# Repair Software: a Software Platform to Monitor and Optimise the Repair Process for Power Meters

**1 -Caio César Henrique Cunha , 2 -Paulo Regis C. Araújo, 1- Rhuan Victor Crescêncio de Santiago, 1-Thiago Pinho Serpa, 1-Cilis Aragão Benevides, 4- Antonio Clecio Fontelles Thomaz, 1-Antonio Andrio Cordeiro Lima, 3- Henrique Reis Innecco e 3-Jânio Alysson de Oliveira.**

[1] Laboratório Altis Lab, Instituto Iracema, Avenida Dom Luiz, 609, sala 202 e 203, CEP 60160196 - Fortaleza, Ceará.**
[2] Programa de Engenharia, Instituto Federal do Ceará (IFCE), avenida 13 de maio, 2081, CEP 60040-215- Fortaleza, Ceará;*
[3]Departamento P&D, Eletra Energy Solutions, BR-116, 7698 - Km 16 - Pedras, Itaitinga, CEP 61880-000, Ceará; ***
[4] Programa de Engenharia, Universidade Estadual do Ceará (UECE), Av. Dr. Silas Munguba,1700 ,60714-903,Fortaleza, Ceará. **

caio.cunha@altislab.com.br ,pauloregi@gmail.com , rhuanvcsantiago@gmail.com thiagopserpa@gmail.com , cilis@altislabmais.com.br , clecio@larces.uece.br ,andrioantonio@gmail.com , henrique.innecco@eletraenergy.com ,& janio.oliveira@eletraenergy.com.

**Abstract.** Industries and companies are increasingly concerned with the reliability of their products. The international market competition requires that services, systems, and equipment have excellent durability and good performance. Thus, some techniques must be employed in the production process to allow services, systems, and equipment to have a certain degree of reliability, without the need to use hardware and/or software redundancy to achieve fault tolerance. The use of redundancy in a system results in a higher financial cost, longer response time, and greater energy consumption, factors that are decisive in its competitiveness. Thus, fault prevention techniques are essential to allow the manufacture of products with a certain degree of reliability. We can divide fault prevention into two stages: fault avoidance and fault removal. Fault removal is related to the activities where faults are detected and removed. It is like the testing stage. This work presents a software platform that makes it possible to monitor and optimise the repair process during the testing stage (fault removal) of an power meter assembly line of the Brazilian big factory of eletric devices.

## 1. INTRODUCTION

Industries and companies are increasingly concerned with the reliability of their products. The international market competition requires that services, systems, and equipment have excellent durability and good performance. Thus, some techniques must be employed in the production process to allow services, systems, and equipment to have a certain degree of reliability, without the need to use hardware and/or software redundancy to achieve fault tolerance. The use of redundancy in a system result in a higher financial cost, longer response time, and greater energy consumption, factors that are decisive in its competitiveness. Thus, fault prevention techniques are essential to allow the manufacture of products with a certain degree of reliability. We can divide fault prevention technique into two stages: fault avoidance and fault removal (Burns and Wellings (2009)). Fault avoidance is a technique used to prevent the introduction of defective or low-quality elements in the production of systems or equipment.

As there is a difficulty in identifying a more reliable component or device, equipment manufacturers use components from industries already consolidated in the international market. Another critical stage in the fault prevention technique is the fault removal. This stage is based on visual, functional, and operational tests. In functional and operational tests, the manufacturer must be too familiar with the functional or operational states of his products. Otherwise, we will have an improper test phase, presenting validation errors.

Industries with large-scale productions increasingly rely on the best possible functioning of their productive sectors, and technology comes as a great ally when innovating. Information obtained from equipment at industrial parks is used to assist decision making, improve productivity, reduce waste, and other functionalities. It directly or indirectly aims at reducing costs and increasing the quality of products. Companies that manufacture electronic meters, need to reduce problems, reduce waste, and improve the quality of their products.

