NETLOG 2018
International Conference on Network Enterprises & Logistic Management

STUDY OF THE FLOWCHART OF THE PARACONSISTENT ANNOTATED LOGIC (LPA) ALGORITHM TO SUPPORT DECISION MAKING IN PROJECT RECOUNT IN THE FUNCTION POINT ANALYSIS TECHNIQUE

Luiz A. de Lima[1]*, Jair M. Abe[2], Jonas P. da Silva[3], Caique Z. Kirilo[4], Giovanna Albertini[5]

Graduate Program in Production Engineering, Paulista University,

Rua Dr. Bacelar 1212, CEP, 04026-002 – São Paulo – SP – Brazil,

[1]*luiz@wcisp.com.br, [2]jairabe@uol.com.br, [3]jonassunip@gmail.com, [4]caiquez.kirilo@hotmail.com, [5]anne.albertini@live.com. *Corresponding author.

ABSTRACT.

Since the 1990s, public and non-governmental organizations in the Brazilian market have decided to purchase software and contract software services through biddings and based on the technique of point-of-function analysis worldwide recognized and used by companies with maturity levels and certified by international seals. This technique allows us to measure the size of the software and helps, based on technological productivity, to estimate the calculation of the effort, time and cost of projects quite accurately. The report of activities in consultancies between client and suppliers in the hiring of software increasingly permeates the need of the project managers to hire new specialists in function points to gauge and decide the continuity of new projects. In this article, the technology aims to propose assistance in the decision of new hires of specialists to recount projects counted by "n" suppliers, due to distortions among specialists' opinions on counting function points in new software projects. In addition, one of the main benefits is characterized by support for decision making by managers or company directors with regard to counting approval. To do this, the goal is to make use of flowchart concepts of the Paraconsistent Annotated Logic algorithm (LPA) to improve the implementation process and to meet the proposition in decision making for recounting. In addition, to propose the study of the method that fulfills the proposition "To authorize the recount of the project in the analysis of function point?". And so, in this study, it suggests a new horizon for public and private companies to mitigate time, cost and resources, giving rise to demands damped by discrepancies in counting function points.

Keywords. *Paraconsistent annotated evidential logic Eτ, Paraconsistent annotatedlogic, Decision-making, Software engineering,Software Metrics;*

1. INTRODUCTION

1.1. SYSTEMS MEASUREMENT

In every scientific community of software engineering, some kind of improvement in the accuracy in software estimation, through attempts of models based on the measurement of software size, is being investigated, such model continues to evolve from the model LOC (Lines -more-codeMetrics)-(Boehm, 2005), UCP (Use Case Points) created by Gustav Karner(Leszekmaciaszek & Kang Zhang, 2015) as described by Schneider and Winters, NESMA (Function Point Metrics) maintained by the BFPUG(IFPUG, 2010) (BrazilianFunction Point UsersGroup) in Brazil, the latter adopted by several Companies, due to its international acceptance(Carvalho, 2011).

In 1986 the IFPUG (InternationalFunction Point UsersGroup) was created and started to control and standardize the technique through the CPM (CountingPractices Manual), currently in version 4.3.1(IFPUG,2010).

In the Netherlands Association of Software Metrics Users, formerly known as NEFPUG (Netherlands Function Point Users Group) was founded in May 1989 at NESMA. The objective of this model is to serve the management of improvements for the construction of information systems. The basis of this model is in the evaluation of the scope and size of the improvements (maintenance) applicable in the measurement of systems, where the results achieved are independent of the professional developers who will use the method. In the search to measure maintenance in any system improvement, the NESMA method(Boehm, 2005) is used, because this is one of the strengths, in relation to the other techniques.

According to researchers(IFPUG, 2010), estimates that use Function Point Analysis techniques to measure software size has also been used as references when it comes to project hours acquisition (software development lifecycle). A function point is a unit of measurement of software that has been used in a consolidated way by the public and private companies, both in Brazil and internationally, as a strategic tool for the process of acquiring services and new systems. In the Function Point count, the size of the software is measured by the quantification of external functionality required by the user, regardless of technology.

In general, error-free, timely and budget-driven software projects are not often found between demands delivered by software factory suppliers because they are largely underestimated and even with inaccuracies in their initial estimates. This imprecision(Boehm, 2005), of effort generally reflects the increase in costs, deadlines, and throughout the project life cycle delays in deliverables, resulting in contractual fines between customers and suppliers.

