the relationship between beta distribution and $\boldsymbol{F}$-distribution and due to the uniform prior applied in (28). Since the numerical results from the Bayesian and classical approach are almost identical (in the case of uniform prior), the choice between these approaches seems to be unimportant. However, there are some reasons, which make the Bayesian approach more preferable.

Both Bayesian and classical approaches lead to some kind of confidence intervals for *p*. This means, in the Bayesian case, that even if dynamic testing has been successful, there still exist one percent possibility (in the example the confidence level was 99%) that the “true value” of *p* is somewhere between 0.001 and 1 and not below that 0.001. In classical model there is a 1% chance that the successful test result has been produced by *p* that is larger that 0.001. Also, it might be tempting to use the upper bound of *p* to calculate, for instance, the probability of failure for a certain period of time. However, this is not the proper way, as there is no absolute certainty of the system reliability.

Table II. *Total number of demands, N, needed if there have been exactly j failed demands, so as to claim (s_0, γ).*

Number of failures, *j*	**Total number of demands, *N*, for (s_0,γ) = (46, 0.99)**	**Total number of demands, *N*, for (s_0,γ) = (1 000, 0.8215)**
0	4 602	4 602
1	9 229	9 681
2	13 855	14 766
3	18 481	19 852
4	23 107	24 938
5	27 734	30 024

The Bayesian framework, however, admits to predict the probability of failure by using the predictive distribution. The reliability requirement can thus be formulated as a pair (s_0, γ), for which

$$P(\text{“no failures in the next } s_0 \text{ demands”}|E) \geq \gamma. \tag{31}$$

The Bayesian predictive distribution for the number of failures K_f the next (future) n_f demands, given that there has been *k* failures in the past *n* demands, is

$$\begin{aligned} P(K_f = k_f|k,n,\alpha,\beta) &= \int_0^1 P(K_f = k_f|p)g(p|k,n,\alpha,\beta)dp \\ &= \int_0^1 \binom{n_f}{k_f} p^{k_f}(1-p)^{n_f-k_f} \frac{1}{B(\alpha+k,\beta+n-k)} p^{\alpha+k-1}(1-p)^{\beta+n-k-1}dp \\ &= \binom{n_f}{k_f} \frac{B(\alpha+k+k_f,\beta+n+n_f-k-k_f)}{B(\alpha+k,\beta+n-k)}. \end{aligned} \tag{32}$$

If the prior is uniform ($\alpha = 1, \beta = 1$), and we have no observed failures in *n* tests, then the probability that there are no failures at the next s_0 tests is

$$P(\text{“no failures in the next } s_0 \text{ demands”}|E) = \int_0^1 (1-p)^{s_0}(n+1)(1-p)^n dp = \frac{n+1}{n+s_0+1}, \tag{33}$$

from which we can calculate whether the requirement (31) is fulfilled and the number of successful additional demands required to fulfill (31). Similarly, by using (32) it is possible to check whether the requirement (31) is met, given that failures have been observed.

The requirement (31) is not easily interpreted. Its word-for-word meaning is: given that *k* failures have been observed in *n* tests, the system is accepted if the probability that it successfully operates s_0 demands is smaller the γ. To set the requirement one has to choose both s_0, and γ, which is not an easy task. One possibility is to relate (31) with the usual Bayesian acceptance requirement: the posterior upper confidence bound (at level γ) is smaller than a fixed number. Then, assuming that no failures have occurred, and given the number of successful tests which is enough in the case of no failures to meet the usual Bayesian requirement, s_0 is determined so that the requirement (31) is met (by using equation (33)). After that the total number of test are determined for the pair (s_0,γ) given *k* failures by using the equation (32). In Tab. II some (s_0,γ) pairs are considered. First, γ is 0,99 as in *pfd*-example and the corresponding s_0 = 46 is calculated as explained in above 46. Secondly, Tab. II shows the total number of demands also for (s_0,γ)= (1000, 0.8215).

Table III. *The increase in the total number of demands after one failure for the three methods.*

Number of failures, *j*	**Parnas method, *N***	**Pfd-based method, *N***	**Reliability prediction -based method, (s_0,γ) = (46, 0.99), *N***
0	4 603	4 602	4 602
1, on first demand	4 604	6 635	9 229
1, on last demand	9 205	6 635	9 229
2, on the two first demands	4 605	8 402	13 855
2, on the two last demands	12 808	8 402	13 855

3.3 Comparison of approaches

In previous sections we have presented several approaches for demonstrating the systems reliability. Actually, on the basis of the models discussed, we may derive a set of acceptance rules, which may have slightly contradictory interpretations. Since the approaches are based on different assumptions and principles, their results and interpretations should be discussed more thoroughly.

The testing rules may be classified according to the statistical principles (classical vs. Bayesian), and the form of acceptance criteria with respect to failures occurred during tests. The Bayesian approaches may be classified as predictive or nonpredictive. Further, both Bayesian and classical approaches may be static or sequential.

