



HAL
open science

THE ROLE OF TREND ANALYSIS IN SOFTWARE DEVELOPMENT AND VALIDATION

Karama Kanoun, Jean-Claude Laprie

► **To cite this version:**

Karama Kanoun, Jean-Claude Laprie. THE ROLE OF TREND ANALYSIS IN SOFTWARE DEVELOPMENT AND VALIDATION. IFAC Int. Conf. on Safety, Security and Reliability (SAFE-COMP'91), Nov 1991, Trondheim, Norway. pp.169 - 172. hal-01985291

HAL Id: hal-01985291

<https://hal.science/hal-01985291>

Submitted on 17 Jan 2019

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.

THE ROLE OF TREND ANALYSIS IN SOFTWARE DEVELOPMENT AND VALIDATION*

Karama KANOUN and Jean-Claude LAPRIE
LAAS-CNRS, 7, avenue du Colonel Roche, 31077 Toulouse (France)

Abstract: The aim of this paper is to show how reliability trend analysis can help the designer in controlling the progress of the development activities and appreciating the efficiency of the test programs. Software reliability growth is first characterized and practical recommendations for trend analysis are discussed. Application of trend tests to some data sets collected on real systems illustrates the proposed method.

Introduction¹

Generally Software reliability studies are based on reliability growth models application in order to evaluate the reliability measures. When performed for a large base of deployed software systems, the results are usually of high relevance (see e.g. [Ada 84, Kan 87] for examples of such studies). However, utilization of reliability growth models during early stages of development and validation is much less convincing: when the observed times to failure are of the order of magnitude of minutes or hours, the predictions performed from such data can hardly predict mean times to failure different from minutes or hours ... which is so distant of any expected reasonable reliability as is not very helpful to perform such estimations. In addition, when a program under validation becomes reliable enough, the times to failure may simply be large enough in order to make the application of reliability growth models impractical, due to the (hoped for) scarcity of failure data. On the other hand, in order to become a true engineering exercise, software validation should be guided by quantified considerations relating to its reliability. Statistical trend tests provide for such guides.

This paper is devoted to the presentation of trend tests which are intended to help the management of the development and validation process. It will be shown that, for several circumstances, trend tests give information of prime importance to the developer. Emphasis will be put on the way they can be used during project progress and on practical results that can be derived from their use.

The paper is composed of three sections. The first section is devoted to the characterization of reliability

growth. In the second section, trend tests are presented and discussed; the type of results which can be drawn from trend analysis are stated. The third section is devoted to exemplifying the results from the first and second section on failure data collected on real systems.

1. Reliability growth characterization

1.1. Practical considerations

Software lack of reliability stems from the presence of faults, and is manifested by failures which are consecutive to fault sensitization². Removing faults should result in reliability growth. However, it is not always so, due to the complexity of the relation between faults and failures, thus between faults and reliability, which has been noticed a long time ago (see e.g. [Lit 79]). Basically, complexity arises from a double uncertainty: the presence of faults, the fault sensitization via the trajectory in the input space of a program³. As a consequence, one usually observes reliability trend changes, which may result from a great variety of phenomena, such as:

- variation in the utilization environment: the variation in the testing effort during debugging, change in test sets, addition of new users during the operational life, etc.,

¹ * This work was supported in part by the ESPRIT BRA project 3092 "Predictably Dependable Computing Systems" (PDCS).

² Precise definition of faults, failures, reliability, etc. are given in [Lap 87].

³ As an example, data published in [Ada 84] concerning nine large software products show that for a program with a mean lifetime of fifteen years, only 5% of the faults will be activated during this period.

- dependency of faults: some software faults can be masked by others, i.e. they cannot be activated as long as the latter are not removed [Ohb 84],
- variation in time delay between the detection of an error and its removal; this is closely dependent on the nature of the activated faults: some faults are more difficult to identify than others and take longer time to be removed.

With this in mind, reliability decrease may not, and usually does not, mean that the software has more and more faults; it does just tell that the software exercises more and more failures per unit of time under the corresponding conditions of use. Corrections may reduce the failure input domain but more faults are activated or faults are activated more frequently. However, during fault correction new faults may be also introduced — regression faults — which can deteriorate or not software reliability depending on the conditions of use. Last but not least, reliability decrease may be consecutive to specification changes.