Some failures are detected during an assembly and testing processes: defective parts, LCD and LED do not turn on, power meter does not calibrate, RAM does not address, among others. Much of these problems are detected after the assembly process and in the testing stage. However, many companies do not have a diagnosis of the symptoms, location, and causes of these defects. And they don't even know how to relate them to optimise the testing process. In some industries, professionals identify defects based on their experience. In some cases, they know the symptom and cause of the failure, but not its location. Or they see the location of the defect (by visual inspection), but not its cause.

Many companies do not survey the number of defects that appear in the assembly and testing processes of their products. And they do not know how to relate these defects with the causes and symptoms. So this work presents a methodology and software capable of quantifying, registering and relating the causes, symptoms, and location of all products manufactured by an Brazilian factory of electric meter.

The relevant contributions of this work can be summarised by the set of benefits presented below:

- Analysis and design of the repair process flow. Survey of functional and non-functional problems for the electronic meters of ABNT, IEC e DLMS protocols.
- An architecture of the repair software (Repair- Software), together with the modelling and creation of the repair system database.
- Diagnostic and testing management modules refer- ring to DLMS, IEC, and ABNT protocols to be included in Repair Software.
- Command execution control management modules referring to DLMS, IEC, and ABNT protocols to be included in Repair Software.
- Modelling and creation of the intelligent traceability protocol to be included in Repair Software.

## 2. STATE OF ART REVIEW

According to Burns and Wellings (2009) the fault prevention can be divided into two stages: fault avoidance and fault removal. Fault avoidance attempts to limit the introduction of potentially faulty components during the construction of the system. For example, the use of the most reliable components within the given cost and performance constraints can improve system reliability. They also comment that the fault removal consists of procedures for finding and then removing the causes of errors. This stage represents the system testing procedures, and according to the authors, it cannot guarantee that all faults are detected and removed. In this paper, we create a methodology and elaborate software to improve the system testing stage.

According to Li et al. (2019), attention has been paid to fault detection (FD) and fault-tolerant control (FTC), due to the increasing demands on system reliability and safety. The authors also comment that many FD methods and schemes have been proposed, which can be generally classified either as model-based (S. X. Ding (2013), Simani et al. (2003)) or data-driven approaches (S. Qin (2003), S. J. Qin (2006)). But in these papers the authors do not present methodologies and software to register, quantify and relate the faults, causes and symptoms.

Zhang et al. (2019) comment that fault diagnosis and fault tolerant control are important research topics, which are used to detect and locate the source of faults, estimate the size of defects, and further ensure the stability and reliability of systems. But these authors present error detection in system runtime as a technique to carry out fault tolerance control. In our work, we implemented new techniques, composed by methodologies and software, to improve the fault detection and to remove it in testing procedures, and consequently to improve reliability.

Kong et al. (2019) present a platform that can monitor each device's power supply process. This platform can detect faults, and the fault tolerant method used by the authors allows protection in real time. Once again, these authors propose fault detection as a stage of fault tolerance. Generally, these techniques use redundancy to detect a fault and recover the system to correct and operational state. As seen earlier, redundancy increases the complexity, financial costs, longer response time, and higher energy consumption of the system. Our solution proposes testing procedure improvement as a good strategy for fault prevention and not fault tolerance.

According to Ye et al. (2019), fast and accurate fault diagnosis is the basis of fault tolerance. They comment that in a complex system it is easier to obtain descriptive information of the malfunctions rather than strictly quantitative information. In the paper presented by these authors, they propose techniques to make fault diagnosis in system run time, and not fault prevention techniques, that do not need more complex solutions in system operation.

## 3. **METHODOLOGY**

### 3.1 Methodology for the survey and update of the technical documentation of Company products

In this stage, the project's research team defined the methodology for the survey and update of the technical documentation of products analyse in this article. The objective was to determine and write the methods for organising and collecting information and documentation related to the repair sector. We follow these steps:

- what information is being collected;
- where the data is located;
- how this information is interrelated;
- what to do if a power meter outside the scope needs to be added within this new procedure that is being created.

