



Software Failure Data Analysis of two Successive Generations of a Switching System

Mohamed Kaâniche, Karama Kanoun

► To cite this version:

Mohamed Kaâniche, Karama Kanoun. Software Failure Data Analysis of two Successive Generations of a Switching System. 12th IFAC International Conference on Computer Safety, Reliability and Security (SAFECOMP'93), Oct 1993, Poznan, Poland. 10.1007/978-1-4471-2061-2_24 . hal-01212234

HAL Id: hal-01212234

<https://hal.science/hal-01212234>

Submitted on 7 Oct 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.

Software Failure Data Analysis of two Successive Generations of a Switching System

Mohamed Kaâniche and Karama Kanoun
LAAS-CNRS, 7 Avenue du Colonel Roche
31077 Toulouse Cedex — France

Abstract

Experimental studies dealing with the analysis of data collected on families of products are seldom reported. In this paper, we analyse the failure data of two successive products of a software switching system during validation and operation. A comparative analysis is done with respect to: i) the modifications performed on system components, ii) the distribution of failures and corrected faults in the components and the functions fulfilled by the system, and iii) the evolution of the failure intensity functions.

1 Introduction

Most current approaches to software reliability evaluations are based on data collected on a single generation of products. However, many applications, not to say the great majority, result from evolutions of existing software: there are families of products, the various generations resulting from evolutions for implementing new functionalities. A new approach that is aimed at the incorporation of past experience in predicting the reliability of a new, but similar, software has recently been proposed in [1]. This approach requires the identification of parameters which characterize past experience to be incorporated in the evaluation of the software reliability. Clearly, the identification of these parameters will be based on the analysis of data collected over the whole family of products. Experimental studies dealing with the analysis of families of products are seldom reported [2, 3]. The data considered in this paper were collected on the software of two successive generations of the Brazilian Electronic Switching System (ESS)—TROPICO. Throughout this paper, the two products will be identified as PRA and PRB. PRA was first developed and allows connection of 1500 subscribers. The processing capacity of the TROPICO system was subsequently increased with the release of PRB which allows the processing of up to 4096 calls. Many PRA software components have been reused for the development of PRB and additional components were developed.

The failure data collected on each one of these products have been considered respectively in [4] for PRA and [5] for PRB. While our previous work was mainly devoted to reliability analysis and evaluation, this paper is concerned with the qualitative as well as quantitative analysis of the failure data. Our objective is to do a comparative analysis of the two successive products based on the data collected during the end of validation and the beginning of operation. Emphasis will be put on the evolution of the software and the corresponding failures and corrected faults.

This paper is composed of five sections. Section 2 gives a general overview of the TROPICO switching system. It describes the main functions performed by the system and presents some statistics about the evolution of PRB with respect to PRA.

Section 3 describes the test environment and the failure data collected. Section 4 presents some of the results derived from the collected data. Finally, Section 5 outlines the main results obtained from the analysis of both products.

2 Software Description

2.1 General description

The TROPICO ESS software features a modular and distributed structure monitored by microprocessors. The software can be decomposed into two main parts, that is, the **applicative software** and the **executive software**.

Two categories of components can be distinguished in the TROPICO ESS software: i) **elementary implementation blocks (EIB)**, which fulfil elementary functions and ii) **groups** of elementary implementation blocks according to the main four functions of the system. These groups are:

- Telephony (TEL): call processing, charge-metering, etc.
- Defense (DEF): on-line testing, traffic measurement, error detection, etc.
- Interface (INT): communication with local devices (memories, terminals),...
- Management (MAN): trunk and subscriber signalling tasks, communication with external devices,...

2.2 Evolution of PRB with respect to PRA

The development of PRB started while PRA was under validation. Many PRA components have been reused for the development of PRB and additional ones were developed. Three types of EIBs can be distinguished:

- *new*: specifically developed for PRB;
- *modified*: developed for PRA and modified to meet the requirements of PRB;
- *unchanged*: corresponding to PRA EIBs included in PRB without modification.

Figure 1 gives the number of EIBs and the size of the software for PRA and PRB. The software of PRA and PRB was coded in Assembly language. A 10 percent increase of the PRB size can be noticed relative to PRA. Only 4 new EIBs were developed for PRB. All the EIBs of PRA have been reused with or without modifications for PRB.