1.2. Formal definitions

From what precedes, it can be seen that software reliability may be characterized by means of two types of random variables: the inter-failure time or the number of failures per unit of time (i.e. the failure intensity). These two random variables are not independent: knowing the inter-failure times it is possible to obtain the failure intensity (the second form needs less precise data collection). Both of them are considered in our work. The choice between one variable or the other may be guided by the following elements: i) the objective of the reliability study (development follow up, maintenance planning or reliability evaluation), ii) the way data is collected and iii) the life cycle phase concerned by the study.

Let T_1, T_2, \dots denote the sequence of random variables corresponding to inter-failure times, and $F_{T_i}(x)$ the distribution function of T_i . Reliability growth is characterized by the fact that inter-failure times tend to become larger, i.e.:

$$T_j \leq_{st} T_i, \text{ for all } j < i \quad (1)$$

where \leq_{st} means stochastically smaller than; under the stochastic independency assumption this is equivalent to:

$$F_{T_j}(x) \geq F_{T_i}(x) \text{ for all } j < i \text{ and } x \quad (2)$$

Data collection in the form of inter-failure times may be tedious mainly during development, in which case it is more suitable and less time consuming to collect data in the form of number of failures per unit of time. The unit of time is function of the type of use of the system as well as the number of failures occurring during the considered units of time.

In this case, reliability growth is expressed by:

$$N(t_1) + N(t_2) \leq_{st} N(t_1 + t_2) \text{ for all } t_1, t_2 \geq 0 \quad (3)$$

where $N(t)$ is the number of observed failures during time interval $[0, t]$. Inequality (3) must be strict for at least a couple (t_1, t_2) and means that the expected number of failures in any initial interval $[0, t_2]$ is no smaller than the expected number of failures in any interval of the same length occurring later $[t_1, t_1 + t_2]$. Let $H(t)$ denote its expectation: $E[N(t)]$. Assuming that the failure process has independent increments leads to:

$$H(t_1) + H(t_2) \geq H(t_1 + t_2) \text{ for all } t_1, t_2 \geq 0 \quad (4)$$

When (4) holds, $H(t)$ is said to be a subadditive function [Hol 74]; when it is reversed $H(t)$ is said to be superadditive and denotes reliability decrease.

2. Reliability growth analysis

Reliability growth can be analysed through the use of trend tests: these tests give a better insights into the evolution of the reliability. Several trend tests may be employed for each type of random variable; due to space limitation, only the most used and significant ones are presented in this section⁴. The presentation of the tests is followed by a discussion on how they can be used for studying software reliability. Some types of results that can be drawn from trend analysis are discussed in the last sub-section.

2.1. Trend tests presentation

2.1.1. Inter-failure times

Among the existing tests, the arithmetical mean and the Laplace tests can be used. The first test consists of calculating τ_k the arithmetical mean of the first k observed inter-failure times θ_i (which are the realizations of $T_i, i = 1, 2, \dots, k$):

$$\tau_k = \frac{1}{k} \sum_{i=1}^k \theta_i \quad (5)$$

When τ_k form an increasing series, reliability growth is deduced. This test is very simple and is directly related to the observed data. It is a graphical test and as such is informal.

A more rigorous test consists of calculating the Laplace factor [Cox 66] for the observation period $t_0, u(t_0)$. The occurrence of the events is assumed to follow a non-homogeneous Poisson process (NHPP) whose failure intensity is decreasing and is given by:

$$h(t) = e^{a+bt} \quad b < 0 \quad (6)$$

If $b=0$ the Poisson process becomes homogeneous and the occurrence rate is time independent.

⁴ For a more complete presentation and discussion on trend tests see e.g. [Asc 84, Gau 90].