As a result, a version of the methodology documentation for the process of gathering information on the products analysed in this article.

The great challenge of the requirement's gathering process was the decentralisation of the documentation that is related to the power meters and its repair process. Also, knowledge of the repair process has much of the process linked to the experience of the technical/fitter team. Currently, we observed there is no transparent process for disseminating knowledge in the repair processes carried out in the company. So, we noted too that the repair process requires the use of technological tools to support the repair process in the functional aspects of all power meter families. These aspects were the biggest obstacle in this information gathering phase because many actions were not documented, requiring a requirement gathering process, which used the observation process as the main instrument to design and understand all stages of the process.

In this stage, we carry out the following actions:

- understand the design of repair process;
- understand how documentation repository process works in the repair process;
- understand how the process of sharing information from repair process documents works.

Based on the information obtained in these actions, we were setting up an environment to centralise all the documentation obtained from the requirements process of each power meter family. This repository aims to consolidate the documentation of each power meter family with the following documents:

- lock diagrams;
- schematic drawings;
- flow of repairs;
- main symptoms related to each power meter;
- list of commands used in the repair process;
- configurations of products analysed in this article;

In the survey of power meter families, a study related to the repair process was based on the fault report and all event records (symptoms and causes of faults from April 2018 to December 2019). The objective of this study was to set up some views, seeking to find the best strategy to start the requirement gathering process. We assembled the followings views:

- family and power meter models and the number of faults (or symptoms);
- symptom and fault location in each power meter model;
- symptom and cause in each power-meter model of power-meter families analysed in this article.

For example, Figure 1 shows tables with power meter models of the same family and power meter families with their respective fault records. These tables include two types of faults: visual and functional, and they have different aspects and impacts. Later, we decided to carry out the flowchart of all functional faults, which are the most critical in the repair process.

A)

| Modelo | Qtde de Registros |
|---|---|
| Cronos 6001-A | 11130 |
| Cronos 7023L | 9325 |
| Cronos 7023A | 5815 |
| Zeus 8023 | 5240 |
| Cronos 6021L | 4753 |
| Cronos 6003 | 3631 |
| Cronos 6021-A | 3137 |

B)

| Linha | Qtde de Registros | Percentual |
|---|---|---|
| CRONOS | 40784 | 78,74% |
| ZEUS | 6833 | 13,20% |
| ARES TB | 2509 | 4,84% |
| ARES THS | 931 | 1,80% |
| APOLO | 739 | 1,42% |
| **TOTAL** | **51796** | **100%** |

Figura 1. A. Power meter of the same family, B. Power meter families with their respective fault records

Table A of Figure 1 shows the number of faults per power meter model of Kronos and Zeus families. Table B of the same figure presents the number of faults and their percentage of all power meter families.

In Figure 2, Table A demonstrates the symptoms that are most common for each power meter model. For example, the most common symptom of Cronos 6001-A model is its calibration error; the most common symptom of Cronos 7023L is damaged LCD; Zeus 8023 is short circuit; Cronos 6021L is internal dirt; Cronos 6003 is damaged base; and Cronos 6021-A is LCD does not turn on.

After this initial study, the requirement gathering process was initiated by the ZEUS family power meter, which is one with more defects and detailed documentation. The survey carried out for this power meter model served as a basis for validating the methodology for surveying the requirements for the documentation that will serve as a record of the work carried out and the actions implemented to improve the repair process. After that, the process of updating, centralising, and improving the technical documentation of the products analyse this article began. This procedure was used as a reference to obtain information from the other families of power meters (ARES, APOLO, PANTHEON, and CRONOS), which were consolidated in the technical documentation of all products analyse in this article.