The classical acceptance criteria and rules are based to the expressions of the upper confidence bounds at selected confidence level (see equations (12) and (29)). The equation (12) gives the upper confidence bound in the case where failures are not observed and equation (29) refers to the case with failures. Parnas [1990] proposes a rule according to which the system is accepted if certain number of successful tests are performed (see eqn (13)), or if failures are observed, the testing process is started again. We refer this approach as Parnas method. Another possibility to deal with failures during test is to apply the upper confidence bound given by (29), and accept the system when the upper confidence bound is below a given value.

The results of all three methods for the case in which the failure probability to be demonstrated is 10^{-3}/demand are summed up in Tab. III, which contains the total number of demands for all the methods in five elementary cases: no failures, one failure and two failures. The consequences of failures are studied with two extreme alternatives: a failure happens on the first demand, and a failure happens on the last demand.

The difference in strategies of methods is quite obvious. In the Parnas method the increase of *N* grows as a function of *already executed* demands with *j*, resulting to a strong reliance on the time of the failures. This is due to the assumption that if failures occur, then the system is changed and the old evidence is not any more relevant. The consequence of this assumption is the great variation of the total number of demands. For instance, if two failures have been detected, the amount of demands may vary between 4 603 and 12 808. The basic form of the Parnas method is based on classical statistical methods; however, it is possible to apply Bayesian thinking also in this method. The results are essentially the same, but they depend slightly on the prior distribution.

The Bayesian approaches ignore the time of the failure in the test, it is not considered important. No matter whether the failure happens in the beginning of the test or on the last demand, the number of test cases is constant. This assumption is acceptable, if the software is not changed after the observation of failure. However, if the software is modified, then this assumption corresponds to the case in which the failure probability of the system doesn't change although the software if different after the failure. This is probably not realistic. If the assumption is accepted, the only remaining question is: how many additional successful test should be made? In this respect the reliability prediction-based approach is much more stringent, since the number of additional tests is larger. The increment is more than two times the increment of the *pfd*-based method and stays very close to linear with *j*. However, also this method is based on the same assumption of the failure probability than the *pfd*-based method: only the form of acceptance criterion is different. The conclusion Littlewood and Wright draw out from the comparison of the two Bayesian approaches, is that testing requirements in the *pfd*-based method are not conservative enough. Littlewood and Wright recommend that the latter method (i.e. the reliability prediction-based) should be used in determining the stopping rule in dynamic testing. In fact, the reliability prediction-based method has been applied as a stopping rule in Sizewell-B protection system certification.

3.4 Other approaches

3.4.1 Sequential approaches

The acceptance rules discussed above are such that they are continued until the system is accepted. However, it may be advantageous to reject the system when certain evidence occurs. This is flexibly done by applying sequential testing procedures. In sequential statistical testing procedures, the system is either accepted, rejected or the test is continued on the basis of observed test results (see e.g. Kapur & Lamberson, 1977). It is worth noticing that the basic assumption behind the sequential approaches is that the test sequence is a Bernoulli trial, and the failure probability remains constant although the system may be modified after a failure.

In case of binomial or Bernoulli trials the one must be able to accept the null hypothesis H_0: $p \le p_U$ or accept the alternate hypothesis H_1: $p > p_0$. Here p_0 is a value of p such that if $p = p_0$, then the probability of accepting H_0 is $(1-\gamma)$. Let p_1 be a value of p such that $p_1 > p_o$, and for $p = p_1$ the probability of accepting H_0 is δ. The quantities $(\delta, p_0, p_1, \gamma)$ define the sequential statistical test, which is based on the sequential probability ratio

$$R = \left(\frac{p_1}{p_0}\right)^j \left(\frac{1-p_1}{1-p_0}\right)^{n-j}, \tag{34}$$

in which n is the number of tests and j is the number of failures observed before the n^{th} test.

In sequential statistical testing, it is not possible to make decision between H_0 and H_1, and the testing is continued, if

$$\frac{\delta}{1-\gamma} < R < \frac{1-\delta}{\gamma}, \tag{35}$$

or, equivalently, if

$$\begin{aligned} &\frac{n}{D}\ln\left(\frac{1-p_0}{1-p_1}\right) - \frac{1}{D}\ln\left(\frac{1-\gamma}{\delta}\right) \le j \\ &\le \frac{n}{D}\ln\left(\frac{1-p_0}{1-p_1}\right) + \frac{1}{D}\ln\left(\frac{1-\delta}{\gamma}\right), \end{aligned} \tag{36}$$

where

$$D = \ln\left[\left(\frac{p_1}{p_0}\right)\left(\frac{1-p_0}{1-p_1}\right)\right]. \tag{37}$$

We may present the inequality (36) as

$$A_n \le j \le B_n, \tag{38}$$

which gives procedural rules for accepting the hypotheses or continuing the test: if $j \le A_n$ we accept H_0 after n tests (i.e. we accept the system), and reject H_0 (i.e. reject the system), if $j \ge B_n$, otherwise we continue the testing. The sequential procedure is presented in Fig. 6.

To interprete the above sequential procedure we first note that $1-\gamma$ is the probability that the hypothesis H_0 is accepted when it is true (or more exactly, when $p = p_0$) and that δ is the probability that H_0 is accepted when $p = p_1 > p_0$. In other words, γ is the probability that we erroneously reject the null hypothesis, and δ is the probability that we accept the null hypothesis, although it is not true. Since the sequential test is based on the likelihood ratio, R, given in (34), it actually aims to discriminate two hypotheses "$p > p_1$" and "$p < p_0$ " from each other. The risks of errors are γ and δ.

