Software dependability of a telephone switching system.

Karama Kanoun, Thierry Sabourin

SOFTWARE DEPENDABILITY OF A TELEPHONE SWITCHING SYSTEM

K.KANOUN

LAAS-CNRS
7, Avenue du Colonel Roche
31077 TOULOUSE Cedex

T.SABOURIN

ALCATEL COMMUTATION
B.P. 344
22304 LANNION Cedex

ABSTRACT

This paper presents a statistical analysis and a software reliability evaluation derived from data concerning a subsystem of a telephone switching system. These data were collected over three years (validation and operational life), from more than one thousand identical systems in several sites. The method adopted consists of analyzing the data, performing reliability growth tests and applying two types of models: reliability growth models taking into account the evolution of the different versions and a structural model concerning a particular version.

INTRODUCTION

The application of reliability growth models to a real system should always be preceded by the analysis of the system, its environment and the corresponding recorded failure data. This preliminary work is required in order to apply the models and interpret the results. Although each dependability evaluation has to be considered as a special case (Tro 86) due to the diversity of: (1) software and corresponding failure data, (2) development and validation methods, (3) collection of data, (4) model types and their application assumptions (in the absolute, if a model is not good, it may only be appropriate to certain data types (Kei 83)), a global approach can be followed. This approach consists of a preliminary study and of model applications.

The preliminary study has to include the following aspects:

- detailed analysis of the system in order to understand its operation and its fault-tolerance techniques,
- study of the software development and validation methods, and organization of data collection in order to choose the right models according to the plausible hypotheses,
- study of the system environment during validation and operational life in order to determine all the failure occurrence conditions and the possible consequences of the latter on the delivered service,
- statistical analysis of the failures and subsequent corrections of the latter, in order to establish, on the one hand, the relations between the software components, the failure consequences and the occurrence conditions of these failures, and, on the other, the relations between the failures and the activation rates of the components,
- preliminary data processing allows identification of doubtful data and testing of reliability growth; and hence enables a better insight into the process of software production.

Following these successive analyses, different reliability growth models may be applied. The results can then be easily interpreted and thus influencing the development and validation process. One or several models may thus be chosen to predict the software reliability during operational life.

The structural models taking into account the software components are then used to model the software behavior during its operational life.

This paper analyzes the software failures which have occurred on a subsystem of a telephone switching system. The follow-up of the subsystem was conducted from the validation phase up to a late period in its operational life. The first part presents the system and the available failure data. The second part covers a qualitative and then quantitative study of the failures and corresponding corrections. To do this, such failures are partitioned according to the different software components and the various consequences. In the third part reliability growth is tested, and a set of well-known reliability growth models are applied. The basis of a software reliability model is then verified in the fourth part; this model takes into account the components concerned, the failure consequence and the activation rates of the different components.

RELATED WORK

Numerous papers deal with the different steps of the afore mentioned dependability study. However, papers following a global approach are rare. A theoretical approach of the statistical study has been given in (Nag 82, Bas 84) and application examples are to be found in (Gla 81, Iye 82, Ros 82 or Tro 85).

Theoretical aspects of data processing are studied in (Aïv 70) or (Asc 78). So far, no application example has been published.

Definition and application of the reliability growth models have been discussed in numerous papers, (Mus 75, Goe 79, Lap 84, Kan 85 or Adb 86),

Applications to concrete cases are much fewer because results are often confidential. Software manufacturers and users are still unwilling to concede that there is no perfect software (zero fault) in operational life. However, there are interesting applications of reliability growth models in (Mus 79, Ada 80, Fav 85, Cur 86).

The dependability evaluation during the operational phase (using structural or reliability growth models) has been studied in (Lit 79, Lap 84b and Lap 86), but no result, or experiment has been published so far.

I) SYSTEM AND DATA COLLECTION DESCRIPTION

The system considered is part of a digital switching system and it fulfils telephone switching and operating tasks. It is a fault-tolerant system, comprising two similar units. An active unit performs the system functions, whereas the spare unit updates its own tables from those of the active unit.
Switching from one unit to the other is achieved either: (1) automatically, (2) by error detection (internal or external) in the active unit, (3) periodically, to reduce fault dormancy on the spare unit, (4) decided by the operator.
Hardware is similar in all units and software is not diversified.

System **validation** is achieved by an independent team, submitting the system to some tests. As soon as an error is detected and the fault diagnosed, the validation team fills in a Failure Report (FR). Once the fault has been removed by the development team, a Correction Report (CR) is established.

Following correction, the remaining information concerning the failure are: (1) FR, describing accurately the failure occurrence conditions and the consequences on the delivered service, (2) CR, stating the technical reasons of the failure, the component(s) concerned and the corrections performed.