Faced with these occurrences, there are great losses of new business and renovation in the continuity of projects. Despite imprecise estimates, companies can gain insight into data stored on historical bases for a given period. Such knowledge produces reliable quality and productivity indicators[5] to aid in the choice of systems measurement.

Companies use estimation processes to provide information to foster success in project management and their respective areas of knowledge with real gains in planning and controlling activities throughout the project life cycle(PMI, 2017).

1.2. FUNCTION POINT ANALYSIS

1.2.1. FORMULAS AND PROJECTS

The calculation of function points is standardized in the manual of the practice of counting(Jones, 2010) and are determined according to the type of project. Thus, there are formulas that meet new projects, projects already installed, improvement and post-improvement, as follows:

- The formula for calculation of Function Point in new projects that will follow for the creation process: DFP = (UFP + CFP) * VAF

1.3. PARACONSISTENT LOGIC

The logic for Aristotle is a tool for right thinking. The propositions raised as an argument and inferred in the conclusion are based on observations(Jones, 2010). Therefore, the conclusion and the propositions can not be treated as only truth or falsehood, but always observe, seeking to feed the reasoning to knowledge. The propositions raised from reality must follow three Fundamental Principles of Logic: Principle of identity (X is equal to itself and unlike all the rest); Principle of non-contradiction (no statement can be true and false at the same time); Principle of excluded third (there is no third possibility, besides true and false).

Paraconsistent Logic is among the non-classical logical calla(Abe, 2010) since it contains provisions contrary to some of the basic principles of Aristotelian Logic, such as the principle of contradiction. From the Aristotelian point of view, the three principles of logic prevail. The predecessors of the Paraconsistent Logic were the Polish logician J. Łukasiewicz Lvov in 1878 and the Russian philosopher N.A. Vasilév. Vasilév baptized a logic that became known as imaginary. Łukasiewiczanounced the trivalent Logic: True, False, Possible. The first logical to structure a paraconsistent propositional calculation was the Polish S. Jaśkowski, the disciple of Łukasiewicz.

The term "Paraconsistent" literally means 'next to consistency'. However, 1976 the philosopher scientist Francisco Miró Quesada, called the logic of "Paraconsistente". According to the Paraconsistent Logic, a sentence and its negation may both be true (Newton C. A., Jair Minoro Abe, Afrânio Carlos Murolo, and João I. da Silva Filho in 1999). In the mid-1950s, the Polish S. Jaskowski and the logical mathematician Newton C. A. da Costa proposed the contradiction in the logical structure and became known as the founders of Paraconsistent Logic(Abe, 2015).

1.4. PARACONSISTENT ANNOTATED LOGIC

In the day-to-day of our reality in front of innumerable sources of information, the contradiction constantly occupies a space, bringing uncertainties that will culminate soon or future contestations. In activities such as analysis of clinical exams, in politics, in the analysis of legal processes, in the measurement of software, technical support, in the care of insurers, where at least two specialists are involved, there will always be different points of view. In the case of a system with artificial intelligence, neural networks(Abe, 2015), also known as "machinelearning", which starts from the study of pattern recognition, the appearance of contradiction in logical reasoning is inevitable when we try to reflect human behavior. In response to the contradiction we have the LogicParaconsistent.

The annoted paraconsistent logic is a class of Paraconsistent Logic that works with propositions of type p (μ, λ), where p is a proposition and (μ, λ) indicate the degrees of favorable evidence and contrary evidence, respectively.

The pair (μ, λ) is called the annotation constant, with the values of μ and λ being limited(Abe, 2010), between 0 and 1.

The evidence is reflected in the collection of the value or degree, which is a number found in the set of real with an interval between 0 and 1. Given the characteristics to meet an analysis in a given proposition, we were able to extract that degree or value. The source of information can be a specialist in software measurement, where the value extracted is based on the level of knowledge of this specialist, the experience of count counts or even the experience gained in his profession of function point analyst. So, thinking about the paraconsistent analysis system that addresses uncertainties, one must start with repository modeling containing knowledge of the information collected from the specialists in function point analysis.