The expected number of tests until the decision (accept or reject) can be determined by using the operating characteristic curve (O.C. curve), which gives the probability of accepting H_0 given that the true failure probability is p [see Kapur & Lamberson, 1977]. The O.C. curve is given by

$$P_{OC}(p) = \frac{B^h - 1}{B^h - A^h}, \qquad (39)$$

where

$$p = \frac{1 - \left(\frac{1-p_1}{1-p_0}\right)^h}{\left(\frac{p_1}{p_0}\right)^h - \left(\frac{1-p_1}{1-p_0}\right)^h}, \qquad (40)$$

and

$$A = \frac{\delta}{1-\gamma},\ B = \frac{1\text{-}\delta}{\gamma}. \qquad (41)$$

Eqns. (39)–(41) actually present the O.C. curve for each h, which determines uniquely the parameter p. The expected number of tests until the

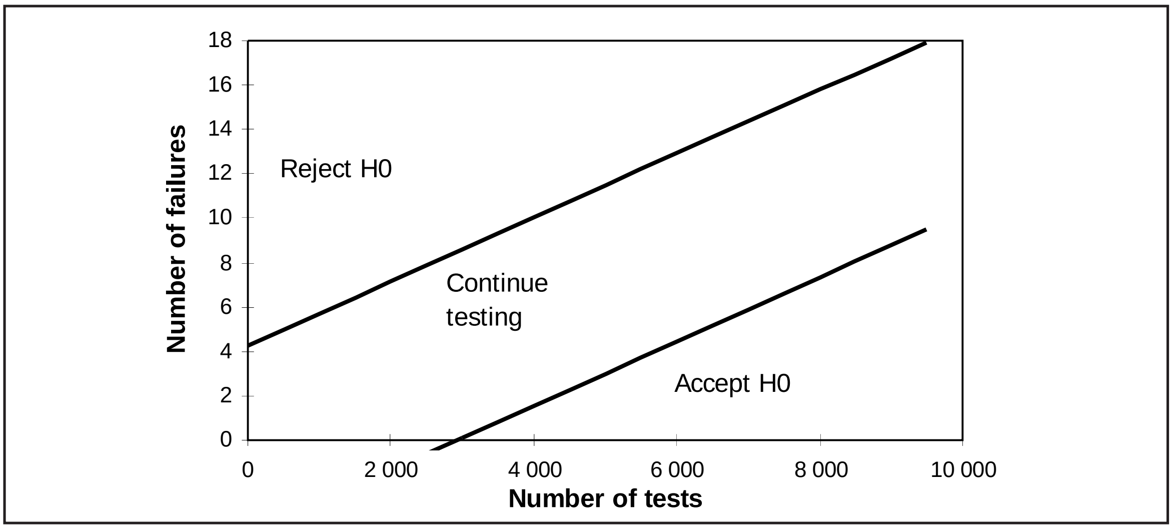

Figure 6. *Classical sequential acceptance testing procedure.*

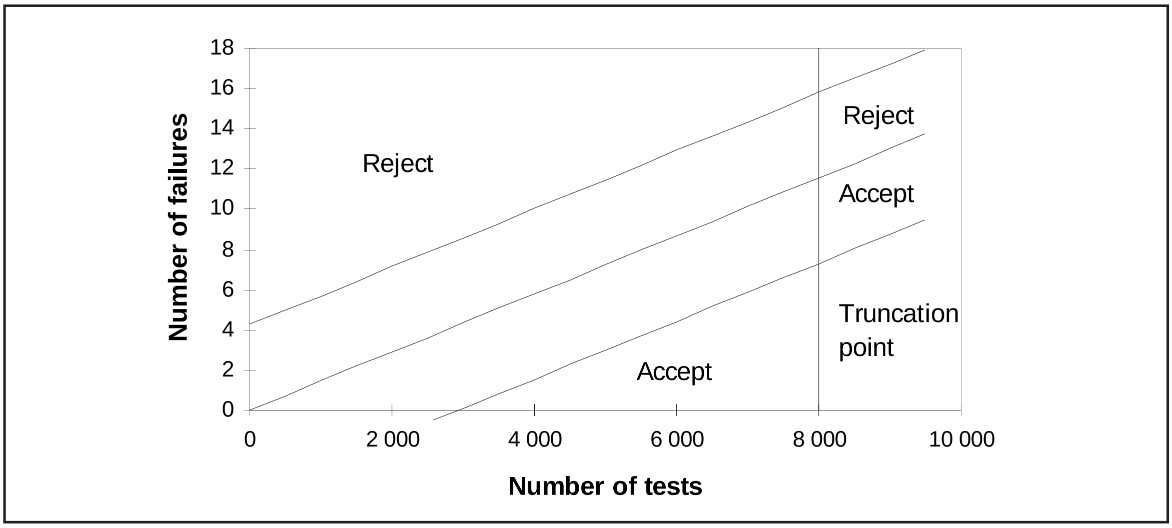

Figure 7. *Classical truncated sequential test procedure.*

decision when the true failure probability is *p* can be evaluated from

$$E(n,p) = \frac{P_{OC}(p)\ln(A) - (1 - P_{OC}(p))\ln(B)}{p\ln\left(\frac{p_1}{p_0}\right) - (1-p)\ln\left(\frac{1-p_1}{1-p_0}\right)}. \quad (42)$$

The number of tests until decision may be rather large, and it may be advantageous to truncate the test. This may be done as presented in Fig. 7, where the test is stopped at the test n_s and the decision is made accordingly.