At the beginning of the **operational life** errors are detected either by the validation team (still working on the system) or on the operation sites. In each case a FR and a CR are fulfilled.
The software faults discovered by the development team (still inspecting and testing the software), do not give rise to the creation of FR; the CR is systematically filled in. This is why the FR and CR numbers differ for the same period of time.

II) SOFTWARE FAILURE ANALYSIS

II.1) PRELIMINARY ANALYSES

The intervals between failures are issued from the FR set. These intervals do not correspond to lengths of time between failures but to intervals between writings of FR. The faults detected during validation do not lead to FR creation, but to that of CR. The intervals between corrections are taken from these CR sets: they enable to identify the components concerned and the types of faults.

Failure occurrence conditions

System activation falls under: (1) *telephone* solicitation, asked by the subscriber, (2) *operating* solicitation, i.e., all the operating and maintenance orders given by the operator.
The encountered occurrence conditions have been divided into three cases according to these various activation types:

- A: the failure occurred during the *telephone solicitation;* in this case there is neither a hardware failure in the system, nor unit switching, nor an operator order to execute,
- B: the software failure occurred after an *operator's order,* while the system was free from hardware failures and no switching occurred during the execution of the order,
- C: the failure is due to a *combination* of various events, it gathers all the cases of failure occurrence, except cases A and B, it includes all hardware failures activating a software fault.

Failure consequences

The failure consequences have been partitioned into four classes corresponding to the most frequent consequences:

- **GLobal UNAvailability** (GLUNA): the result of either an undetected failure of the active unit, or a serious failure leading to a complete reinitialization of the system,
- **PArtial UNAvailability** (PAUNA): it corresponds to the failures leading a) to the unavailability of one (or several) telephone circuit(s), internal connection(s) b) to the loss of some phone calls; but the telephony function is not completely lost,
- **LOss of one UNIt** (LOUNI): it corresponds to the failures leading to a unit switching, the system still operating normally,
- **ABandon of an OPErating command** (ABOPE): occurring when the operator's order is not executed.

For the last two classes the telephone function is not affected.

Software decomposition

A double software decomposition has been made, regarding:

- the **treatments** which fulfil elementary functions,
- the **groups** which consist of treatments according to the four main functions of the system. These groups are:
  - TELephony (**TEL**): all telephone switching routines,
  - OPErating (**OPE**): operating routines including initialization and out-line failure localization after detection of an error,
  - DEFense (**DEF**): covers all in-line fault-tolerance software (detection and recovery),
  - MONitor (**MON**): the executive software.

II.2) FAILURE ANALYSIS, MAIN RESULTS

58 FR and 136 CR were recorded over approximately three years. We will first analyze data from FR, then from CR and then conduct a combined analysis. Only the most important results are reported here, the whole analysis appearing in (Sab 86).

II.2.1) Failure Analysis (FR)

Figure 1 shows the proportions of the failure occurrence conditions and of the consequences.

Failure occurrence conditions

Case A	31 %
Case B	25 %
Case C	44 %

Failure consequences

GLUNA	13 %
PAUNA	24 %
LOUNI	30 %
ABOPE	33 %

Figure 1: Failure occurrence conditions and consequences

Occurrence conditions

It can be noticed that 44% of the faults occurred according to case C. This case gathers all the faults following a hardware failure or a combined hardware failure - operator's order or an inefficient unit switching. This proportion corresponds to malicious faults due to interactions between treatments. It can be seen that the occurrence conditions are rather well divided into the three considered cases.

Consequences

13% of failures lead to global unavailability. It is worth noting that these failures only concern a part of the telephone switching system. The corresponding faults were often introduced during the specification phase and are therefore difficult to solve. Indeed, most faults occur when the system is subject to heavy load. Besides, very few losses of phone calls are noticed, because these faults seldom lead to the FR creation.
The study of the evolution of the occurrence conditions and of the failure consequences, according to time, shows that the faults in case B are removed earlier than the others. On the other hand, the serious faults remain in the system for a rather long period. The simple coding faults are quickly corrected. The specification or design faults, occurring only in cases of overload or frequent solicitations, are discovered and corrected late.

II.2.2) Correction analysis (CR)

The faults are well distributed among the three groups: TEL, DEF and OPE (Figure 2). The MON group includes 15% of the corrections only, but it corresponds to the basic software and its utilities which are validated early.

	Volume	Faults
TEL	33 %	29 %
OPE	28 %	26 %
DEF	26 %	30 %
MON	13 %	15 %

Figure 2: Proportions of volume and faults per group

According to the different treatments, the number of faults per Kilo-Byte (KB) ranges from 0.9 to 4.6, implying a ratio of 1 to 5 between the best and the worst treatments. This may suggest that the number of discovered failures per component is generally proportional to the treatment volume. These results and those of the previous paragraph confirm those published in (Ros 82 and Kit 83). Figure 3 illustrates these results for the group TEL.