Under this hypothesis ($b=0$), the statistics:

$$u(t_0) = \frac{1}{n} \sum_{i=1}^n (s_i - t_0) \frac{1}{t_0} \ln \left(\frac{1 - F(1; 2n)}{1 - F(1; 2n)} \right) \quad (7)$$

(where n is the number of failures in $[0, t_0]$, and s_i the time of occurrence of failure i , $i=1, \dots, n$)

is approximately normal distributed with zero mean and unit variance. Negative value of $u(t_0)$ means that the considered statistics is below the mean and therefore indicates $b < 0$, i.e. a decreasing failure intensity. On the other hand, positive values suggest an increasing failure intensity.

2.1.2. Failure intensity

Two very simple graphical tests can be used (the plot giving the evolution of the observed cumulative number of failures or failure intensity versus time) as well as some analytical tests (among which the Laplace test and the superadditive test). These four tests are briefly presented and discussed hereafter.

Graphical tests

Figure 1 gives the observed cumulative number of failures and the corresponding trend: this trend is directly related to relation (3); a concave curve (down) indicates reliability growth (i.e. inequality (3) holds) and, in the converse, a convex curve results in reliability decrease. The observed number of failures per unit of time n_k (the failure intensity) corresponding to these situations is given in figure 2 ($N_k = \sum_{i=1}^k n_i$).

The Laplace test

Following the method outlined in [Cox 66], the Laplace factor is derived in [Kan 91] and is given by:

$$u(k) = \frac{1}{k} \sum_{i=1}^k (i-1)n_i - \frac{1}{k} \sum_{i=1}^k (i-1)n_i \quad (8)$$

As previously, negative values of $u(k)$ indicate reliability growth whereas positive values indicate reliability decrease.

The superadditive test

The Laplace test is well suited to test monotonic trend versus no trend; when the collected data do not exhibit monotonic trend, the superadditive test is more adapted as discussed in [Hol 74, Asc 84]. This test is based on relations (3) and (4): when these relations hold reliability growth is deduced.

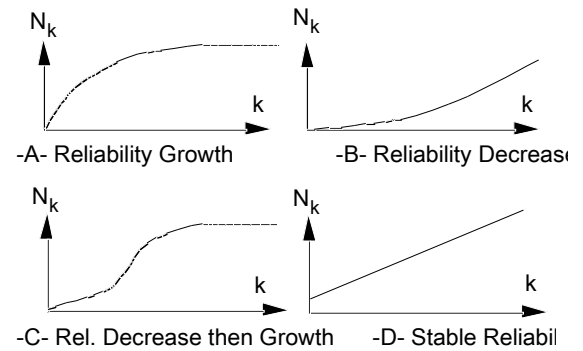


Figure 1: Cumulative number of failures and reliability evolution

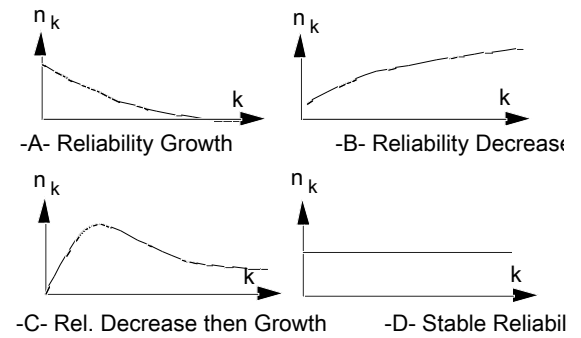


Figure 2: Failure intensity and reliability evolution

2.2. Discussion, practical recommendations

Comparison of several trend tests among which the Laplace test and the superadditive test according to optimality and/or consistency is carried out in [Hol 74, Asc 84, Gau 90]. The latter reference recommends the use of the Laplace tests in several situations mainly when processing raw data or under the NHPP assumption. However it is not well adapted to identify reliability fluctuation and even trend changes such as situation c of figures 1 and 2. Indeed it is well suited to test monotonic trend only. On the other hand, the superadditive test is more adapted to non-monotonic trend, but it is more difficult to be used in a systematic way. The Laplace test suffers from the fact that it is specific of a given model (expression (6)) and one has to associate a specific expression of this factor to each model.