Kapur and Lamberson [1977] discuss also a Bayesian sequential procedure to discriminate between hypotheses H_0: $p \le p_U$ and H_1: $p > p_0$. The test criterion is based on the posterior probability $P(p>p_0|\boldsymbol{E})$. If this probability is large, we would suspect that H_1 is true. A natural choice for the decision rules is

1. Accept H_0 if $P(p>p_0|\boldsymbol{E}) \le \delta$

2. Reject H_0 if $P(p>p_0|\boldsymbol{E}) \ge 1\text{-}\gamma$

3. Continue testing if $\delta < P(p>p_0|\boldsymbol{E}) < 1\text{-}\gamma$.

The time to decision in Bayesian sequential testing may also be long. For that purpose it may be necessary to truncate the test. Following the reasoning behind the classical sequential test we first select two values, p_0 and p_1 such that $p_1 < p_0$. If $P(p>p_0|\boldsymbol{E})<\delta$, then most likely $p<p_0$, and we may accept the system. On the other hand, if $P(p<p_1|\boldsymbol{E})<\gamma$, then most likely $p>p_1$, and we reject the system. The region $p_1 < p < p_0$ is a compromise region that must be agreed upon. In addition to the above Bayesian sequential testing, it may be possible to develop similar approaches for the predictive approach.

3.4.2 A dynamic Bayesian approach

The approaches discussed above are based on the assumption that *the system either is not modified after observation of failures or that the failure probability remains constant independently on the system modifications*. The only exception to this is the Parnas approach, which is based on the assumption that after an observation of failure, the system is totally new and the evidence from earlier test cannot be used. The reality is probably between these two extreme cases.

One possibility to describe the situation in more realistic way is to apply dynamic Bayesian models. In these models it is possible to assume that the failure probability changes after the repair of each failure. However, the new failure probability depends on the earlier, and thus the evidence from earlier test can be utilized in the reliability estimation. In the following, the dynamic Bayesian approach is described in a schematic way.

It is assumed that the initial failure probability of the system is p_1, which is unknown, and thus it is modelled by a probability distribution, $p_1 \sim \pi(p_1)$. The first failure occurs at the test n_1. The distribution $\pi(p_1)$ is the prior distribution for p_1. The corresponding evidence is E_1 = *{1 failure in n_1 tests}* and it is describe by the likelihood function

$$L_1(E_1|p_1) = p_1(1-p_1)^{n_1-1}, \quad (43)$$

since it is assumed that the first n_1 tests form a Bernoulli sequence.

The distribution of p_1 is updated by using the Bayes' rule and the likelihood $L_1(E_1|p_1)$, and the resulting posterior distribution is

$$\pi(p_1|E_1) = \frac{\pi(p_1)p_1(1-p_1)^{n_1-1}}{\int_0^1 \pi(p_1)p_1(1-p_1)^{n_1-1}dp_1}. \quad (44)$$

When the first failure occurs, the system is modified and the failure probability gets a new value, p_2. If it is assumed that p_2 doesn't depend on p_1, the evidence E_1 doesn't give any information on p_2, and the model is equivalent to the Bayesian version of the Parnas model. However, if there is dependence between p_2 and p_1 , E_1 says something on p_2, too. The dependence between p_2 and p_1 can be modelled by a conditional distribution, $\pi(p_2|p_1)$. The prior distribution for p_2 is determined on the basis of the evidence E_1 and the conditional distribution $\pi(p_2|p_1)$:

$$\pi(p_2|E_1)=\int_0^1\pi(p_2|p_1,E_1)\pi(p_1|E_1)dp_1=\int_0^1\pi(p_2|p_1)\pi(p_1|E_1)dp_1, \tag{45}$$

since it is assumed that p_2 depends on E_1 only through p_1.