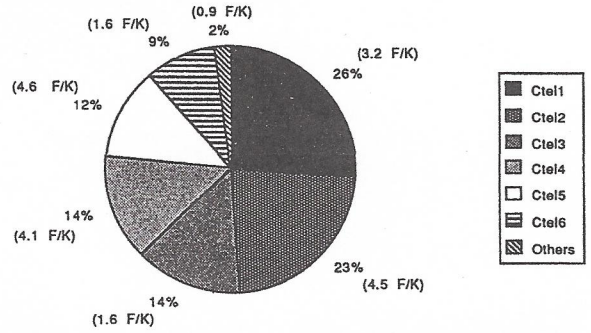

Figure3: fault partition for the group TEL, corresponding volumes

Figure 4 shows the evolution of the number of corrections per group. As foreseen, a slight delay in the validation of the operating software can be noticed. The residual failure rate in operational life is about $4\ 10^{-8}$/Hour.KB.

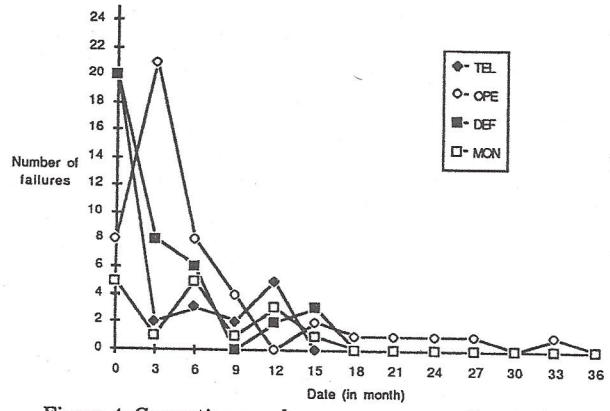

Figure 4: Correction number per group according to time

II.2.3) Combined FR and CR analysis

♦ **Correlations between treatments (or groups) and occurrence conditions:** The DEF software faults always occurred in case C. The **defense** does not seem to have any adverse effect in the absence of hardware failure. Besides, half the faults occurred after an operator's order and the others on telephone solicitation (case A), in the TEL and OPE software.

♦ **Correlations between fault consequences and occurrence conditions:** Most of the faults leading to global unavailability are due to hardware failures or coincidences during or after unit switching. The software component updating and reconfiguration after a unit switching must be carefully validated (prior to marketing the product).

♦ **Correlations between fault consequences and groups (or treatments):** An example of these correlations is given in Figure 5. The operator's orders activate faults which do not lead to service interruption. The validation of some parts of the operating software may thus be continued in parallel with the beginning of the operational life without disrupting too much the telephone service delivery.

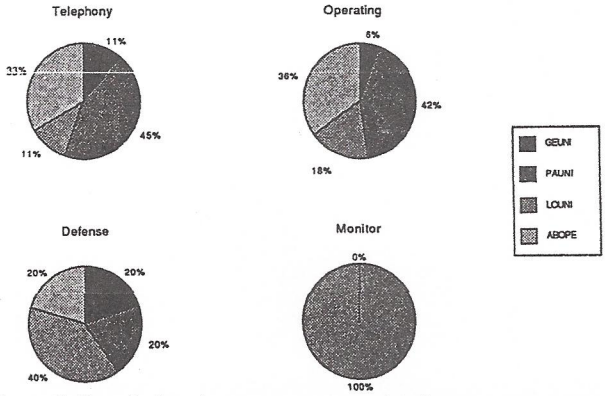

Figure 5: Correlations between groups and failure consequences

It can be noted that the MON faults give rise to losses of one unit (LOUNI) only, and thus do not lead to service interruption.

These last correlations enable to appreciate the fault-tolerance impact on the delivered service. Indeed, the DEF component, which is the main software component of the fault-tolerance implementation, has 30% of the faults. Among these faults, 20% led to global unavailability, i.e., 6% of the total number of faults; this corresponds to almost 50% of the global unavailability case (Figure 1). However, in many cases, the fault-tolerance allowed avoidance of global unavailability; 30% of the faults led to unit switching, but some of them were due to inadvertent orders and the available data did not permit us to assess this proportion.

II.3) CONCLUSIONS

It appears that the failures generally resulted from major system solicitations. These solicitations may originate from the operator, a hardware failure or a heavy telephone load. Besides, a time-lag equally occurs between the debugging of the different components, according to their activation rates.

The software number of identified and removed fault seems to be almost proportional to volume. During the three years of validation and operational life, 1 to 10 faults per KB (according to components) were observed, the average being about 5 faults per KB (for all the systems being validated and in operation).

The subsystem software failure rate equals $3\ 10^{-6}\ h^{-1}$.

The coding faults generally occurred at the beginning of the validation period. The persistent software faults are often due to design or specification and are more difficult to correct than other faults. These results are consistent with (Gla 81).

This qualitative analysis allows to establish a correspondence between the components concerned and the different failure consequences. The quantitative analysis, described in the next two chapters, consists of testing reliability growth, and then applying software reliability growth models.