For our purpose, we adopt a pragmatic point of view: we do not use the Laplace test as a statistics with confidence interval, but merely as an indicator to test the trend. Moreover the investigation of the evolution of this factor will help in detecting trend fluctuation. Actually, this factor can be evaluated step by step (at each unit of time or after each failure) and trend change of this factor indicates *local* trend change in the data. This is illustrated in figure 3-a: Considering for instance only data from the trend change point A leads to negative values (i.e. reliability growth) as indicated in figure 3-b. Periods of reliability growth and decrease can thus be identified.

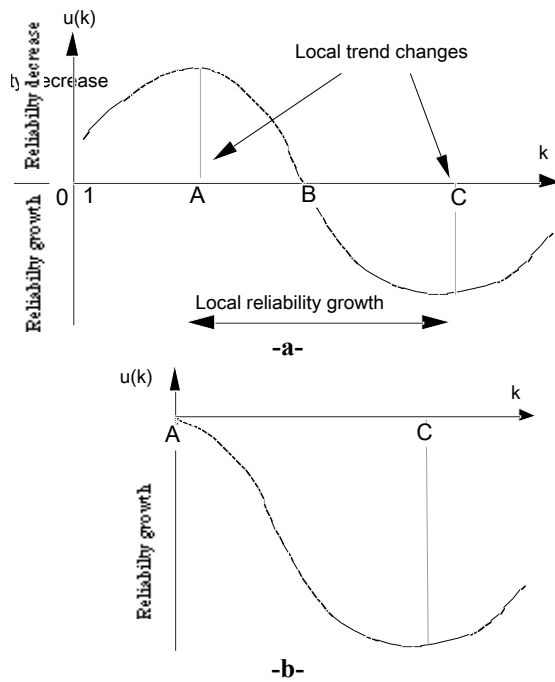


Figure 3: Laplace factor and local fluctuation

This phenomena is illustrated by figures 4 and 5 which relate to the TROPICO-R switching system studied in [Kan 91]. Figure 4 gives the Laplace factor for the whole data set from validation to operation. At the beginning of the validation, reliability decrease took place, as a result of the correction of 28 faults during the third unit of time whereas only 8 faults were removed during the first two time units and 24 during the next two time units; applying the trend test without the data belonging to the three first units of time leads to reliability growth (figure 5).

In real situations, we will use the Laplace test to analyse the trend considering the sign of its factor as well as the evolution of this factor with time. A Laplace factor oscillating around a constant value (within a bound of -2 and +2) implies stable reliability.

2.3. Results which can be issued from trend analysis

Trend analyses are of great help in appreciating the efficiency of test activities and controlling their progress. They help considerably the software development follow up. Indeed graphical tests are more often used in the industrial field [Gra 87, Lev 91, Val 88].

Reliability decrease at the beginning of a new activity such as i) new life cycle phase, ii) change in the test sets within the same phase, iii) adding of new users or iv) activating the system in a different profile of use, etc., is generally expected and is considered as a normal situation. Reliability decrease may also result from regression faults. Trend tests allow to detect this kind of behavior. If the duration of the period of decrease seems long, one has to pay attention and, in some situations, if it keeps decreasing