One way of representing the paraconsistent logic that allows to perceive the real reach and thus extract results to support in the decision making, is faced with the understanding of the diagram and its degrees of certainty and uncertainty, grouped in extreme states identified in the results (1 to 4) and non-extreme states displayed in the results (5 to 10), with adjustable control values representing limit values: C1 = C3 = ½ e C2 = C4 = -½; C1: Vcve = maximum value of certainty control; C2: Vcfa = minimum value of certainty control; C3: Vcic = maximum value of the uncertainty control; C4: Vcpa = minimum value of the uncertainty control;

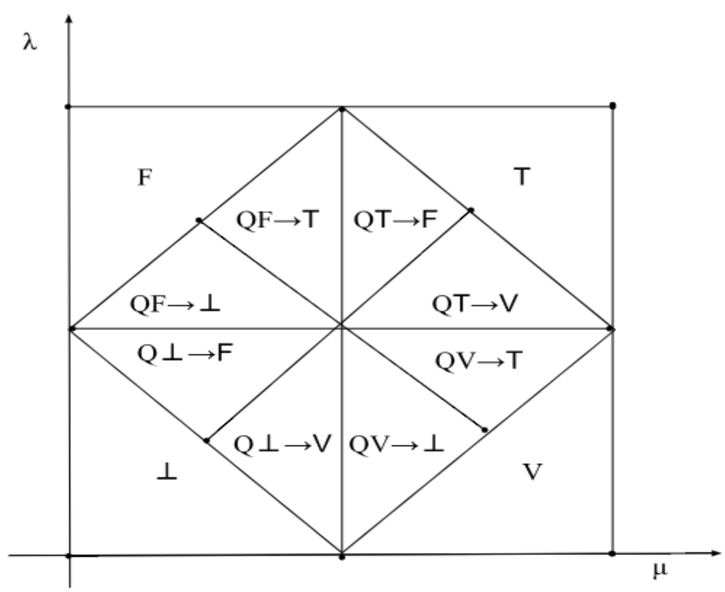

FIG. 1. Extreme and non-extreme States. Source : (Abe, 2015)

In the representation of the diagram the following understandings with symbols(Abe, 2015) and their 10 possible results were used, being 1 to 4 extreme states and 5 to 10 non-extreme states:

Extreme State:

1. V, True
2. F, False
3. T, Inconsistent
4. ⊥, Paracompletenessn

Non-extreme:

5. QV→ T, almost true tending to the Inconsistent
6. QV→ ⊥, almost true tending to Paracompletenessn
7. QF→ T, almost false tending to the Inconsistent
8. QF→ ⊥, almost false tending to Paracompletenessn
9. QT→ V, almost inconsistent tending to Truel
10. QT→ F, almost inconsistent tending to false

2. MATERIAL AND METHODS

The methodology applied in this article followed the paradigm Design Science Research(A.R. Hevner, 2007). According to Hevner(A.R. Hevner, 2007), this research reflects on cycles of related activities. Each cycle(Clive, 1991) must be matched firstly in the relevance of the subject addressed, along with the elicitation of requirements

and criteria of evaluation of the research. In the second cycle, a process must be sought to solve the problem according to relevance, requirements, and criteria raised. And finally, to foster the generation of knowledge by the cycle of rigor

The problem addressed by this research concerns the process of recounting in function points, by a new expert in measurements already counted by a group of experts of the same importance (weight) in the experience in counts above 1000 FP. However, we did not discard different weights in a future approach. The identification of the problem was based on the literature and the consulting work on process improvement in the acquisition of software and services from software factories with organizations, where a database (count values since 2010 and CFPS of the same importance) was made available by consulting Winforma. It should be noted that, by secrecy, no identification of those involved (projects, suppliers, clients, CPFS) was made available. To know the state of the art about the counting of discrepancies involving contracting systems and services, it was necessary to review the literature and map processes that resulted in the discovery of gaps in the research topic, mainly: project managers, size discrepancies even software between counts made by specialists, processes of implantation of non-classical logics, lack of tools with algorithms of artificial intelligence, neural networks(Abe & Nakamatsu, 2013), based on studies of logic, since it contains provisions contrary to some of the principles of Aristotelian logic, as the principle of contradiction.

Considering the problem raised and identified gaps, it proposes processes and subprocesses of the paraconsistent logic that is able to guide the implementation of the tool as support in decision making. This proposal should address evaluations and provide benefits and efficacy to the process in deciding to hire resources for recounting in new projects that use point-of-function analysis such as systems measurement. His evaluation was based on a case study of software factories consulting in counts defended by specialists.