The evidence on p_2 from test is similar than that on p_1, i.e. one failure is observed after executing n_2 tests, and the corresponding likelihood, $L_2(E_2|p_2)$, follows the same kind of law as E_1 (see eqn. (43)). Combining (45) and the likelihood $L_2(E_2|p_2)$ similarly as (44), the updated distribution $\pi(p_2|E_2,E_1)$ is obtained. The procedure is continued recursively, and finally the distribution $\pi(p_k|E_k,...,E_2,E_1)$ is obtained after observing k failures. The failure probability after k^{th} failure is removed is p_{k+1} and uncertainty about it is described by the distribution

$$\begin{aligned}\pi(p_{k+1}|E_k,\ldots,E_2,E_1)&=\int_0^1\pi(p_{k+1}|p_k,E_k,\ldots,E_2,E_1)\pi(p_k|,E_k,\ldots,E_2,E_1)dp_k\\&=\int_0^1\pi(p_{k+1}|p_k)\pi(p_k|,E_k,\ldots,E_2,E_1)dp_k.\end{aligned} \tag{46}$$

Next it is assumed that the system has been successfully tested n_{k+1} times, and this evidence is denoted by E_{k+1}. Again it is possible to determine the updated distribution corresponding to this evidence

$$\pi(p_{k+1}|E_{k+1},\ldots,E_2,E_1)=\frac{\pi(p_{k+1}|E_k,\ldots,E_2,E_1)(1-p_{k+1})^{n_{k+1}}}{\int_0^1\pi(p_{k+1}|E_k,\ldots,E_2,E_1)(1-p_{k+1})^{n_{k+1}}dp_{k+1}}. \tag{47}$$

The distribution (47) expresses the uncertainty of the failure probability of the system after observing the evidence $E_{k+1},...,E_2,E_1$. The system can be accepted, for example, at the first time when the upper 99% fractile of the posterior distribution (47) is below the stated limit.

In order to apply the above model in practice, one must specify, firstly, the prior distribution $\pi(p_1)$ and, secondly, the conditional distributions $\pi(p_{i+1}|p_i)$, which describe how the failure probability of the modified system is related to that of the earlier version. This may be difficult, and the results may depend strongly on the parametrization of the distributions. One possibility to model $\pi(p_{i+1}|p_i)$, is to apply logistic regression type of model, in which

$$\mathbf{ln}\left(\frac{p_{i+1}}{1-p_{i+1}}\right)=\mathbf{ln}\left(\frac{p_i}{1-p_i}\right)+\omega_{i+1}, \tag{48}$$

in which ω_{i+1} ~$N(\mu,\sigma)$ is a normally distributed noise term. Independently on the parametrization of the distributions, it is not possible to determine the posterior distributions of the model analytically, and one must use numerical methods, e.g. those based on Monte Carlo sampling (see Tanner, 1991).

To illustrate the above approach the following example is considered. It is assumed that three failures are observed during testing. The succesive times between failures are *1000, 1500*, and *2000*. After the third failure *400* successful has been performed. The total number of tests is *4900*. Assuming that the initial failure probability, $p_{1,}$ follows the prior distribution defined by

$$\mathbf{ln}\left(\frac{p_1}{1-p_1}\right)=\omega_1, \tag{49}$$

in which ω_1 *~N(-2,3)* is normally distributed and that the noise terms follow the normal distribution *N(0,2)*, it is possible to apply Monte-Carlo simulation to obtain the distribution of failure probability after the third failure (p_4). The initial failure probability has rather broad distribution; 5% fractile is *1.0*10^{-3}* , the median is *1.1*10^{-1}* and the 95% fractile is approximately *1.0.* The prior distribution of p_1 (the initial failure probability) is presented in Fig. 8.

When a failure occurs, the system is modified and the failure probability changes according to the equation (48). The jump of the failure probability may be negative or positive with the same probability, since it is assumed that ω_i follows the normal distribution with zero mean, ω_i ~ *N(0,2)* . However, the change may be rather large compared to the present failure probability, since the variance of ω_i is rather large. The evidence after the first failure indicates that the failure probability is approximately *10^{-3}*; and the corresponding posterior expected value obtainde by Monte-Carlo simulation is indeed *1.3*10^{-3}*. The posterior expecteted value of the failure probability after the second failure is *8.1*10^{-4}* and immediately after the third failure *5.0*10^{-4}*. The posterior expected value after the the whole test (*4900* test, *3* failures) is *5.6*10^{-4}*. The posterior 5% fractile is *8.6*10^{-6}*, the median is *2.4*10^{-4}* and the 95% fractile *2.0*10^{-3}*. The posterior distribution based on the whole body of evidence is in Fig. 9.

Given that some failures have occured during testing, the additional successfull test required to meet the reliability requirement by using the above method lie between the Parnas method and the simple Bayesian approach. However, in the Bayesian dynamic approach explicit assumptions on the changes of the failure rate after software repair are not needed: the recursive Bayesian nature of the dynamic approach fits the change in failure rate according to the evidence from tests. However, it is possible to include stromger assumptions on the impact of software modificarions into the model. For example, the possible software reliability growth can be modelled.

3.5 Tests based on approximately specified operational profiles

Above we have discussed the analysis of test with large number of successes give that the test cases are samples from the "true" operational profile. Usually, the operational profile can be determined only approximately, and there is possibility that the test cases do not faithfully represent the actual operational profile. This fact causes difficulties into statistical reliability estimation.

As stated above, the tests based on exactly specified operational profile give approximative answers to the question "what is the probability that an input from the operational profile leads to an incorrect response?". On the another hand, the operational profile is defined by Musa et al [1997] as "the set of run types that the program can execute along with the probabilities which they will occur". Mathematically this definition is equivalent to the probability distribution of the run types. As a probability distribution, the operational profile represents the uncertainty about the realization of different input states in the input sequence.