A)

| Modelo | Sintoma |
|---|---|
| Cronos 6001-A | Medidor não Calibra |
| Cronos 7023L | LCD danificado |
| Cronos 7023A | Componente sem Protrusão |
| Zeus 8023 | Curto Circuito |
| Cronos 6021L | Sujeira Interna |
| Cronos 6003 | Base Danificada |
| Cronos 6021-A | LCD não liga |

B)

| Localização | Qtde de Registros |
|---|---|
| Teste do LED | 13617 |
| Inspeção SMT | 8801 |
| Calibração | 8701 |
| Inspeção PTH | 4168 |
| Verificação | 4035 |
| Endereçamento | 2612 |

Figura 2. A. Power meter of the same family, B. Power meter families with their respective fault records

Correlation between the tests and the possible causes of faults this stage intended to create a repair troubleshooting documentation. For this purpose, a survey of the primary defects (or symptoms) of this power meter family was carried out; the location where the fault was detected; the cause of the symptom; the area causing the defect, among other information. This process was carried out in several locations at the Company plant (for example PTH sector, STM sector, initial Assembly sector, etc.) for a power-meter family. The great challenge of this process was that the procedures to be carried out and their possible solutions were not mapped. This information was recorded with the experience of the technical team.

We, aiming to define the best work methodology, made a pair of Zeus family power meters. We verified that in 111 defects (or symptoms) found, more than 81.24% of the occurrences were concentrated on 13 defects (or symptoms). This result was obtained with more than 6833 occurrences presented between April 2018 and February 2019. Based on this information, we observed what procedures had to be carried out to relate the symptoms to be repaired and their repair solutions, based on the experience of the repair team. All the information collected had to be confirmed with the engineering sector, because, many times, the repair team gave incomplete information or without a theoretical foundation. We used the same procedure to make parrot os for the other families of power meters. The results obtained with these procedures were: for Cronos family power meters, it was found that in 174 defects (or symptoms), more than 80.41% of the occurrences are concentrated on 25 defects (or symptoms); for Ares family power meters, we found that in 113 defects (or symptoms), more than 80.24% of the occurrences are concentrated on 19 defects (or symptoms); and for Apollo family power meters, in 40 defects (or symptoms) found, more than 81.24% of the occurrences are concentrated on 8 defects (or symptoms). Figure 3 shows a table with parrot os for all power meter families.

| LINHA | QTDE (REG / SINTOMAS) | PARETO 80% |
|---|---|---|
| ZEUS | 6833 / 111 | 5551 / 13 |
| CRONOS | 40784 / 174 | 32794 / 25 |
| ARES TB | 2509 / 90 | 2024 / 18 |
| ARES THS | 931 / 74 | 751 / 19 |
| APOLO | 739 / 40 | 604 / 8 |

Figura 3. Table of parrot os for all power meter families

Standardisation to identify and record new defects or symptoms in the System We noticed that there was a lack of standardisation to create new defects or symptoms. As a result, several symptoms were recorded repeatedly, or there were symptoms with little occurrences or that were very generic. We analysed the symptoms recorded within the EA system, and the possibilities for improvement were identified. A list with improvements was created, and the decision on whether to comply with these changes to correct the symptoms database was made by quality area in partnership with engineering area. This procedure allowed updating the defect database by correcting duplication, generic defects, as well as standardising the creation of new errors.

Initially, we verified that automation during the process of identifying defects could aggregate more benefits to the manufacturing process. When we analysed the situation with more detail, we noticed that some cases could generate many errors of automatic notifications, such as when a component is not fitted correctly in the testing machine. Then, engineering team and we decided that the automatic symptom notification was not a good idea because various defect notifications could happen, thus causing several false positive notifications.

