SoRel: A tool for reliability growth analysis and prediction from statistical failure data
Karama Kanoun, Mohamed Kaâniche, Jean-Claude Laprie, Sylvain Metge

To cite this version:

HAL Id: hal-00852434
https://hal.archives-ouvertes.fr/hal-00852434
Submitted on 19 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
SoRel: A Tool For Reliability Growth Analysis and Prediction From Statistical Failure Data*

K. Kanoun  M. Kaâniche  J.-C. Laprie  S. Metge**
LAAS - CNRS
7, Avenue du Colonel Roche - 31077 Toulouse Cedex (France)

Abstract*

This paper presents a tool for Software (and hardware) Reliability analysis and evaluation: SoRel. The tool implements a global method for reliability follow up and evaluation in presence of reliability growth due to design fault removal. SoRel is composed of two parts: application of trend tests and reliability growth models. The paper presents the method, trend tests and reliability growth models implemented in SoRel, shows how they can help during the validation process and describes some functionalities of the tool. Main features of the demonstration are outlined.

Introduction**

The objectives of software reliability evaluation in presence of reliability growth are numerous and are closely related to the point of view adopted (the supplier or the customer), and the life-cycle phase concerned. The supplier is interested in the management of the validation and maintenance activities whereas the customer is more concerned by the reliability of the resulting product in operational life. SoRel — which is a tool for Software reliability analysis and prediction — is in achieving these objectives thanks to the bined use of trend tests and of reliability growth models. It provides qualitative and quantitative elements concerning, for instance, a) the evolution of the reliability response to the debugging effort, b) the estimation of number of failures for the following periods of time so as to plan the test effort and the numerical importance of test and/or maintenance team and c) the prediction of reliability measures such as the mean time to failure, failure rate or the failure intensity.

SoRel is based on a global method for software reliability follow up and evaluation that has been developed at LAAS and applied to several real-life systems (see e.g., [7] or [8]). This method relies on qualitative and quantitative analyses, it is intended better define the users' real needs in the field of software reliability. It is briefly reviewed in the paper.

The paper is divided into three sections. Section 1 outlines the method implemented by SoRel, gives general overview of the tool and shows the type of results that can be obtained from SoRel. Section 2 is devoted to the description of the tool. Section 3 describes the demonstration.

1. SoRel general presentation

SoRel is composed of two modules allowing respectively the application of trend tests and reliability growth models. It is able to operate on two types of failure data a) inter-failure times and b) number of failures per unit of time (i.e., failure intensity), allowing application of two types of reliability growth model respectively time domain and interval domain as called [15]. Figure 1 gives the organization of SoRel as it is seen by the user. The reliability evolution is analysed through trend test application. Selection of the model to be applied is based on the result of trend tests and the objectives the analysis. The remaining part of this section presents the trend tests and the reliability growth model implemented in SoRel.

---

*The work reported in this paper was partially supported by the ESPRIT Basic Research Action on Predictably Dependable Computing Systems (Action no. 3092).

**Now, with CEP-SYSTÈMES Toulouse.
Two reliability trend tests are available: the arithmetic mean and the Laplace test— for both inter-failure times and failure intensity data.

1. Arithmetical mean test
   This test consists of calculating $\tau_k$, the arithmetic mean of the first $k$ inter-failure times (resp. number of failures per unit of time). When $\tau_k$'s form an increasing series (resp. decreasing), reliability growth can be deduced. This test is very simple and its interpretation is easy as it is directly related to the collected measure.

2. Laplace test
   The Laplace test, which is a statistical test, consists of calculating the Laplace factor, $u$, whose expressions are given in Figure 2. In practice, in the context of reliability growth, negative values of the Laplace factor suggest reliability growth whereas positive values suggest reliability decrease; values oscillating between -2 and +2 indicate stable reliability.

These practical considerations are deduced from the significance levels associated with the statistics, for instance, for a significance level of 5% the null hypothesis "no trend against trend" is rejected for $u > 1.96$.

$$u(k) = \frac{(c - m) \sqrt{k^2 - 1}}{2 \sqrt{y_k}}$$

$$c = \frac{1}{k-1} \sum_{i=1}^{k-1} s_i$$

$$m = \frac{s_k}{2}$$

$$s_i = \sum_{j=1}^{i} t_j$$

$t_j$ : time interval between failures (j-1) and j
$s_i$ : instant of failure i occurrence
$n$ : number of failures observed

The users of SoRel can utilize the Laplace test as a conventional statistical test. However, in our approach we extended it to identify global and local trends [9]. As the Laplace factor is evaluated using all the data collected up to the unit of time considered, it reflects the global variation of reliability. Local fluctuations can be detected by studying the variation of $u(k)$: for example, when $u$ is positive and tends to decrease it suggests a decrease in the number of failures observed over the considered period which means that, locally, reliability tends to increase although a global decrease is observed. This is summarized in Figure 3. Global reliability decrease over A and B is due to data observed during period A, if the latter are not considered for evaluation purposes, period B will display local and global reliability growth.