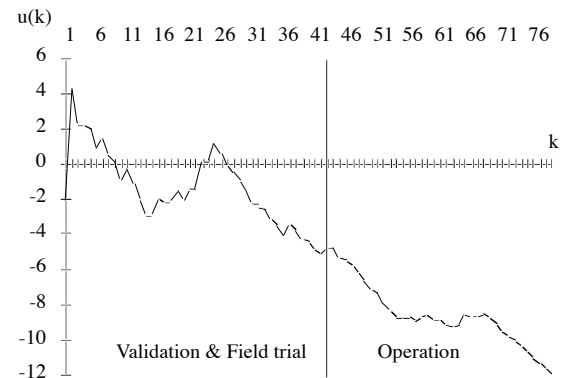


Figure 4: Laplace factor for the TROPICO-R considering the whole data set

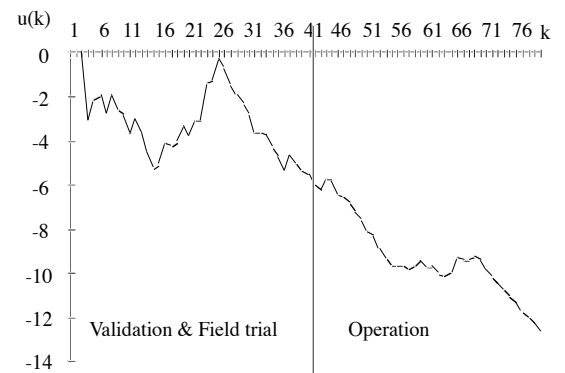


Figure 5: Laplace factor for TROPICO-R without considering the first three failure data

this can point out some problems within the software: the analysis of the reasons of this decrease as well as the nature of the activated faults is of prime importance in such situations. Such analysis may help in the decision to re-examine the corresponding piece of software.

Reliability growth after reliability decrease is usually welcomed since it indicates that, after first faults removal, the corresponding activity reveals less and less faults. When calendar time is used, mainly in operational life, sudden reliability growth may result from a period of time during which the system is less used or is not used at all; it may also result from the fact that some failures are not recorded. When such situation is noticed, one has to be very careful and, more important, an examination of the reasons of this sudden increase is essential.

Stable reliability indicates that the corresponding activity has reached a "saturation": application of the corresponding tests set does not reveal new faults, or the corrective actions performed are of no perceptible effect on reliability; one has either to stop testing or to introduce new sets of tests or to proceed to the next phase. More generally a test set has to continue to be applied as long as it exhibits reliability growth and stopped when stable reliability is reached.

Finally, it is noteworthy that trend analyses may be of great help for reliability growth models to give

better predictions since they can be applied to data displaying trend in accordance with their assumptions: failure data can be partitioned according to the trend and two types of reliability growth models can be applied: i) when the data exhibit reliability decrease followed by reliability growth, an S-Shaped model [Ohb 84] can be applied, ii) in case of reliability growth most of the other existing reliability growth models can be applied.

3. Application to real systems

Four different systems are considered:

- the first one, called **system A**, corresponds to a system which has been observed during validation and a part of operational life [Met 90],
- the second, to system 27 published in [Mus 79], called **system B** hereafter,
- the third one is also issued from [Mus 79], system SS4, called **system C**,
- the last one corresponds to the system considered in section 2 [Kan 91] and is called **system D**.

3.1. System A

The Laplace factor for this system is given in figure 6. System A displayed reliability decrease during the validation, reliability growth took place during operational life only. This is confirmed by figure 7 where the Laplace test is applied separately to each phase. It can also be seen that some reliability fluctuations took place from unit time 15, this fluctuation is due to introduction of new users.

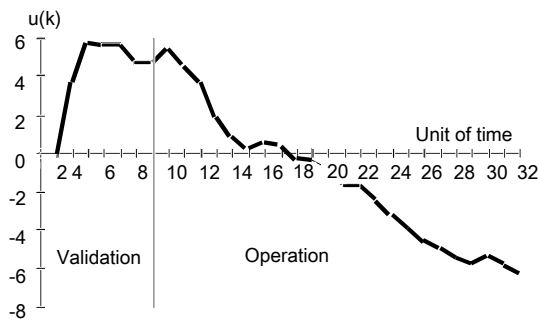


Figure 6: Laplace factor for System A considering the whole data set

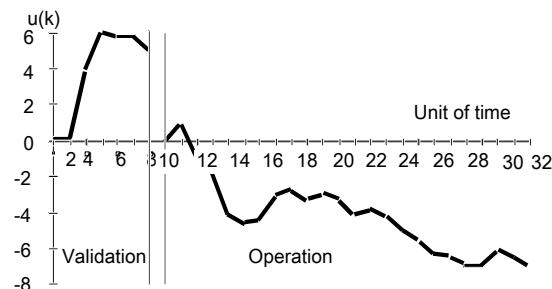


Figure 7: Laplace factor for System A considering each phase separately

Clearly, applying reliability growth models during validation would not have given helpful insights. An S-Shaped model can be applied to the whole data set and

any reliability growth model to operational data.