III) RELIABILITY GROWTH STUDY

III.1) DATA USED

Standard software reliability models are based on the observation of time intervals between software failures. In our case these intervals will be taken from FR and CR. There will be two types of intervals: (1) between the writing of FR, (2) between the recorded corrections (CR).

The observation period covers the validation phase up to the operational life. These intervals are modified to take into account the increase in the number of systems put into operation.

III.2) RELIABILITY GROWTH TEST

Two complementary tests were performed: the arithmetic mean test (Kan 86) and the Laplace's test (Asc 78). Let t_1, t_2 ... t_n be the whole set of the observed intervals between failures.

The first test consists of calculating τ_k, the arithmetic mean of the first k data. When τ_k's form an increasing series, a reliability growth can be deduced. The second test is more formal and consists of calculating the Laplace indicator u. Reliability growth is characterized by a negative u. Figure 6 shows the results of the Laplace's test applied to the first M data.

F R

M	u
14	+0.58
44	+0.16
52	-1.12
58	-5.43

C R

M	u
30	-3.00
70	-4.39
110	-6.41
136	-15.5

Figure 6: Results of the Laplace's test

From the FR, different phases have to be distinguished:

- during a first phase, the system was under validation, there was no reliability growth; this can be explained by the fact that during the validation period new parts of the system are constantly tested,
- a second phase corresponding to the beginning of the operational life of the system; reliability growth is difficult to ascertain, because some faults, which could not be activated during a simulation phase, now occur,
- a third phase corresponding to the continuation of the operational life; reliability growth is clear and a very low failure rate is reached.

However the Laplace's indicators are negative and show a clear reliability growth on the whole data sets. The discontinuity

due to the transition validation-operational life can clearly be seen: the indicator u is positive for M<50 and negative for M>50. Thus, the transition validation-operational life must probably occur around the 50^{th} failure. It seems to correspond to reality.

III.3) RELIABILITY-GROWTH MODELS

The application of reliability models to a particular industrial case enables the system development to be controlled. It is then possible to: (1) choose a model fitting the considered data, (2) evaluate software reliability during validation, in order to manage validation delays and achieve target reliability, (3) evaluate the reliability according to the different components and to the various consequences, to ensure that the software meets the quality requirements. (4) predict software reliability in operational life.

III.3.1) The models

Three models are considered corresponding to the main types:

- the **MUSA** model (**MU**) (Mus 75),
- the **LITTLEWOOD-VERRAL** model (**LV**) (Lit 73),
- a model based on a Non Homogeneous Poisson Process (NHPP), the **HYPEREXPONENTIEL** model (**HE**)

The first two models are classical, the principles of the third one which is more recent (Lap 84b, Kan 85), are recalled below. The intensity function is based on a Cox Hyperexponential law:

$$h(t) = \frac{w_1 Z_1 e^{-Z_1 t} + w_2 Z_2 e^{-Z_2 t}}{w_1 e^{-Z_1 t} + w_2 e^{-Z_2 t}},$$

with $0 < w_1 < 1$, $0 < w_2 < 1$ and $w_1 + w_2 = 1$.

The corresponding hazard rate continuously decreases from an initial value $w_1Z_1 + w_2Z_2$ to a finite asymptote whose value is equal to $\inf(Z_1,Z_2)$. The MTTF which is the expected time to failure i, given that failure (i-1) occurred at time s, is:

$$MTTF = \frac{(w_1/Z_1) e^{-Z_1 s} + (w_2/Z_2) e^{-Z_2 s}}{w_1 e^{-Z_1 s} + w_2 e^{-Z_2 s}}.$$

III.3.2) Model application

The application of a model consists of estimating its parameters. This will be done using the Maximum Likelihood Estimation (MLE) method. Let $t_1, t_2, \ldots t_n$, be the successive intervals between observed failures. To test the appropriateness of a model, the first M_0 data are considered. The parameters of the model are estimated from these data. An estimation of the M_0+1^{th} MTTF: t^*_{Mo+1} is then deduced and compared with the observed value: t_{Mo+1}. The same sequence is repeated with M = M_0+1 to MT. At each step the following MTTF (t^*_{M+1}) is estimated. The quality of the estimation is evaluated by validation criteria such as the residue criterion or the KS criterion. The residue criterion calculates the relative difference between the observed and the estimated values:

$$RES = \sum_{i=1,n} (t_i - t^*_i) / \sum_{i=1,n} t_i.$$

The KS criterion (Cox 86) applies to the distribution functions of the random variables T_i (interval between failures). It calculates the KS distance (KSD) between the unknown distribution function of T_i and the distribution function of the applied model. This distance is issued from the u-plot in (Abd 86).

III.3.3) Application of the models, results

The three models are successively applied to the two data sets issued from the failures and the corrections reports.