<table>
<thead>
<tr>
<th>Inter-failure time data</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u(k) = \frac{c - m}{s_k} \sqrt{12(k-1)}$</td>
</tr>
<tr>
<td>$c = \frac{1}{k-1} \sum_{i=1}^{k-1} s_i$</td>
</tr>
<tr>
<td>$m = \frac{s_k}{2}$</td>
</tr>
<tr>
<td>$s_i = \sum_{j=1}^{i} t_j$</td>
</tr>
<tr>
<td>$t_j$ : time interval between failures (j-1) and j</td>
</tr>
<tr>
<td>$s_i$ : instant of failure i occurrence</td>
</tr>
<tr>
<td>$n$ : number of failures observed</td>
</tr>
</tbody>
</table>
normal situation. Reliability decrease may also result in regression faults. If the duration of the period of ease seems long, one has to pay attention and, in situations, if it keeps decreasing this can point out problems within the software: the analysis of the reasons of this decrease as well as the nature of the rated faults is of prime importance in such situations.

**Reliability growth models** are implemented: the exponential model (Kanoun-Laprie) [11], the Weibull model (Goel-Oukamoto) [4], the S-Shaped model (Yamada et al.) [17] and the doubly stochastic model (Littlewood-Verrall) [12]. The S-Shaped model is an interval domain model, the doubly stochastic model (DS) is time domain whereas the hyperexponential model (HE) and the exponential model (EXP) are both interval domain.

These models allow different kinds of behavior to be selected: HE, EXP and DS model a decreasing failure rate; they yield better results when applied to data laying reliability growth, that is, interval C of Figure 1. These models can be applied also to interval B. Figure 3 discarding failure data pertaining to A. The model is characterized by an increasing failure rate owed by a decreasing failure rate, it produces good results when applied to failure data belonging to an interval such as A-B-C. For D, prediction is delayed until observing an interval of reliability growth.

Figure 4 summarizes the characteristics of these models. Depending on the model, the main quantities that are evaluated are: the mean time to next failure (or MTTF), the failure intensity, the cumulative number of failures and the residual failure rate of the software. It is worth noting that HE is the only model allowing evaluation of the residual failure rate in operation.

Model execution is carried out in two steps: maximum likelihood or least square. Both of them need a numerical optimization procedure to estimate the parameters of the models. For two-parameter models (EXP and SS), the numerical values are obtained via the Newton-Raphson iterative method whereas for three parameter models (HE and DS) they are evaluated via the Powell numerical method [13].

SoRel enables the user a) to determine how well the selected model fits the data and b) to compare estimations issued from several models. The goodness-of-fit criteria are: a) the Kolmogorov-Smirnov statistics [1], b) the prequential likelihood [2] and c) the residue [6]. The first two criteria are evaluated only for inter-failure time data. The residue is evaluated for both failure data types, it is based on the difference between the observed measure and its expectation from the model.

<table>
<thead>
<tr>
<th>Model</th>
<th>$h(t)$ or $\lambda(t)$ shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyperexponential</td>
<td>$h(t) = \frac{\omega e^{\varphi t} + \omega e^{\varphi t} \exp t}{\omega e^{\varphi t} + \omega e^{\varphi t} \exp t}$</td>
</tr>
<tr>
<td>Exponential</td>
<td>$h(t) = N \Phi \exp \Phi t$</td>
</tr>
<tr>
<td>S-Shaped</td>
<td>$h(t) = N \Phi^2 t \exp \Phi t$</td>
</tr>
<tr>
<td>Double Stochastic</td>
<td>$\lambda(t) = \frac{\alpha}{t + \psi(i)}$, $\psi(i) = \beta_1 + \beta_2 i$</td>
</tr>
</tbody>
</table>

Figure 4: Reliability growth models implemented in SoRel

A model can be analysed according to its retrodictive capability and predictive capability. The retrodictive capability expresses model ability in reproducing the observed behavior of the software. The predictive capability reflects the model ability in predicting future behavior of the software, from the observed failure data. Retrodictive and predictive capabilities are measured through goodness-of-fit criteria.

### 1.3. SoRel within the software validation process and reliability evaluation

SoRel has been used to follow up and evaluate the reliability of some of these systems as summarized in Table 1.
For the validation phase, the main results concern the evolution of reliability in response to debugging activities and the prediction of the number of faults that will be activated over the next periods of time \cite{5}. During operation, the objectives of reliability analysis are more complex: we give hereafter some examples illustrated through the results obtained for the first three systems: electronic switching systems (ESS). For the E10-B, the evaluation of the residual failure rate in operation carried out from failure data collected on the hardware in operation \cite{6} allowed the dependability of the ESS to be evaluated (accounting for hardware and software). For the TROPICO-RESSs we have followed complementary approaches \cite{8,10}:

- from the supplier point of view, estimation of the maintenance effort to provide in operation in order to satisfy the correction reports issued from the various customers,
- from the customer point of view, estimation of the residual failure rate in operation in order to evaluate the impact of software reliability on the whole ESS reliability.