	#EIB	size (kbytes)
PRA	29	319.416
PRB	32	350.800

Figure 1: Number of EIBs and size of PRA and PRB

Figure 2 shows the amount of modification performed on PRB with respect to the number of EIBs and to the size of the software. About 67% of PRB code results from the modification of the PRA code. About 75% of the modified EIB's belong to the applicative software and 84% of unchanged EIB's to the executive. Thus, the increase of the TROPICO capacity mainly led to major modifications of the applicative software with only minor modifications of the executive.

When considering the four functions and the distribution of the three types of EIB of PRB, we notice that most of the unchanged modules belong to INT (about 60%).

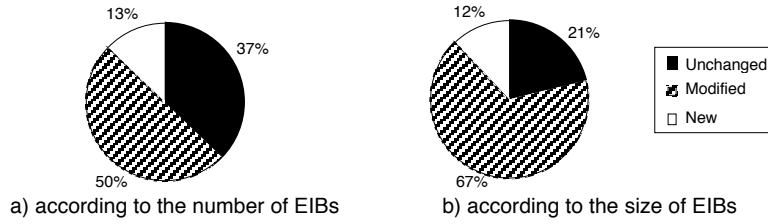


Figure 2: Distribution of unchanged, modified and new EIBs in PRB

3 Test Environment and Failure Data

3.1 Test Program

The software test program for TROPICO consists of four steps: 1) unit test, 2) integration test, 3) validation test, and 4) field trial test. The first three steps correspond to the test phases usually defined for the software life cycle. Field trial consists of testing a prototype in a real environment, which is similar to the operational environment. It uses a system configuration (hardware and software) that has reached an acceptable level of quality after completing the laboratory tests.

The description of the whole quality control program for TROPICO is given in [6, 7]. The test program carried out during validation and field trial test is decomposed into four kinds of test (functional, quality, performance and overload tests). PRA and PRB validation were carried out according to this program. Figure 3 shows, for the period of data collection on PRA and PRB, the length of validation in months, field trial and operation phases. As can be seen, no field trial tests were performed for PRB. This is because many PRA components were reused for the development of PRB, and PRB was put in operation while PRA had already been operating for several months.

	validation	field trial	operation
PRA	10	4	13
PRB	8	0	24

Figure 3: Validation, field trial and operation length for the period of data collection (months)

During the operational phase, the number of PRAs and PRBs installed on operational sites was progressively increased (see Figure 4). At the end of the data collection period, up to 15 PRAs and 42 PRBs had been installed.

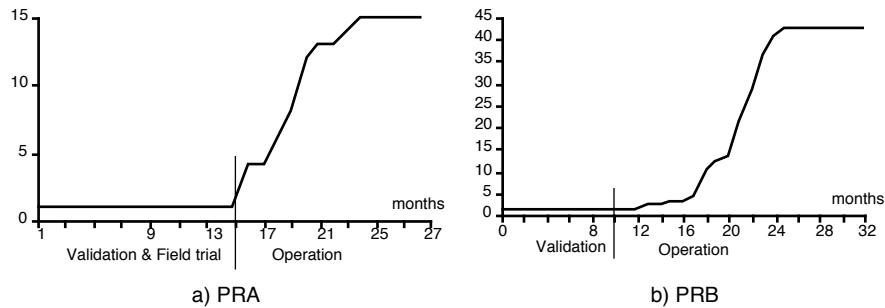


Figure 4: Number of installed sites versus time

3.2 Data Collection

Handling of failure data affecting the TROPICO ESS is through use of an appropriate failure report (FR) sheet containing the following:

- date of failure occurrence;
- origin of failure: description of system configuration in which the failure was observed and of the conditions of failure occurrence;
- type of FR: hardware, software, documentation with indication of affected elementary implementation blocks;
- analysis: identification and classification of the fault(s) which led to failure (coding, specification, interface,...);
- solutions: the proposed solutions and those retained;
- modification control: control of the corrected elementary implementation blocks;
- regression testing: results of the tests applied to the corrected elementary implementation block(s).

Only one FR is kept per observed failure: rediscoveries are not recorded. In other words, if several FR's cover the same failure, only one (the first) is entered into the database. In fact, an FR is both a failure report and a correction report since it also contains information on the fault(s) that resulted in failure.