The failures (FR)

The results of the three models concerning the FR data are gathered in columns 3, 4 and 5 of Figure 7.

	1 M	2 TM	3 MUSA	4 LV	5 HE	6 LV SBS	7 HE SBS
	9	8818.0	+++.+	127.9	5093.6	127.9	5093.6
	10	7596.7	+++.+	7174.5	5560.4	7174.5	5560.4
	11	13525.5	+++.+	9352.1	5786.8	9352.1	5786.8
S2	12	6497.4	7427.0	12228.2	6561.4	12228.2	6561.4
	13	723.0	7183.6	11371.8	6555.8	11371.8	6555.8
	14	724.1	+++.+	9766.5	6069.2	9766.5	6069.2
	15	9603.3	+++.+	7793.6	5657.7	7793.6	5657.7
	16	739.8	+++.+	9119.8	5940.4	4336.3	6783.8
	17	740.9	+++.+	7457.0	5592.8	980.4	6027.0
S3	18	9052.5	+++.+	4798.9	5289.3	-454.5	5440.1
	19	17137.8	+++.+	7312.6	5511.1	3644.7	5804.1
	20	21848.3	7354.3	9835.6	6156.9	7019.6	6832.6
	21	4073.9	11566.9	12104.4	6983.0	20396.1	15757.3
	22	815.9	9946.0	11460.1	6837.5	20202.0	12834.3
	23	4935.7	8246.7	10701.8	6549.7	19341.8	10429.8
	24	33783.7	8341.6	10718.2	6477.4	19339.8	9512.1
	25	867.3	14715.0	13561.5	7665.6	23962.8	12981.9
S4	26	16883.8	13038.9	13026.6	7382.0	23318.5	11465.4
	27	889.7	14090.7	13692.1	7762.4	24752.6	12061.0
	28	2679.3	11818.9	13204.0	7497.5	24063.1	10950.5
	29	894.2	10322.6	12517.3	7318.5	23367.1	10199.7
	30	2692.7	8930.6	11722.1	7089.6	11595.8	9423.6
	31	898.6	8050.3	10756.2	6937.5	6771.4	8904.4
	32	24122.2	7226.8	9304.1	6736.6	-226.2	6971.0
	33	928.9	9742.4	12391.6	7297.7	8427.7	8918.5
	34	4717.2	7561.9	9865.9	6900.7	4283.6	7369.0
	37	4745.1	7259.5	9821.2	6838.6	4135.5	7160.5
S5	38	2857.1	6991.6	9150.7	6780.4	4052.7	6983.9
	39	953.5	+++.+	8495.1	6674.3	3345.2	6701.5
	40	26560.2	+++.+	7335.2	6524.2	1881.6	6329.6
	41	4946.5	+++.+	10207.4	7037.8	7204.5	7546.9
	42	990.4	7870.1	10205.1	6985.4	6864.3	7400.7
	43	1985.3	7202.0	8991.3	6838.8	5875.0	7058.4
	44	993.7	+++.+	8240.8	6723.4	5050.4	6517.2
	46	23044.5	8303.8	9832.5	7045.0	11838.2	13564.4
	48	5332.4	10415.7	12554.2	7516.8	14971.5	13713.7
S6	49	7520.3	10054.8	11606.6	7469.5	14677.6	13141.1
	50	21934.0	10159.0	11483.5	7469.5	14621.2	12132.8
	51	3300.1	11410.9	12742.5	7765.8	16177.0	13104.3
	52	1101.1	10684.9	12024.9	7677.0	15625.5	12256.1
	53	108167.4	11426.1	11547.1	7547.7	14769.3	11240.9
	54	25732.4	26662.6	16615.0	9483.1	31707.9	66374.8
	55	16118.6	28222.9	17529.9	9790.7	32307.1	51221.3
S7	56	71578.3	33967.4	18281.6	9907.2	30812.7	42364.3
	57	3914.3	40851.4	20697.4	11028.9	39303.7	48314.9
	58	111542.1	52754.2	20244.3	10901.4	35627.8	40709.4
RESIDUE =			+0.161+	0.198	0.482	0.0581	0.0583
KSD =				0.180	0.187		

+++++ not applicable
SBS: Stage by Stage application

Figure 7: Model application on the 58 FR data set (time in days)

The first column gives the number of observed failures, the second the interval between failures, the third, fourth, and fifth give respectively the results of the MTTF estimations using the MU, LV, HE models. Then, the RES and the KS Distances values are indicated for the LV and HE models. For the MU model only one partial RES could be calculated on the data for which the model was applicable. This residue cannot be compared to the others, because the MU model is only applicable when the Laplace indicator is negative (reliability growth).