If the operational profile used in tests doesn't correspond exactly to the "true" profile, then the reliability estimates are biased. The worst situation is met when the operational profile is "too deterministic", i.e. tests generated from it are concentrated within a very small area in the input domain. In that case the input space is not covered sufficiently during testing.

The statistical analysis of the use of improperly specified operational profiles is not straightforward and it is left outside the scope of this report. To make such an analysis one should be able to model how the "true" and "approximate" operational profiles are related to each other. This requires suitable parametrizations of the operational profiles, and one possibility is to apply Markov chain techniques [see e.g. Whittaker & Poore, 1993, and Whittaker & Thomason, 1994]. In the case of Markovian models the operational profiles are described by using finite number of parameters, and the comparison between alternative models is easier.

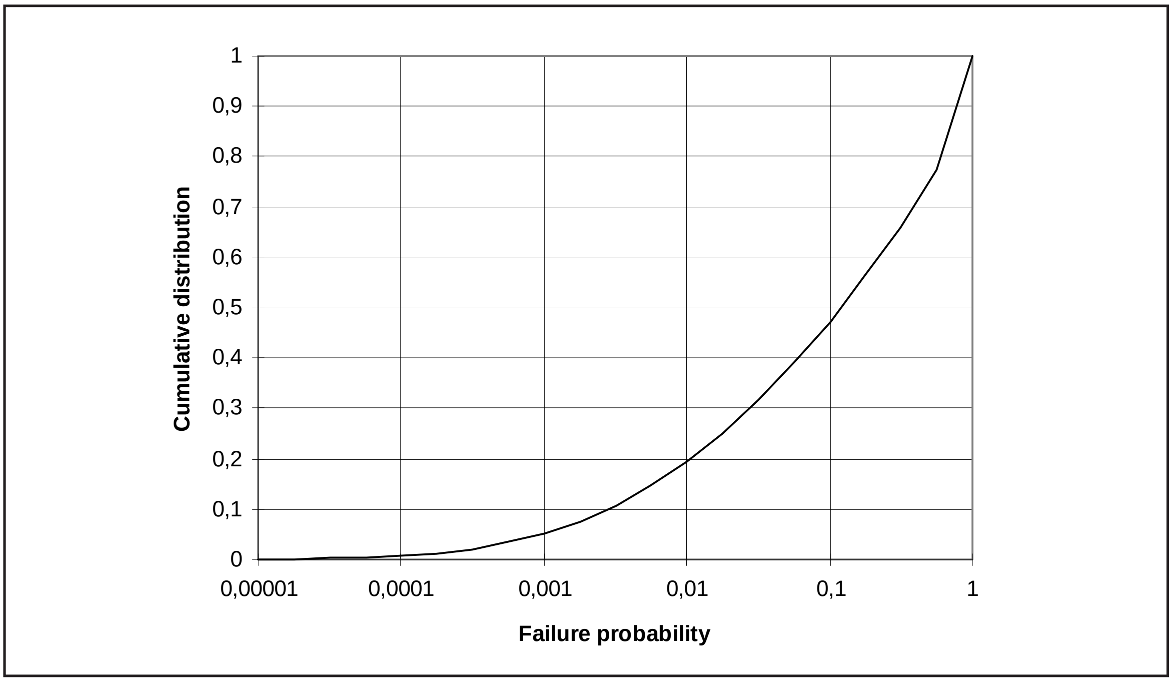

Figure 8. *The prior distribution of the initial failure probability.*

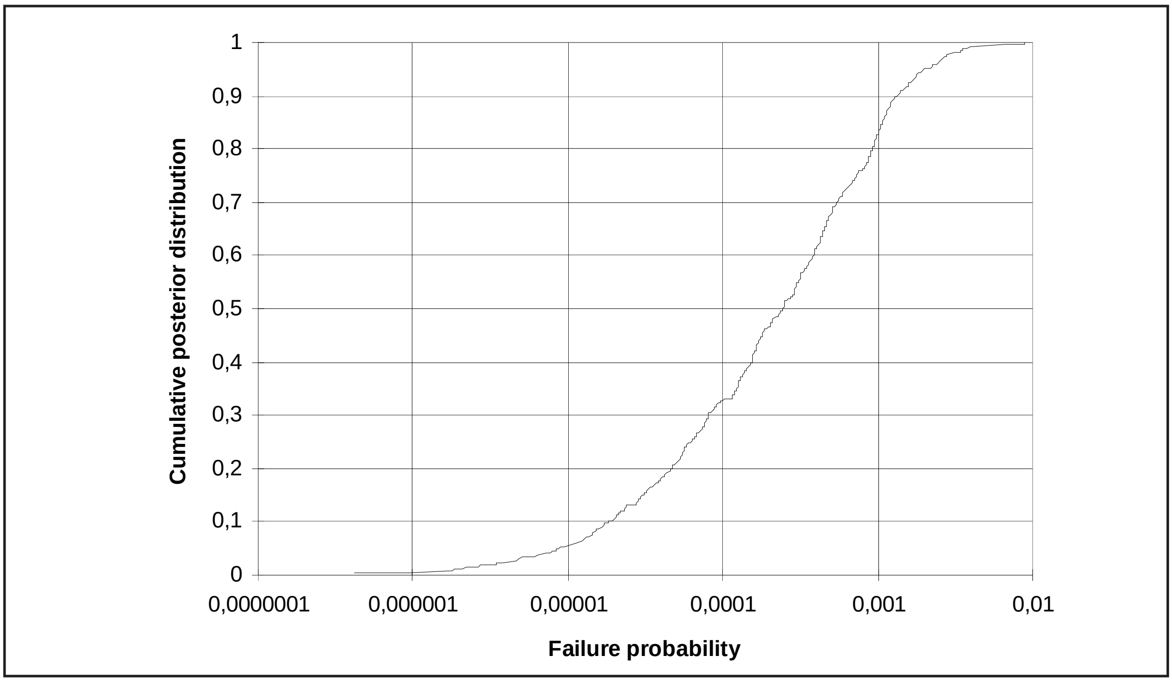

Figure 9. *The posteroir distribution of failure probability.*

4 CONCLUSIONS

The statistical reliability assessment of software based automation systems involves several aspects. First, one has to choose between statistical methods and principles. Then, given the method, one must specify the confidence level or acceptance criteria compatible with the methods. Finally, the method must be applied consistently and the evidence from the tests must be evaluated critically, not only statistically using the selected method, but also qualitatively with respect to the characteristics of the observed errors.

The statistical analyses of reliability demonstration tests discussed above are rather simple, and they can be programmed into spreadsheet programs (e.g. Excel). Further, their application doesn’t require a lot of calculation efforts. It is not necessary to develop new software for statistical analyses, since standard statistical packages provide tools for this purpose. However, the whole test process can be automated, and the automated test harness may also include tools for statistical analyses.

The statistical methods proposed for analysis of test results are based on various assumptions, and they are based on different statistical principles. The most critical assumption deals with the treatment of failures occurred during testing. The Parnas method gives the modified software—after the error has been detected and corrected—a fresh start for dynamic testing, ignoring the evidence obtained before the modification. As Parnas puts it: “Because even small changes can have major effects, we should consider data obtained from previous versions of the program to be irrelevant”.

Another approach is to assume that the failure probability of the software remains the same independently on the modifications or repairs made after occurrences of failures. The failures observed during tests are seen as evidence of poor quality, and no positive credit is given to the modifications. Equivalently, it is assumed that the testing process is a continuing Bernoulli trial. This allows the use of the whole evidence in assessing the software failure probability. Compared to the Parnas method, the total number of tests needed to meet the reliability requirement may be remarkably smaller, and further, is not depending on the position of the failed tests in the sequence of individual test cases (compatibly with the Bernoulli trial assumption).