The results presented in the following sections are based on the analysis of the data collected on the observed failures and on the corrections performed.

4 Relationships derived from the data

This section presents and discusses some of the results obtained from the data.

4.1 Statistics on failures and corrected faults in PRA and PRB

Figure 5 gives the number of failures and corrected faults in PRA and PRB. It can be seen that less failures occurred in PRB even though: i) the period of data collection for PRB is longer than that of PRA (see Figure 3) and ii) a greater number of systems have been in use during the operation phase (see Figure 4). Furthermore, the number of corrected faults exceed the number of failures. This is due to fact that some failures led to the modification of more than one EIB.

	# FR	# corrected faults
PRA	465	637
PRB	210	289

Figure 5: number of failures and corrected faults in PRA and PRB

Figure 6 shows the statistics concerning the number of EIBs that have been corrected because of a software failure. As can be seen, the results for PRA and PRB are similar. For both products about 80% of the failures led to the correction of only one EIB. This is really in favor of software modularity and equally shows that there is little failure interdependence among EIBs.

The analysis of the data corresponding to failures involving more than one component allowed us to identify two pairs of EIBs that are strongly dependent with respect to failure occurrence. For these two pairs, we noticed that the probability of simultaneous modification of both EIBs given that a failure was due to a fault located in one of them, exceeds 0.5. This result was obtained for both PRA and PRB. This type of analysis can be of a great help for software maintenance. It

allows software debuggers to identify the stochastically dependent components and to take into account this information when looking for the origin of failures.

# corrected EIBs	# FR in PRA	# FR in PRB
1	362 (77.8%)	165 (78.6%)
2	72 (15.5%)	33 (15.7%)
≥ 3	31 (6.5%)	12 (5.7%)

Figure 6: Statistics on the number of EIBs affected by a failure

4.2 Distribution of failures and corrected faults per functions

Figure 7 gives the number of failures and corrected faults attributed to the four functions: TEL, DEF, INT and MAN (as defined in Section 2.1). The sum of failure reports attributed to the functions is higher than the total number of failure reports indicated in Figure 5: this is because when a failure is due to the activation of faults in different functions, an FR is attributed to each of them.

	# FR	# corrected faults
TEL	146	190
DEF	138	164
INT	170	191
MAN	78	92
Sum	532	637

a) PRA

	# FR	# corrected faults
TEL	74	102
DEF	67	71
INT	61	68
MAN	31	41
Sum	233	282

b) PRB

Figure 7: Failure reports and corrected faults in TEL, DEF, INT and MAN

When looking at the distribution of corrected faults per functions (Figure 8), we obtain similar figures for both products, in particular DEF and INT. It can be seen that most of the corrections were performed in TEL and INT. This can be explained by the fact that these functions are more activated than DEF and MAN.

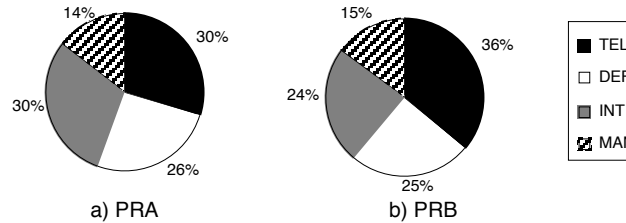


Figure 8: Fault distribution in TEL, DEF, INT and MAN

Furthermore, most of the failures reported led to the modification of only one function (90 %). Among the 465 FRs (resp. 210 FRs) recorded for PRA (resp. PRB), only 54 FRs: 31 during validation, 10 during field tests and 13 during operation (resp. 21 FRs: 10 during validation and 11 during operation) led to the modification of more than one function. This shows that the functions are not totally independent with respect to failure occurrence, although, only a weak dependence was observed. Note that this result, compared to those reported in Section 4.1, shows that less dependence is observed between functions than between EIBs.

4.3 Distribution of PRB faults per EIB type

Figure 9 shows the distribution of corrected faults in PRB when considering the unchanged, modified and new EIBs. Thus more than 80 percent of corrected faults were attributed to modified EIBs. It is noteworthy that almost the same distribution was obtained when considering data from validation or from operation only.