It can be noted in Figure 7, that during a long period of time there is no reliability growth. The LV and HE models easily follow a steady or a decreasing reliability, whereas the MU model does not. The LV model gives lower RES and KSD than the HE model and it is therefore more in keeping with the studied data. On the other hand the LV model follows the data so closely that it does not smoothe enough, as does the other model. Smoothing allows to obtain a more stable value of the estimated MTTF and to ignore local fluctuations. Besides, most of these fluctuations are due to slowing down of the validation activity. The HE model experiences great difficulties in following a strong and sudden

reliability growth. This is due to the influence of the first data corresponding to the validation phase. Finally, the results are relatively good because the KS distance is significant at 5% level for both LV and HE.

The Corrections (CR)

The results obtained are approximately similar to those obtained from FR. The LV and HE models remain pessimistic, whereas the MU model is very optimistic.

Progressive elimination of first data

This strong growth approximately begins after a break-point, corresponding to the transition from validation to operational life. This discontinuity brings about problems on the model applicability. That is why the first validation data were progressively eliminated, in order to reduce their restrictive and penalizing influence. To do this, the LV and HE models have been applied by stages. Seven stages were defined during the observation period: (S_1 ... S_7) including (2, 2, 2, 3, 3, 6, 6) months respectively. The S_1 stage data were used for the first initialization during the S_2 stage estimation. Similarly the S_i data were used for the initialization during the model application on S_{i+1} stage, whereas the S_1 ... S_{i-1} stages were deleted. The results of the stage application of the LV and HE models are gathered in columns 6 and 7 of Figure 7. The estimations fit better the observations and the residues are clearly smaller: 4 times less for the LV model, 10 times for the HE model. The MTTF are more significant and give a better evaluation of reliability in operational life at the end of the observation period.

III.3.4) Conclusions, encountered problems

Generally the application of any model is difficult to achieve. It is time consuming and the initialization parameters are often difficult or even impossible to find. LV and HE models are pessimistic, the MU model is optimistic, but the results do not differ greatly. It can therefore be stated that the models are almost similar, and the choice of a model will be based on other criteria such as: the failure rate evolution when time tends to infinity, the smoothing property, or the ease of use.

The HE model is very easy to use and smoothes data well. On the other hand, the dynamic LV model fits data well, but estimates negative or null MTTF in case of sudden decrease in the observed data. This is apparently due to the basic hypotheses of these models: the LV model assumes software to follow the cycle (failure, correction) during the validation, whereas the HE model assumes the cycle (several failures, corrections). This last assumption is more in keeping with reality and thus fits better the variations of observed data. The HE model remains very pessimistic, but by progressively eliminating the first validation data this defect is bypassed. Besides, the HE model tends toward an exponential model when t tends to infinity, whereas the two other models assume the MTTF to become infinite (null failure rate). Consequently the HE model could be used in operational life and will be used in the sequel.

IV) BASES OF A RELIABILITY MODEL IN OPERATIONAL LIFE

First, the software reliability structural model is described. This model addresses the software in the stabilized operational phase. Then the appropriateness of this model to the collected data is verified.

IV.1) THE STRUCTURAL MODEL

The detailed analysis is presented in (Lit 79, Lap 84a, Lap 84b). Consider a n-component software. Each component is failure-prone and its failure process is assumed to be a Poisson process of parameter λ_{pi}, $1 \le i \le n$. Exchanges of control take place among the components in such a way that at any time one and only one component is active. Let π_i be the average proportion of time spent in component i. The resulting software failure rate is:

$$\lambda_{eq} = \sum_{i=1,n} \pi_i \lambda_{pi} = \sum_{i=1,n} \lambda_{pi}, \qquad (1)$$

where λ_{pi} is the equivalent, apparent failure rate of component i.

It is worth noting that λ_{pi} takes into account the possible failures during the transfer of control between component i and the other components. It can be noticed that these formulas are verified if the component concerned is a treatment or a group.

For our purpose we extend this model to take account of the failure consequences in the following manner. Let λ_{cj} be the apparent failure rate of the whole software according to consequence j. We have:

$$\lambda_{eq} = \sum_{j=1,Nc} \lambda_{cj}, \qquad (2)$$

where Nc is the number of defined consequences.

IV.2) DATA PARTITIONS

Two decompositions will be considered:

- decomposition into groups, the decomposition into treatments will not be considered here owing to the limited amount of data concerning some treatments,
- decomposition into four subsets according to the consequences: GLUNA, PAUNA, LOUNI and ABOPE.

This choice was made in order to reach a tradeoff between the number of components, their volume and the available information about each components. With this choice, only formulas (1) and (2) can be verified.

IV.3) COMPONENT FAILURE RATES ESTIMATION

In order to verify these formulas we have to evaluate the apparent failure rates for the whole data sets and for all subsets. These rates are estimated using the HE model which is successively applied to:

- the four data sets, according to the first decomposition (into components) and to the whole CR set,
- the four data sets, according to the second decomposition (according to consequences) and to the whole FR set.