In addition to of static Bernoulli model, it is possible to apply the classical and Bayesian sequential procedures for acceptance testing as discussed in section 3.4.1. The only difference between these methods and the conventional Bernoulli model is that instead of rejecting or accepting the systems, there is the third alternative to continue the testing. In the sequential models, it is assumed that the failure probability remains the same independently on the modifications.

The origin of differences between the Parnas and above approaches is in the error correction procedure that can cause unpredictable changes in the reliability of the software. The Parnas approach forgets that corrective actions have ever been done, and the other methods assumes that the *pfd* (probability of a failure per demand) remains the same. A method which can be seen as a compromise between the Parnas method and the method based on pure Bernoulli assumption, was illustrated in section 3.4.2. It is a dynamic Bayesian method, in which the evidence from the whole testing process can be utilised, and in which the failure probability of the software changes randomly after each modification. In this method, the dynamics of the failure probability is modelled with a stochastic process, and after error correction the reliability may be better, the same, or worse than before.

In addition to the above methods, also a predictive Bayesian approach was considered in this study. The model is based on the assumption that the failure probability remains the same during the process. The form of the acceptance criterion adopted in this method is complicated and it consists of stating acceptance limits to the probability to obtain a failure free test series of given length with a fixed confidence. The method is more conservative than the (Bayesian and classical) Bernoulli models, in the sense that the total number of tests required is larger. However, it is not as conservative as the Parnas method. The problem with this method is the complexity of the acceptance criterion, which is not easily interpreted.

All the approaches discussed above require the specification of various confidence levels, which determine the risk to accept a system with too large failure probability, or conversely to reject a system with acceptable failure probability. The statistical basis of the confidence statements and concepts was discussed in more detail in section 2. It must be recalled that depending on the statistical principles (e.g. classical vs. Bayesian), the various confidence statements have different interpretations.

The choice of confidence levels depends basically on the decision makers risk attitude. The decision maker, which in this case is the safety authority, must choose the confidence levels in such a way that the risk is acceptable. In choosing the confidence levels, the decision maker must evaluate the costs of accidents due to failures of the system, consider other costs e.g. due to testing and finally choose the confidence level so that the costs balance each other. The problem here is that the costs are not only monetary, and they cannot be measured with same units. Further, many of the factors having impact on the decision are connected to safety culture, which is not easily evaluated. The application of decision analytic thinking could be one approach when confidence levels are chosen.