3.2. System B

System B is an example of systems which exhibit two phases of stable reliability; transition between them took place about failures 23-24 (figure 8). This system was under test and one has to know what happened at this time and why no reliability growth took place. It was not possible from the published data to identify the reasons of this behavior. In this case, data may be partitioned into two subsets each of them being modeled by a constant failure rate: the failure rate of the second subset (from 24 to 42 being lower than the failure rate of the first subset).

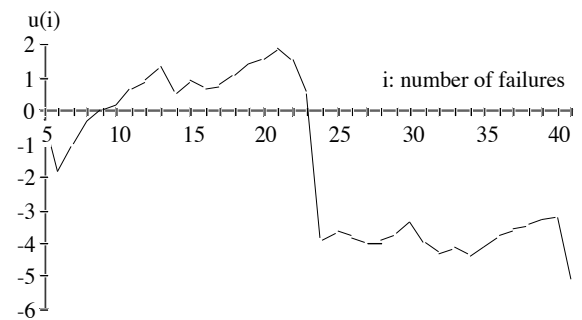


Figure 8: Laplace factor for System B

3.3. System C

Data gathered on System C correspond to operational life. Application of the arithmetical mean in figure 9 shows that the mean time to failure is almost constant: it is about $230 \cdot 10^3$. The corresponding Laplace factor oscillates between -2 and +2 indicating also stable reliability. In this case, a constant failure rate is well adapted to model the software behavior and is of simpler application than a reliability growth model. This result is not surprising since the software was not maintained (no fault correction).

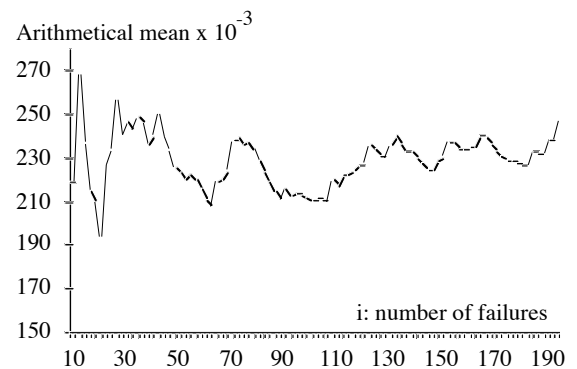


Figure 9: Arithmetical mean for System C

3.4. System D

Trend tests for this system are displayed in figures 4 and 5. Trend tests applied separately to each phase is illustrated in figure 10.

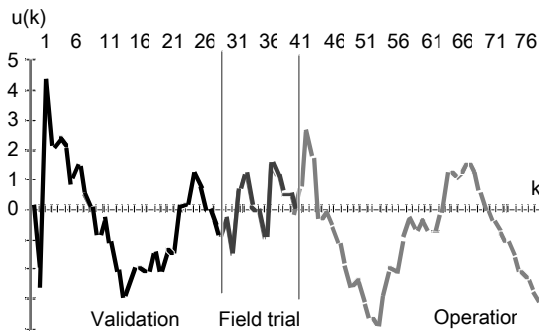


Figure 10 - Laplace factor of System D for each phase

It is interesting to comment figures 4 and 10 at the same time:

- the local reliability decrease from $k=14$ to $k=25$, was induced by the changes in nature of the tests within the validation phase: this period corresponds to the application of *quality* and *performance* tests after *functional* tests in the previous period; this decrease is due to their dynamic nature (traffic simulation) which activated new parts of the program,
- transitions from validation to field trial and from field trial to operation did not give rise to a reliability discontinuity, which means that the tests applied during the end of validation are representative of operational conditions,
- figure 4 indicates that from $k=55$ up to $k=70$ reliability tends to be stabilized: $u(k)$ is almost constant, suggesting a local reliability decrease; this behavior is reinforced when considering the trend results obtained for operational data only in figure 10 where reliability decrease is more evident; from $k=70$ this trend is reversed; this failure behavior is directly related to the number of installed exchanges over the periods considered, during which about twelve exchanges were installed and the number of failures reported by the users increased; by time unit 70, a new system had been released and no additional former system had been installed, which corresponds to the period of local reliability growth.