Results

Figs 8 and 9 give for each component and for each consequence:

- MT = total number of data for the corresponding set or subset,
- $\lambda_s = \inf(Z_1, Z_2)$, the stationary failure rate estimated by the HE model, λ_{si} is an estimate of the apparent failure rate (λ_{pi}, λ_{ci}),
- RES = residue calculated on the data sets,
- KSD = corresponding Kolmogorov-Smirnov Distance.

COMPON.	MT	λ_s (10^{-7} h^{-1})	RES	KSD
TEL	41	7.522	0.70	0.33
DEF	38	27.376	0.58	0.25
OPE	48	7.327	0.75	0.28
MON	16	8.331	-0.06	0.17
C.R.*	136	47.483	0.84	0.42

* All data from Corrections Reports.

Figure 8: Results according to components

CONSEQ.	MT	λ_s (10^{-7} h^{-1})	RES	KSD
GLUNA	7	1.206	0.16	0.53
PAUNA	18	7.946	0.72	0.45
LOUNI	16	3.126	0.48	0.22
ABOPE	18	3.650	0.42	0.23
F.R.*	58	38.223	0.52	0.20

* Data from FR.

Figure 9: results according to consequences

It appears that the sum of the data numbers (MT) of the four subsets (TEL, DEF, OPE, MON) (143) is greater than the total of collected data (136), because some failures needed corrections in several treatments.

Comments

Let us begin with the component decomposition. Using (2) and results of Figure 8, the global failure rate is:

$$\sum_{i=1,4} \lambda_{si} = 50.556\ 10^{-7}\ h^{-1}.$$

The estimation of the failure rate on the whole data set (CR) gives:

$$\lambda_{CR} = 47.483\ 10^{-7}\ h^{-1} \# \sum_{i=1,4} \lambda_{si}.$$

Let us consider now the consequence decomposition:

$$\sum_{j=1,4} \lambda_{sj} = 15.928\ 10^{-7}\ h^{-1}.$$

The estimation of the failure rate on the whole data set (FR) gives:

$$\lambda_{FR} = 38.223\ 10^{-7}\ h^{-1} \# \sum_{j=1,4} \lambda_{sj}.$$

These results show that the application of reliability-growth models in order to calculate failure rates of components is consistent with the structural model. It can also be seen that λ_{CR} is greater than λ_{FR} but with the same order of magnitude. This can be accounted for by the fact that corrections are always recorded, whereas failures do not necessarily lead to the creation of a FR.

It can also be noticed that the failure rates obtained a) directly without application of any model (cf II.3) b) from the HE model are of the same order of magnitude. The difference between all these failure rates does not seem to hinder a dependability evaluation. Finally it can be noted that the structural model seems capable of following the system behavior when using the apparent failure rates λ_{pi} and λ_{ci}.

The results are therefore encouraging but must still be refined using π_i and λ_{pi}which can be evaluated:

- π_i according to the functional characteristics of the software, i.e., the dynamic behavior corresponding to the interactions between components,
- λ_{pi} according to the characteristics of each component, volume, programming language...

CONCLUSIONS

This study has been conducted on the information concerning the failures of a telephone switching system, which were observed over a long period of time including the validation and operational phases. The preliminary analysis of data furnished interesting results which led to a better understanding of the software behavior.

Reliability growth tests (arithmetical mean and Laplace tests) have been performed. There was no reliability growth during the validation phase and at the beginning of the operational life, then a great reliability growth was noticed. The application of these tests has highlighted the transition between the validation and operational phases.

Reliability growth models have been applied to subsets of data according to the different components and the failure consequences and in eliminating progressively the first validation data.

The structural model has been applied during the operational phase. This first attempt to validate this model, using data from a real system, seems to be convincing and must be continued. It is the first step in the validation of the X-ware reliability theory (Lap 86).

This study has yielded many results and has highlighted the need for a systematic, complete data collecting mechanisms. This led to the definition of the organization of data gathering (data being related to software reliability (Sab 86)).

ACKNOWLEDGEMENTS

The authors wish to thank: J.C.Laprie (LAAS) for his encouragement and inciting remarks during the elaboration of this work, G.Fiche and F.Le Corre (ALCATEL COMMUTATION) for their constructive suggestions and criticisms, A.Costes (LAAS) for his helpful comments when reading earlier versions of the paper.

REFERENCES

Ada 80 E.N.Adams: "Minimizing cost impact of software defects", IBM Research Division. Rep. RC 8228 (35669), April 11, 1980.

Abd 86 A.Abdel-Ghaly, P.Y.Chan, B.Littlewood: "Evaluation of Competing Software Reliability Predictions", IEEE Trans. on Soft. Eng., vol SE-12 n°9, Sept. 1986.

Aïv 70 S.Aïvazian: "Etude Statistique des Dépendances", Editions MIR Moscou 1970 .