The choice between Bayesian and classical statistical inference is an issue of continuing debate. The orthodox Bayesians reject all other methods, although they may lead to practically same results. The reason for rejection of other approaches is the fact that (according to Bayesians) the classical statistics is not following the rules of probability calculus. There is much truth in this opinion: indeed such principles as the likelihood principle are violated when classical statistics is applied. Further, consistent application of Bayesian methods makes it possible to use almost any type of evidence, which can be coded into probability models. This is, however, made on the cost of objectivity: according to Bayesians the probabilities are subjective. The flexibility of Bayesian models is an advantage, which should be utilised also in the reliability assessment of software based systems. The basic version of the Parnas method is based on classical statistical concepts (e.g. classical confidence levels), but the Bayesian analysis based on the same assumptions is easily made. Similarly, it is possible to develop classical counterparts for the simplest Bayesian models. However, dynamic or predictive Bayesian methods can not be easily translated to classical frequentistic models. From the practical point of view, the differences between the decisions derived from Bayesian and classical models are in many cases the same.

The assessment method needs to be supported by an assessment framework that is able to deal with the non-quantitative issues involved in the assessment. These include for instance error classification and analysis. From the point of view of reliability assessment methods, the basic procedure of error detection and correction can be repeated over and over again. In practice, several repetitions are certainly rare especially if the errors are classified as safety critical. Continuing the testing of a system which has high reliability and safety requirements and which has already revealed many errors, is waste of time and resources. Other kind of assessment methods should then be applied to uncover the real reason to failures.

REFERENCES

Adam EN. Optimizing preventive maintenance of software products. IBM Journal of Research and Development, vol.28, no.1. 1984.

Anderson J-O. Experience from licensing and installing programmable electronic in the power range monitoring safety system in Barsebäck NPP. EHPG Meeting, Storefjell, Norway, 8–12.3.1993: 1–7 + apps. 8 pp.

Berger JO, Wolpert RL. The likelihood principle. Second edition. Hayward, California, Institute of Mathematical Statistics. 1988: 1–208.

Box GEP, Tiao GC. Bayesian inference in statistical analysis. Addison-Wesley Publishing Company, Reading. 1972: 1–588.

Bunn DW. Applied decision analysis. McGraw-Hill Book Company, New York, 1984: 1–251.

Butler RW, Finelli GB. The infeasibility of quantifying the reliability of life-critical real-time software. NASA Langley Research Center. 1992.

Fine TL. Theories of probability, An examination of foundations. Academic Press, New York. 1973.

Goel AL. Software reliability models: Assumptions, limitations and applicability. IEEE Transactions on Software Engineering. SE-11,12 (Dec.). 1985: 1411–1423.

Haapanen P, Heikkinen J, Korhonen J, Maskuniitty M, Pulkkinen U, Tuulari E. Feasibility studies of assessment methods for programmable automation systems. STUK-YTO-TR-93. 1995: 1–54 + app. 1 p.

Holmberg J, Pulkkinen U. Regulatory decision making by decision analyses. report STUK-YTO-TR 70, Finnish Centre for Radiation and Nuclear Safety, Helsinki. 1993: 1–31 + apps. 3 pp.

IEEE. 1994. IEEE Standards Collection: IEEE standard classification for software anomalies.

Kapur KC, Lamberson LR. Reliability in engineering design. John Wiley & Sons, New York. 1977: 1–586.

Leveson N. Software safety; why, what and how. ACM Computing Surveys, vol. 18, no 2. 1986.

Littlewood B. 1980. Theories of software reliability: How good are they and how can they be improved? IEEE Transactions on Software Engineering. SE-6 (Sept.). 1980: 489–500.

Littlewood B. 1994. Learning to live with uncertainty in our software. SHIP project report.

Littlewood B, Wright D. On a stopping rule for the operational testing of safety-critical software. SHIP project report. 1995.

Mann NR, Schafer RE, Singpurwalla ND. Method for statistical analysis of reliability and life data. John Wiley & Sons. New York. 1974: 1–564.

Martz HF, Waller RA. Bayesian reliability analysis. John Wiley & Sons, New York. 1982: 1–745.

Musa JD, Iannino A, Okumoto K. Software reliability. Measurement, Prediction, Application. McGraw-Hill Book Company. New York. 1987: 1–621.

Nelson W. Applied life data analysis. John Wiley & Sons. New York. 1982: 1–634 p.

Parnas DL, van Schouwen AJ, Kwan SP. Evaluation of safety-critical software. Communications of the ACM, vol. 33, no. 6, June 1990.

Ravn AP (ed.). Embedded, real-time computing systems. Volume I, ESPRIT BRA 3104, Provably Correct Systems, ProCoS, Draft Final Deliverable, October 1991.

Tanner MA. Tools for Statistical inference. Observed data and data augmentation methods. Springer-Verlag, New York. 1991: 1–110.

Whittaker JA, Poore JH. A Markov analysis of software specifications. ACM Transactions on Software Engineering and Methodology, Vol. 2, No 1, 1993: 93–106.

Whittaker JA, Thomason MG. A Markov chain model for statistical software testing. IEEE Transactions on Software engineering, Vol 20, No 10, 1994: 812–824.