The flowchart that relates identified problems, test procedures, and their solutions as a methodology for improving test and repair procedures, we decided to develop a repair flowchart. Initially, we planned to make a repair flowchart for each power meter; that is, each power meter and each symptom would have a different repair flowchart. But, we observed that this procedure was not needed, because the repair flowchart was the same for all power meters of the same family. Then, instead, create a flowchart individually for each power meter, we decided to make a flowchart for each power meter family. Thus, the survey of flowcharts was carried out by the power meter line, with a particular focus on the Zeus family.

We observed that the approach used to search the information could be improved to maximise time. Initially, flow information was collected directly from the repair team, but each repair operator provided a different opinion and often did not match with the engineering team's view. So, we decided to analyse some flows carried out by the engineering team, and, with this vital background, to gather the data again with the repair team. After that, the basic model defined by the engineering team with our modifications, based on data collected from the repair team, had to be approved by the engineering team.

The methodology used was based on the following elements: 1) observation of the work routine in the repair process; 2) analysis of technical documentation; 3) process design validation with stakeholder team;4) Flowchart validation with practical cases found in Eletra's production and repair sector.

This survey process started with the Zeus family power meters, in which more than 45 flowcharts were described to meet the main problems, procedures, and solutions. For example, 1) LCD does not connect; 2) Alarm activated; 3) Communication failure; 4) Failure in optical communication; 5) Power meter does not calibrate; 6) 12V Voltage Failure, etc.

Figure 4 shows the Zeus family production flow chart, which contains the power meter production flow with all the tests that are carried out. In this flowchart, we can observe visual tests that are represented by blue boxes, such as PTH and SMD inspections. The purple boxes represent functional tests, such as insulation, calibration, and verification tests, addressing and LED criteria, integration with a communication module and laser tests, and final and process qualities.

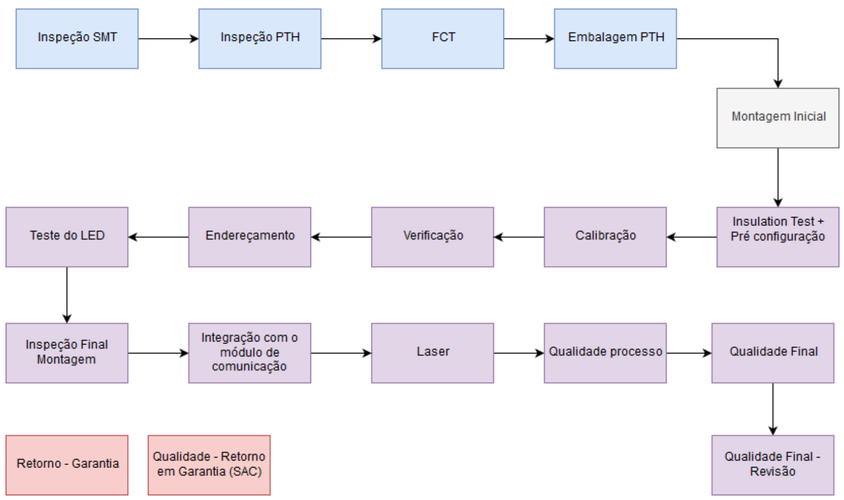


Figura 4. Production flow chart and repair points for Zeus family

3.2 Repair Software for Controlling Test Execution and Repair Procedures

We revised repair flowcharts for creating repair software. First, we carried out a detailed study of power meter communication protocols of all power meter families produced by Company, which are composed as follows, 1) DLMS Protocol for Zeus and Pantheon power meters, 2) ABNT Protocol for Ares power meters, and 3) IEC Protocol for Cronos and Apolo power meters. After, our team analysed the development platform EA, which will be the basis to develop the repair software. On this platform, the entire architecture of the solution will be assembled. Furthermore, other functionalities will be included, such as the development of records, solution reports, and solution integration that will support verification and validation processes of the functional problems for all power-meter families produced by Company. For communication protocols, ABNT, DLMS, and IEC, all functionalities were enumerated. As an example, for ABNT and DLMS protocols, two kinds of readings were included: simple reading and standardised reading. In this stage, we had a significant challenge with the ABNT communication protocol, which uses the NBR 14522 standard. This standard is not very explanatory and deficient in relevant information for all functionalities of power meters, and power meters have some features not covered by ABNT standard commands.