The contribution in the generation of knowledge is in the proper application of existing methodologies contributing to the construction of a knowledge base. In this research, some foundations of literature review, artificial intelligence[12] with paraconsistent logic and evaluation methods were studied. It is highlighted as the main advancement, the new method called a unified process of annoted paraconsistent logic (UPAPL) as an aid in the decision-making of recounting and the study of propositions that can be met with the use of paraconsistent logic. The solution can be used by other institutions and even contribute to the creation of new decision-making tools.

3. RESULTS AND DISCUSSION

3.1. MEASURING SOFTWARE USING FUNCTION POINT

The counting process is based initially on the choice of the type of projects, is it can be development, improvement or even application. The development project is characterized by demands on counting new systems. The improvement project is classified when there is a need for maintenance/improvement in existing systems. The demand for application projects is characterized by demands on counting already implemented systems. It is important to note that you can have a measurement from your new system, its improvements, and always keep the updated count in the application design as a way to guarantee function point repositories.

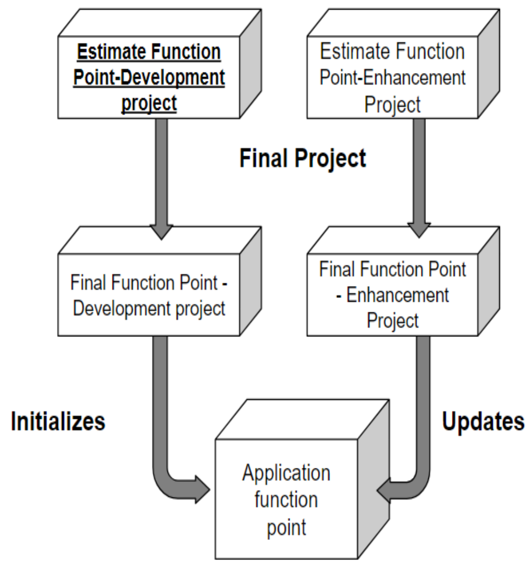

FIG. 2. Relationship between of Types of Function Point Counts (source: author)

3.2. CONTRIBUTION AND FUNCTIONAL COMPLEXITY

The complexities were standardized in the CPM manual by the IFPUG. In this pattern, the function points have been distributed according to the functionalities and their respective complexities.

3.3. PARA-ANALYZER ALGORITHM

The Para-Analyzer algorithm uses, as the object of analysis, values of the degrees of favorable evidence and of contrary evidence to draw conclusions with the results of the degrees of Contradiction and Certainty(Abe & Nakamatsu, 2015).

The definition of the Paraconsistent Decision Method (MPD) proposed in the studies(Carvalho, 2011), reflects the method that assists decision making through Paraconsistent Logic(Costa & Abe, 1999).

One of the ways of representing the paraconsistent annotated logic(Costa & Abe, 1999) with possible implementation in a particular programming language, is to launch the use of the flowchart, where we have:

Initially limit values: ULV limit value Paracompleteness, TLV Limit value True, ILV Limit value Inconsistent, FLV False threshold value can be standardized with values between -1 and +1. Then, the values of X1 and X2 were entered, where the first one asks for the degree of conviction (Success) and the second one asks for the degree of Uncertainty (Sucess). Given these values, we calculate the degree of certainty GCe (X1-X2) and the degree of contradiction: GCo (X1 + X2-1) to verify the possibility of the answer being true, false, inconsistent, complete. Since this step is impossible to answer, it follows in the "X" flow to explore the possible answers offered by the paraconsistent logic

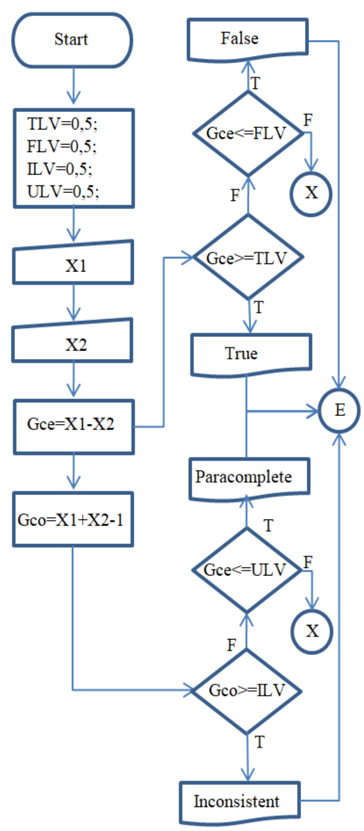

FIG. 3. Paraconsistent logical flowchart: True, False,

Inconsistent, Paracomplete (Source: Author).