Applying reliability growth models blindly to this data set would have conducted to non significant results, on the other hand, using trend analysis results leads to trustworthy results [Kan 91].

Conclusion

Trend analyses constitute a major tool during the software development process. It has been shown how the results can guide the designer to control the progress of the development activities and even to take the decision to re-examine the software. Trend analyses are also of prime importance when reliability evaluation is needed. They allow periods of times exhibiting reliability growth and reliability decrease to be identified in order to apply reliability growth models to data exhibiting trend in accordance with their modeling assumptions. Trend tests

and reliability growth models are part of a global method for software reliability analysis and evaluation which is presented in [Kan 88] and has been applied successfully to data collected on real systems [Kan 87, Kan 91].

References

- Ada 84** E.N. Adams, "Optimizing Preventive Service of Software Products", *IBM J. of Research and Development*, vol. 28, no. 1, Jan. 1984, pp. 2-14.
- Asc 84** H.Ascher, H.Feingold, *Repairable Systems Reliability: Modeling, Inference, Misconceptions and Their Causes*, Lecture notes in statistics, Vol. 7, 1984.
- Cox 66** D.R.Cox, P.A.W.Lewis, *The Statistical Analysis of Series of Events*, London, Chapman & Hall, 1966.
- Hol 74** M.Hollander, F.Prochan, "A Test For Superadditivity For The Mean Value Function of a Non Homogeneous Poisson Process", *Stoch. Proc. and their applications*, vol. 2, 1974, pp. 195-209.
- Gau 90** O.Gaudoin, "Statistical Tools For Software Reliability Evaluation", Phd thesis, Joseph Fournier Univ. Grenoble I, Dec. 1990, in French.
- Gra 87** R.B.Grady, D.R.Caswell, *Software Metrics: Establishing a Company-wide Program*, Hewlett Packard Company, Prentice Hall, Inc., 1987.
- Kan 87** K.Kanoun, T.Sabourin, "Software Dependability of a Telephone Switching System", Proc. 17th *IEEE Int. Symp. on Fault-Tolerant Comp. (FTCS-17)*, Pittsburgh, Pennsylvania, July, 1987, pp. 236-241.
- Kan 88** K.Kanoun, J.C.Laprie, T.Sabourin, "A Method for Software Reliability Growth Analysis and Assessment", Proc. of *Software Engineering & its Applications*, Toulouse, France, Dec. 1988, pp. 859-878.
- Kan 91** K.Kanoun, M.Bastos Martini, J.Moreira De Souza, "A Method for Software Reliability Analysis and Prediction — Application to The TROPICO-R Switching System", *IEEE Transactions on Software Engineering*, April 1991, pp. 334-344.
- Lap 87** J.C.Laprie, "Dependability: a Unifying Concept for Reliable Computing and Fault Tolerance", *Resilient Computing Systems*, Vol. 2, T.Anderson Editor, Collins & Wiley, 1987.
- Lev 91** Y.Levendel, "Software Quality Improvement Process: When to Stop Testing", Proc. of *Software Engineering & its Applications*, Toulouse, France, Dec. 1991, pp. 729-749.
- Lit 79** B.Littlewood, "How To Measure Software Reliability and How Not To", *IEEE Trans.Reliability*, vol. R-28, no. 2, June 1979, pp. 103-110.
- Met 90** S.Metge, "Reliability Analysis and Evaluation of Two Telecommunications Software", LAAS research report n° 90.112, May 1990, in French.
- Mus 79** J.D.Musa, "Software Reliability Data", Data and Analysis Centre for Software Rome Air Development Centre (RADC) Rome, NY, 1979.
- Ohb 84** M.Ohba, S.Yamada, "S-Shaped Software Reliability Growth Models", Proc. 4th *International Conference on Reliability and Maintainability*, Perros Guirec, France, 1984, pp. 430-436.
- Val 88** V.Valette, "An Environment For Software Reliability Evaluation", Proc. of *Software Engineering & its Applications*, Toulouse, France, Dec. 1988, pp. 879-897.