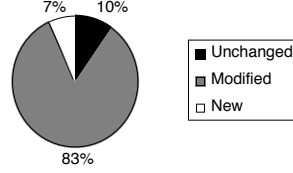


Figure 9: Distribution of FR per type

When reviewing the mean values of fault density (number of faults per Kbyte)* in the three types of EIBs, we obtain the following figures: 0.95 for modified EIBs, 0.75 for new ones and 0.49 for unchanged ones. One may think that the modified EIBs are more error prone than the new and unchanged ones, and would conclude that it is better to create new components than to modify already existing ones. However, we should be careful when analysing this type of result. In fact, as only 4 new EIBs were developed for PRB, no significant conclusions with respect to this particular point could be derived from this analysis.

An analysis of the average values of fault density presented in Figure 10 shows a significant decrease of the fault density of PRB EIBs when compared to PRA. Also, it can be seen that the fault density of all unchanged and modified PRB EIBs significantly decreased when compared to the values computed for PRA. This indicates an enhancement of the quality of the software. The experience cumulated during the validation and operational use of PRA led to a better understanding of the system and contributed to the improvement of the quality of PRB code.

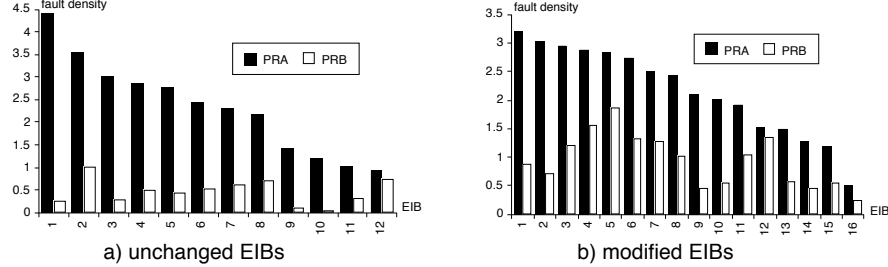


Figure 10: Fault densities of PRB EIBs compared to PRA

4.4 EIB size and fault density in PRA and PRB

Scatter plots of fault density per EIB (number of faults per Kbyte) versus the size of the EIB were plotted for PRA and PRB. It was difficult to ascertain any trend within these plots. Our objective was to analyse a possible significant dependence between the EIB fault density and their size.

* Note that the fault density as defined here is different from the commonly used one (i.e., number of faults per kilo lines of code); the latter is not available for this application.

Figure 11 gives for PRA and PRB the fault density average values for three categories of EIB size. The fault density is almost constant, it is around 2 faults per Kbyte for PRA and 1 fault per Kbyte for PRB. This illustrates the improvement of the quality of PRB code with respect to PRA and thus confirms the results reported in Section 4.3. As the size of PRA and PRB EIBs is measured in Kbytes and not in kilo lines of code, it is difficult to compare these values to other fault density values obtained which are reported for instance in [8, 9].

Size	PRA	PRB
EIB size > 15 Kb	1.80	1.08
10 Kb <EIB size< 15 Kb	2.02	0.68
5 Kb <EIB size< 10 Kb	2.31	0.60
EIB size< 5 Kb	2.56	0.71

Figure 11: Average values of PRA & PRB fault density versus EIB size

Figure 12 shows that the PRB modified EIBs exhibit higher fault densities on average than unchanged EIBs. However, it should be noticed that the number of EIBs in each category of size is small. Also, it can be seen that most of unchanged EIBs have a small size (less than 5 kbytes) compared to modified EIBs.

Size	modified	unchanged
EIB size > 15 Kb	1.25 (6 EIBs)	0.06 (1 EIB)
10 Kb <EIB size< 15 Kb	0.86 (5 EIBs)	0.75 (1 EIB)
5 Kb <EIB size< 10 Kb	0.8 (4 EIBs)	0.4 (4 EIBs)
EIB size< 5 Kb	0.26 (1 EIB)	0.55 (6 EIBs)

Figure 12: Fault density average values of modified and unchanged PRB EIBs versus size

4.5 Evolution of failure occurrences with respect to time

Figure 13 shows the evolution of the failure intensities of PRA and PRB during the period of data collection: for both products—even though the failure intensity is globally decreasing during the operational phase—the trend is not monotone. The local variations observed are due to the progressive installation of new systems (see Figure 4). It is noteworthy that the impact of the number of operational systems on the evolution of the failure intensity has been reported in several papers, see for instance [10, 11].