In this stage of the flowchart, there is a great possibility of being almost true tending to the inconsistent, or inconsistent tending to the True, because the GCe and Gco conditions result in some response and when there is no possibility to answer, it follows in the "Y" flow to explore the possible answers offered by the paraconsistent logic.

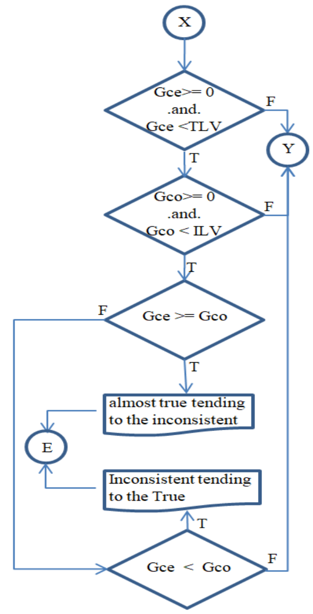

FIG. 4. Paraconsistent logical flowchart: Almost True tending to the Inconsistent, Inconsistent tending to the True (Source: Author).

In this stage of the flowchart, there is a great possibility of being almost true tending to the Paracompletenessn or full Paracompletenessn tending to the True, since the GCe and Gco conditions result in some response. And when there is no possibility to answer, it follows in the stream "Z" to explore the possible answers offered by the paraconsistent logic.

In this stage of the flowchart, there is a great possibility of being almost false tending to full Paracompletenessn, or full Paracompletenessn tending to false, because the GCe and Gco conditions result in some response. And when there is no possibility of an answer, it follows in the flow "W" to explore the possible answers offered by the paraconsistent logic

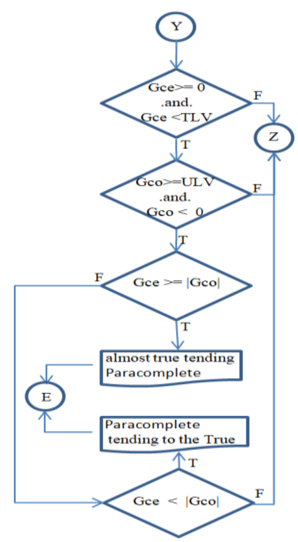

FIG. 5. Paraconsistent logical flowchart: Almost True tending to Paracompletenessn, Paracompletenessn tending to the True (Source: Author).

.

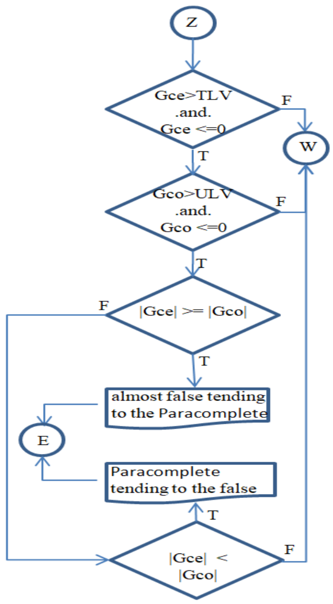

FIG. 6. Paraconsistent logical flowchart: Almost False tending to Paracompletenessn, Paracompletenessn tending to False (Source: Author).

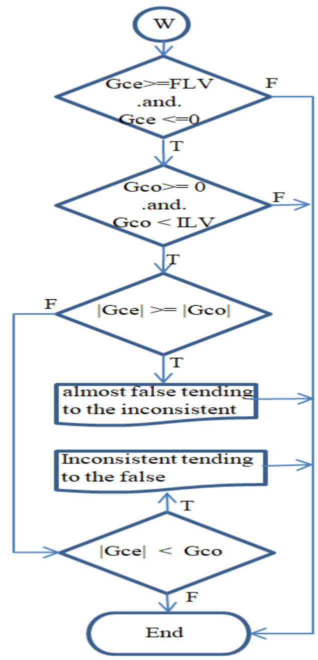

FIG. 7. Paraconsistent logical flowchart: Almost False tending to Inconsistent, Inconsistent tending to False (Source: Author).

In this final (Fig.7) step of the flowchart, there is a great possibility of being almost false tending to the inconsistent, or inconsistent tending to false, because the GCe and Gco conditions result in an answer.