SoRel is composed of two modules "TREND" and "MODELS" corresponding respectively to trend analysis and model application. The two modules accept the same input data files which can be created and changed by word processing or graphic editor. Numerical results are displayed immediately on the screen during the execution process. Additionally, the corresponding curves can be plotted upon user’s request. The results are also recorded in the form of ASCII files that can serve as input to other Macintosh applications (such as Excel) allowing instance comparison of results issued from different model applications.

The main menu commands allow cancellation of the last changes (Resume) and exit from the program (Quit). The features specific to each module are described in the rest of the section.

2.1. "TREND" module

Selection of the trend test to be applied is achieved indicated in Figure 5. SoRel prompts the user for the data set type as shown in Figure 6 and then for the input file name (Figure 7). The user can either a) choose the file name among the available data set names in the same directory or b) enter directly the name of the file input (Figure 8).

<table>
<thead>
<tr>
<th>System</th>
<th>Languages</th>
<th>Volume</th>
<th>Observation</th>
<th>Observations System of Failure and CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>E10-B</td>
<td>Assembler</td>
<td>100 k-bytes</td>
<td>3 years</td>
<td>Val. / Op. 1400 58 FR / 136 CR</td>
</tr>
<tr>
<td>TROPICO-R 1500</td>
<td>Assembler</td>
<td>300 k-bytes</td>
<td>27 months</td>
<td>Val. / Op. 15 461 CR</td>
</tr>
<tr>
<td>TROPICO-R 4096</td>
<td>Assembler</td>
<td>350 k-bytes</td>
<td>32 months</td>
<td>Val. / Op. 42 227 CR</td>
</tr>
<tr>
<td>Telecommunication Equipment</td>
<td>PLM-86</td>
<td>$5 \times 10^5$ inst.</td>
<td>16 months</td>
<td>Val. 4 2150 FR</td>
</tr>
<tr>
<td>Work station</td>
<td>various</td>
<td>--</td>
<td>4 years</td>
<td>Op. 1 414 FR</td>
</tr>
</tbody>
</table>

Table 1: Characteristics of some real-life software systems studied by SoRel

**SoRel Description**

SoRel runs on Macintosh II-xx computer equipment with an arithmetical co-processor. The human / machine interface has been denoted special attention. It is menu-driven and uses the multiple management facilities of the Macintosh. The program is modular and new reliability growth tests and models can easily be added. It is written in Pascal (5,000 lines of code) and requires about 200 K bytes of memory. A user guide and a tutorial are available. The user guide aims how to use the tool and provides examples as well as samples of input and output files. The tutorial presents the method, the trend tests and the

![Figure 5: Trend test selection](image)
The trend test may be applied to sub-sets of the data recorded in the selected file in order to highlight the local trend: the user indicates the rank of the first data item to be considered (Figure 9). Figure 10 gives an example of graphical results.

2.2. "MODELS" module

The user selects the model (Figure 11) according to the trend displayed by the data set and to the evaluation objectives. The user has to indicate the input data type as well as the measure to be evaluated (Figure 12), then he is prompted for all the input needed to apply the select model as indicated in Figure 13.

The dialogue areas are defined as follows. Area I for initial Parameter Setting, is needed when the model parameters are evaluated via the Powell method [13]. The user has to supply initial approximations of the parameters at the optimum. Area DP, for Data Partition, allows the user to define a) the data sub-set from which the parameters will be evaluated and b) the prediction interval. Another required input is the use of a window for model Calibration, area C. Finally area V points out the interval over which the Validation criteria will be evaluated (this interval may or may not correspond to the prediction interval). The Set Options command allows the user to adjust some parameters of the optimization procedure, such as the maximum number of iterations and the convergence criteria which have been given fixed values by default.
Figure 12: Input data type and output measure selection

Figure 13: Model application initialization

Figure 14: Numerical results

Figure 15: Graphical results

The demonstration

SoRel will be demonstrated using failure data collected on a real-life software system: the TROPICO-RESS. Data have been collected over about two years ending the end of validation and the beginning of operational life. The two types of input data files (inter-failure times and failure intensity) will be addressed. Since the results are displayed immediately, it is possible to take several executions to show the main features of the tool.

Acknowledgments

We wish to acknowledge the very helpful advice we received during the development of SoRel from our
colleagues, in particular Yves Crouzet and the late and sorely missed Christian Béounes.

References