In order to evaluate the reliability of PRA and PRB as usually perceived by the users, we need to consider the failure intensities corresponding to an average system (i.e., the failure intensity divided by the number of systems in use). Figure 14 shows the evolution of the failure intensities of PRA and PRB for an average system. It can be seen that the failure intensities of both products decreased globally during operation thus exhibiting reliability growth.

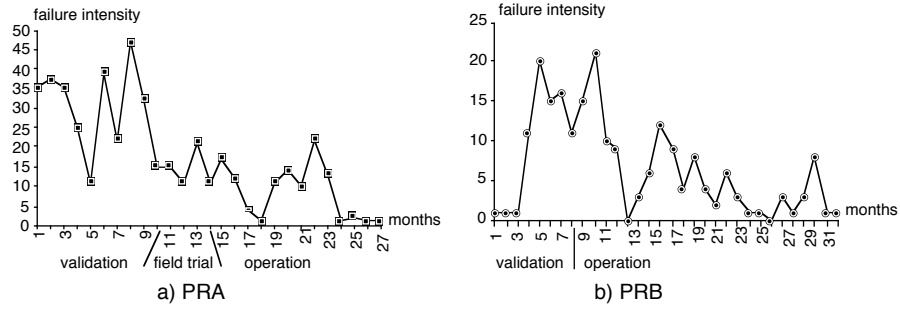


Figure 13: PRA and PRB failure intensities

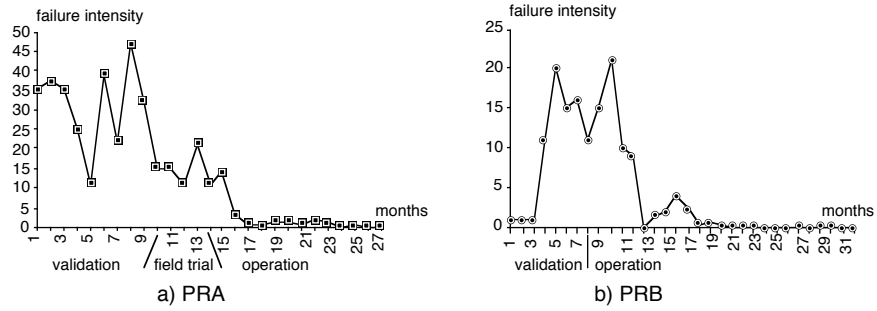


Figure 14: PRA and PRB failure intensities for an average system

In order to compare the reliability of PRA and PRB, we plot in the same figure (Figure 15) the failure intensities observed for an average system during operation. Unexpectedly, the reliability of PRB is worse than that of PRA. The same holds for the groups of functions TEL, DEF and INT (Figure 16). This is surprising because, as PRB has been developed from PRA which has been validated and extensively used one would anticipate that its reliability would be better than that of PRA. This may be explained by the fact that major modifications had been performed on PRA in order to adapt the system to the new specifications and no field trial test had been performed before the introduction of the system in the field. Note that about 80 % of PRB failures recorded during operation occurred during the first year of operation.

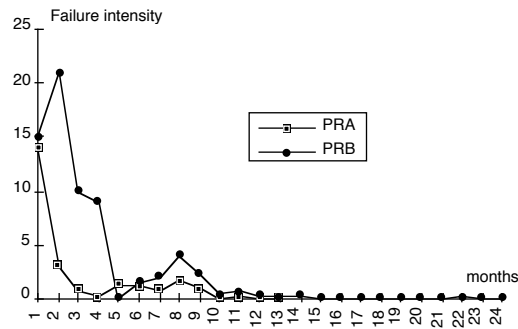


Figure 15: PRA and PRB failure intensities for an average system during operation