Another challenge found in this protocol study was to establish communication with the power meters using the DLMS communication protocol, which is still incipient in Brazil. Without any reference from other national competitors, some types of data and objects had to be understood and then created to meet some features of the power meters.

Software architecture Due to Company uses as the primary technology for software development languages .NET and C#, the company asked us to use the same languages because the future software maintenance will be carried out by the engineering team.

After a detailed analysis, we observed that the ideal architecture model must be based on the graphical interface of repair software developed within the EA platform (within the browser). The main challenge to create the architecture was communicating with the power meter, which must be connected through an optical port with the repair operator's local computer. After that, the repair technician performs his activities in a browser, which will be running on a Web page with the EA's version. Another difficulty was the library used in the communication layer with the power meters, which was developed by the engineering team, and has its version developed in Java. This layer, known as MCI, has the rules of the three communication protocols (DLMS, ABNT, and IEC).

We elaborated on an architectural model that uses the communication between the power meter and the Web browser. Engineering team suggested using the SignalR framework, which creates a communication layer using WebSockets. Our team started testing this technology in which it is well documented and did not present many technological challenges for its use. However, to use the SignalR technology, which is developed in C#, it needed somehow to have the MCI library also developed in C#, but it was established in Java. Based on this new challenge, the idea that came up was to convert the library using conversion tools between languages since the library reported here is very complicated. It would be hard to rewrite it in another language. After this analysis, the architectural model was elaborated and it is presented in Figure 5.

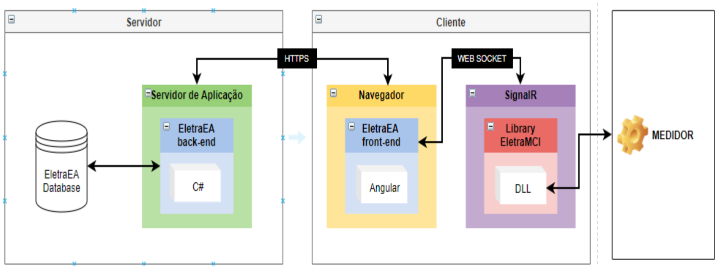

Figura 5. Architectural model of repair software

We tested some conversion tools. The first one was IKVM.NET, which aims to convert the library .JAR of MCI in DLL. Another tool validated was JNBridge Pro tool, which creates an interoperability bridge, genera- ting a set of proxies that exposes the APIs of the classes and manages communications between the .NET and Java classes. And finally, we checked the jni4net tool, which also converts a library .JAR of MCI in DLL.

After exhaustive tests, we converted MCI's library from java to a DLL. Still, some dependencies were not able to be converted. So, our team decided not to continue with the conversion technique because in case any future error occurred in the system, there would be no way to guarantee that it was a development bug or due to unsuccessful conversion.

After necessary tests, we confirmed that the MCI's library was working correctly regarding its two ways of accessing the power meter, both optically and by TCP. The next step was the creation of a socket server in Java language to connect with the browser. For this purpose, a basic HTML page was created without a front-end framework. The native WebSockets library was used to access the socket server created in Java. The MCI's library uses some information to decide which form of communication and which parameters need to be passed on to get access to the power meter. This information needs to be collected through the front-end, transferred via WebSocket to the socket server in Java, which will later be transferred to MCI's to access the power meter. The tests were successful, thus getting the browser to receive information from the power meter. After this validation, the architectural model was changed, and it is presented in Figure 6.