Asc 78 H.Ascher, H.Feingold: "Application of Laplace's Test to Repairable System Reliability", 1er Colloque international sur la fiabilité et la maintenabilité, Paris, 19-23 Jun. 1978.

Bas 84 V.R.Basili, D.M.Weiss: "Methodology for Collecting Valid Software Engineering Data", IEEE Trans. on Soft. Eng. vol. SE-10 n°6, Nov. 84, pp 728-738.

Cox 66 D.R.Cox, P.A.W.Lewis : The Statistical Analysis of Series of Events, London: Chapman & Hall, 1966.

Cur 86 P.A.Currit, M.Dyer, H.D.Mills "Certifying the Reliability of Software", IEEE Trans. on Soft. Eng., vol. SE-12, n°1, Jan. 1986.

Fav 85 J.Favrot, C.Lamy, F.Michel: "Reliability Evaluation of Programs for Irradiated Nuclear Fuel Testing during the Qualification Phase", Tech. et Science Informatiques, vol. 4, n°2; 1985

Gla 81 R.L.Glass: "Persistent Software Errors", IEEE Trans. on Soft. Eng., vol 7, n°2, Mar 1981.

Goe 79 A.L.Goel, K.Okumoto: "Time-dependent Error-detection Rate Model for Software and other Performance Measures", IEEE Trans. on Rel., vol. R-28 n°3, Aug. 1979, pp 206-211.

Iye 82 R.K.Iyer, S.E.Butner, E.J.Mac Cluskey: "A Statistical Failure Load Relationship: Results of a Multicomputer Study" IEEE Trans. on Comp., vol C-31 n°5, 1982, pp 697-706.

Kan 85 K.Kanoun, J.C.Laprie : "Modeling Software Reliability and Availability from Development-validation up to Operation", Rapport de recherche LAAS n° 85-042, Mar. 1985, révision Aug 1985.

Kan 86 K.Kanoun: "Validation de Modèles Stochastiques, Application aux Modèles de Croissance de Fiabilité du Logiciel", Rap. de Rech. LAAS n° 86-002, Jan 86 (in French).

Kit 83 B.A.Kitchenham: "The Use of Software Metrics to Assess Software Production Methods", Proc. 13th Int. Symp. Fault-Tolerance Computing (FTCS 13), Milan, Italy, Jun. 28-30 1983, pp. 135-138.

Lap 84a J.C.Laprie : "Dependability Evaluation of Software Systems in Operation", IEEE Trans. on Soft. Eng. Vol. SE-10, No 6, Nov. 1984, p. 701-714.

Lap 84b J.C.Laprie: "Dependability Modeling and Evaluation of Software-and- Hardware Systems", Invited survey to the 2nd GI/NTG/GMR Conference on Fault-tolerant Computing, Bonn, Germany, Sep. 19-21, 1984.

Lap 86 J.C.Laprie: "Towards an X-ware reliability theory", LAAS Report n. 86-364, Dec. 1986.

Lit 79 B.Littlewood: "Software Reliability Model for Modular Program Structure" IEEE Trans. on Rel. vol 28, n°3 Aug. 1979.

Mus 75 J.D.Musa : "A theory of Software Reliability and its Application", IEEE Trans. on Soft. Eng., Vol. SE-1, Sept. 1975, pp. 312-327.

Mus 79 J.D.Musa: "Software Reliability Data" Data and analysis center for software, Rome Air Development Center (RADC) Rome, NY, 1979.

Nag 82 P.M.Nagel, J.A.Skrivan: "Software Reliability: Repetitive Run Experimentation and Modeling, BCS 98124, Boeing Computer Service Company, Seattle, Washington, Feb. 1982.

Ros 82 D.J.Rossetti, R.K.Iyer: "Software-related Failures on the IBM 3081: A Relationship with System Utilization, CRC Tech. rep. n°82.8 jun. 1982.

Sab 85 T.Sabourin, G.Fiche, F.Lecorre, K.Kanoun : "Evaluation de la Fiabilité et de la Disponibilité d'Autocommutateurs Téléphoniques, Etude des Anomalies" Rap. de rech. LAAS.85.099-ATD.85.331, Mai 85 (in French).

Sab 86 T.Sabourin, K.Kanoun : "Failure Analysis and Modelization of Software Behavior in an Electronic Switching System" 5ème Colloque international sur la fiabilité et la maintenabilité, Biarritz, 6-10 Oct. 1986 pp 92-97 (in French).

Tro 85 R.Troy, C.Baluteau: "Assessment of Software Quality for the Airbus A 310 Automatic Pilot" Proc. 15th Int. Symp. Fault-Tolerant Comp., Ann Arbor, Michigan, Jun. 1985, pp. 438-443.

Tro 86 R.Troy, Y.Romain: "A Statistical Methodology for the Study of the Software Failure Process and its Application to the ARGOS Center", IEEE Trans. on Soft. Eng. vol SE-12 n°9, Sep. 1986.