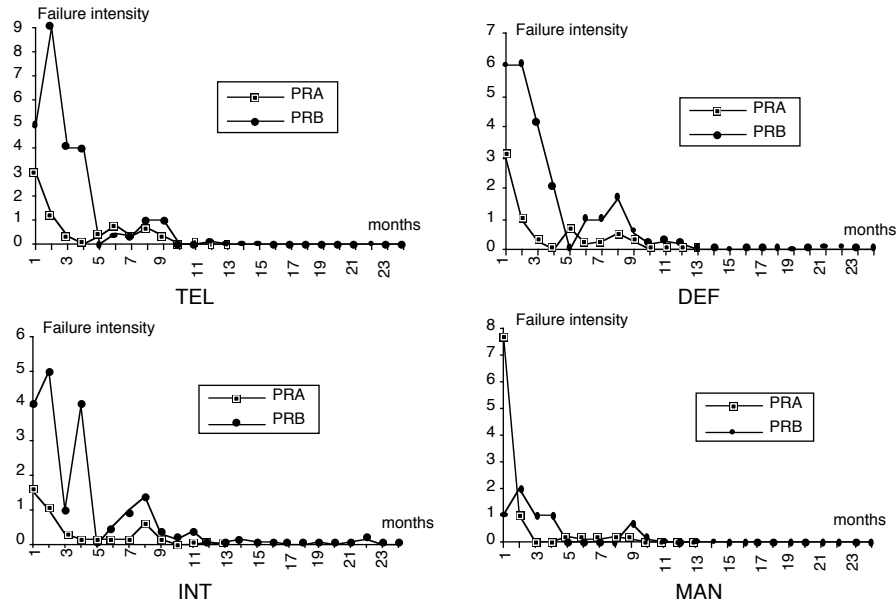
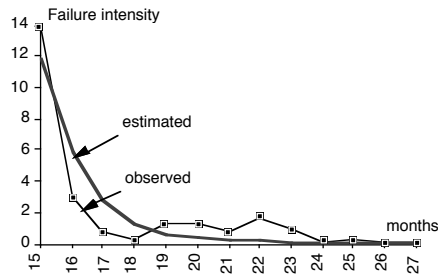


Figure 16: Component failure intensities for an average system during operation *

Typically, a new system experiences a maturing period during which its reliability is relatively low but afterwards, reliability keeps improving and becomes better than that of its predecessors. In fact, if we look at the long term evolution of the failure intensity functions of PRA and PRB (see Figure 17) it can be seen that the residual failure rate of PRB evaluated by the hyperexponential model [12] is less than the residual failure rate evaluated for PRA. Similar results are also obtained for TEL, DEF, and INT (Figure 18). For both products, the evaluations are based on the data collected during the last year of operation.

It is noteworthy that the same was noticed in [10] for successive releases of a wide-distribution software product and in [2] for three successive products of a family of ultra-available computers designed by AT&T Bell Laboratories.



* Note that for Figures 15 and 16, the X-axis indicates the number of months since the system was put in operation.

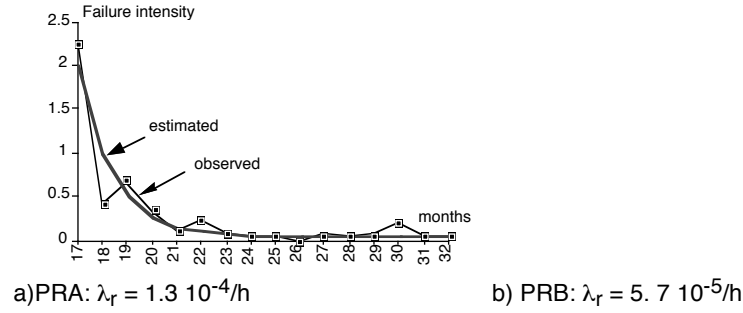


Figure 17: Estimation of PRA & PRB failure intensities

	PRA	PRB
TEL	$2.6 \cdot 10^{-5} /h$	$1.2 \cdot 10^{-6} /h$
DEF	$4.3 \cdot 10^{-5} /h$	$1.4 \cdot 10^{-5} /h$
INT	$4.2 \cdot 10^{-5} /h$	$2.9 \cdot 10^{-5} /h$
MAN	$1.4 \cdot 10^{-6} /h$	$8.5 \cdot 10^{-6} /h$

Figure 18: Residual failure rates evaluated by the hyperexponential model

5 Concluding Remarks

The data considered in this paper allowed us to analyse the evolution of the software and the failures of two consecutive products of the TROPICO ESS. The main results derived are as follows:

- A high percentage of failures was attributed to modified EIBs.
- For both products, about 80 % (resp. 90 %) of the failures led to the correction of only one EIB (resp. function). Therefore, only a weak dependence with respect to failure occurrence was observed between components.
- The fault density of PRA and PRB is almost constant with respect to size. It is about 2 faults per Kbyte for PRA and 1 fault per Kbyte for PRB.
- The fault density values of all modified and unchanged PRB EIBs are lower than those of PRA EIBs. This shows an improvement of the quality of PRB code with respect to PRA.
- Comparison of the PRA and PRB failure intensities during operation shows that PRB experienced a maturing period during which its reliability was relatively low but afterwards, its reliability improved and became better than that of PRA.

The comparative analysis provides insight into the evolution of the software and the reliability of two successive products of the TROPICO ESS. However, the results obtained did not allow us to identify the various factors that influence the evolution of the reliability of a family of products. In order to reach this objective, additional information is needed concerning for instance: i) the development process and ii) more than two successive generations of products. Furthermore, the collection and analysis of several failure data sets relative to different families of products will be of great help in this learning phase.

6 Acknowledgements

This work was partially supported by the ESPRIT Basic Research Action on Predictably Dependable Computing Systems (Action no. 6362). The authors are indebted to Marta Bastos Martini and Jorge Moreira de Souza from CPqD for their cooperation in supplying the information needed for this study and their assistance in analysing the data. We also would like to thank Sylvain Metge for his help in processing the PRB data within the framework of a student's project, and Jean-Claude Laprie and Jean Paul Blanquart for their constructive comments.

7 References

- [1] Laprie J. C., "For a Product-in-a Process Approach to Software Reliability Evaluation", in Proc. *3rd Int. Symp. on Software Reliability Engineering*, pp.134-139, Raleigh, NC, USA, October 1992.
- [2] Wallace J. J. and Barnes W. W., "Designing for Ultrahigh Availability: The Unix RTR Operating System", *Computer*, pp.31-39, August 1984.
- [3] Sullivan M. and Chillarege R., "A Comparison of Software Defects in Database Management Systems and Operating Systems", in Proc. *22nd IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-22)*, pp.475-484, Boston, MA, USA, July 1992.
- [4] Kanoun K., Bastos Martini M. and Moreira de Souza J., "A Method for Software Reliability Analysis and Prediction—Application to The TROPICO-R Switching System", *IEEE Trans. Software Engineering*, SE-17 (4) April 1991.
- [5] Kanoun K., Kaâniche M. and Laprie J.-C., "Experience in Software Reliability: From Data Collection to Quantitative Evaluation", in Proc. *4th Int. Symp. on Software Reliability Engineering (ISSRE'93)*, Denver, CO, USA, November 1993.
- [6] Vianna B., "R&D at TELEBRAS-CPqD: The TROPICO System", in Proc. *Int. Conf. Communications (ICC 88)*, pp.622-626, Philadelphia, PA, USA, June 1988.
- [7] Vianna B., Cunha E. C. and Boin F. F., "Hardware Quality Control in the TROPICO System", in Proc. *Int. Conf. Communications (ICC 88)*, pp.632-636, Philadelphia, PA, USA, June 1988.
- [8] Basili V. R. and Perricone B. T., "Software Errors and Complexity: An Empirical Investigation", *Communications of the ACM*, 27 (1), pp.42-52, January 1984.
- [9] Grady R. B. and Caswell D. L., *Software Metrics: Establishing a Company-Wide Program*, 288p., Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA, 1987.
- [10] Kenney G. Q. and Vouk M. A., "Measuring the Field Quality of Wide-Distribution Commercial Software", in Proc. *3rd Int. Symp. on Software Reliability Engineering (ISSRE'92)*, pp.351-357, Raleigh, NC, USA, October 1992.
- [11] Musa J., Iannino A. and Okumoto K., *Software Reliability: Measurement, Prediction, Application*, Computer Science Series, McGraw-Hill, New-York, 1987.
- [12] Laprie J. C., Kanoun K., Béounes C. and Kaâniche M., "The KAT (Knowledge-Action-Transformation) Approach to the Modeling and Evaluation of Reliability and Availability Growth", *IEEE Trans. Software Engineering*, SE-17 (4), pp.370-382, April 1991.