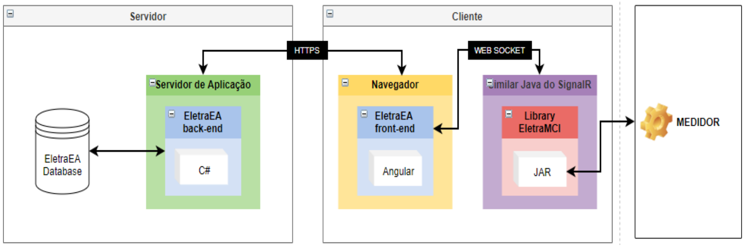


Figura 6. New architectural model of repair software

## 4 RESULTS

After initial tests, to complete the cycle and ensure that the proposed architecture works, the Web socket communication part with the Java socket was developed within ES. We carried out the final test cycle to ensure that the proposed idea was fulfilled. Figure 7 shows a repair software's screen with all functionalities to support the repair procedures. On the left side, the screen shows the software's functionalities in which we selected the functionality called symptoms or fault type. This functionality has two options, new and search. The option called new allows to include a new symptom or fault type. And the option called search is used to select a symptom or fault type. On the right side of the screen, the software presents the search functionality of symptoms. Columns represent these characteristics: the first column is the symptom's identifier; the second column is the symptom's name; the third column is symptom's state which can be activated or not; fourth is to inform if exists the symptom is the cause; and, finally, the fifth column identifies the object to be repaired.


Figura 7. Software screen for symptoms

Figure 8 shows a repair software's screen with all functionalities to support the repair procedures. On the left side, the screen shows the software's functionalities in which we selected the functionality called fault cause. This functionality has two options, new and search. The option called new allows to include a new fault cause. And the option called search is used to select a fault cause. On the right side of the screen, the software presents the search functionality for fault causes. Columns represent these characteristics: the first column is the fault cause that can be dust, damaged functional component, guide of damaged LED, and so on; the second column is to inform if is necessary to inform the component location; the third column is to inform if is necessary to carry out all tests again; fourth is the state of the fault cause, activated or not; and, finally, the fifth column identifies who registered the fault cause.


Figura 8. Software screen for fault causes

## 5. CONCLUSION

This work developed a software platform to monitor and optimise the repair process for power meters called Repair Software. First, We had to survey functional and nonfunctional problems for the electronic meters of ABNT, IEC, and DLMS protocols. After, an architecture of the repair software (Repair Software) and the modelling and creation of the repair system database were proposed. We developed diagnostic and testing management modules referring to DLMS, IEC, and ABNT protocols in Repair Software. Other modules, such as the command execution control management and the intelligent traceability protocol, were also included in Repair Software.

In the test stage, Repair Software successfully showed the power meter family and models and the number of faults or symptoms associated with them. It worked well to show the location of symptom and fault in each power meter model. Repair Software also obtained the correct relations between symptoms and causes in each power meter model of power meter families produced by Company.

## 6.ACKNOWLEDGMENTS


This work was partially supported by Fundação de Cultura e Apoio ao Ensino, Pesquisa e Extensão (FUNCEPE) and Ministério de Ciência, Tecnologia e Inovação (MCTI)


## 7.References


[1] Alan Burns and Andy Wellings. Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX. *Addison-Wesley Educational Pu- blishers Inc*, 2009.

[2] L. Li, S. Ding, H. Luo, K. Peng, Y. Yang. Performance- based Fault-Tolerant Control Approaches for Industrial Processes with Multiplicative Faults. *IEEE Transactions on Industrial Informatics*, 2019.

[3] S. X. Ding. Model-Based Fault Diagnosis Techniques - De- sign Schemes, Algorithms and Tools. *2nd Edition, London: Springer-Verlag*, 2013.

[4] S. Simani, S. Fantuzzi, and R. J. Patton. Model-Based Fault Diagnosis in Dynamic Systems Using Identification Techniques. *Springer-Verlag*, 2003.

[5] S. Qin. Statistical process monitoring: Basics and beyond.
*Journal of Chemometrics*, vol. 17, pp. 480-502, 2003.