3.4. PROPOSITIONS

By using Paraconsistent Annotated Logic (LPA) (Abe, 2011) in supporting decision making in project recount in the function point technique, one has the possibility to mitigate numerous defenses between clients and suppliers that cause a longer design time and bottlenecks of new demand.

In order to do so, we propose propositions, which reflect projects without recount needs using the function point analysis technique, and as the object of studies to apply the paraconsistent logic, such as:

- True: The counts have the same score between the contractor and the contractor.
- Inconsistent: The counts are the same in some respects and different in others, between the contractor and the contractor.

Other propositions that reflect projects with recount needs were grouped using the function point analysis technique and as the object of studies to apply the paraconsistent logic in four states, such as:

- False: Counts are different between contractor and contractor.
- Paracompletenessn: The counts are missing information. Not being able to analyze.

3.5.NORMALIZATION

3.5.1. MAXIMIZATION AND MINIMIZATION

The raw data set must be transformed into paraconsistent data, and thus enable analysis with the paraconsistent logic. Regarding the project counts in each group of experts, the count with the minimum score and maximum score between the counts made by the specialists is separated, since among the specialists.Therefore, we refer to the minimum and maximum function point totals with the paraconsistent logic with zero and one.

Table 1. Normalization with Maximization and Minimization

Weight (above 1000FP)	**Specialist**	**Count**	**Min e Máx**	**Normalization**
02	CFPS1	15PF	Máx.	1
01	CFPS2	09PF	Min.	0
00	CFPS3	13PF	-	-

(Source: Author).

3.6.GLOBAL ANALYSIS

In order to reflect the joint influence of all factors with weight in each decision, one must take into account the Global Analysis (center of gravity) (Carvalho, 2011) that represent on the Cartesian plane the factors in the reticulate(Abe, 2009), and are collected by the degree of conviction (Success) and the degree of uncertainty (Hesitus). The calculation of the Global Analysis(Abe, 2009), can be extracted by the weighted average of the evidence of conviction and uncertainties resulting from all the factors. When the weights in each decision are equal, the Global Analysis should be calculated by the arithmetic mean of the evidence of belief and uncertainty, becoming the geometric center.

3.7.RECOUNT DECISION-MAKING

Function point specialists count systems using the CPM(Vazquez & Simões & Albert, 2010) manual so that it is possible for Project Managers(PMI & Pressman, 2011) to seek results through the mental-cognitive process of the specialists or group of experts who have worked on the measurement in function point analysis(Vazquez & Simões & Albert, 2010). The decision-making(Carvalho & Abe, 2011) process consists of choosing one of several alternatives. The unified process of annoted paraconsistent logic(Abe, 2010) is proposed as an aid in the decision-making of recounting, as follows:

Table 2. Unified macro process Paraconsistent annotated logic

Item	Process	SubProcess
A	Definition	Define Proposition; Define Factors; Define Section; Define Database;
B	Transformation	Generate Normalization; Use Evidence (favorable and unfavorable);
C	Calculation	Calculate (Maximization; Minimization;Evidence [Resultant Min and Max];Degree [Gce:Certainty,Gco:Contradiction];Globals Analysis (Gce:Certainty,Gco: Contradiction);
D	Parameterization	Parametrize Limitvalues;
E	Processing	Process Para-Analyzeralgorithm;
F	Decision-making	Assists decision-making;

(Source: Author)

4. CONCLUSION

In this article, I had the perception of the importance of the research with regard to the organization in the implementation of the macro process using the Paraconsistant Annotated Logic, which was proposed in six processes (Definition, Transformation, Calculation, Parameterization, Processing, Decision Aid) with respective subprocesses necessary for the success of a structuring of the technological tool, in addition to providing studies of the main propositions and defining which can be met as the focus of the project in the gains to mitigate recounts by new specialists of software factories, thereby increasing the productivity of demand in the cycle and project life. In this evolution of the studies, it was possible to arrive at the main proposition "to recount project in an analysis of function point?", To make possible the implantation of the flow for the algorithm of Paraconsistent Annotated logic (LPA).

In terms of market needs at the national level of companies that need to contract new software and postpone such hiring due to discrepancies found between counts made by specialists (with the same importance "weight" in the experience in counts) from different suppliers of software factories. It is expected that significant results culminate in the construction of the technological tool to support decision making, which reflects in the support to the decision by the managers in the proposition "recount project in an analysis of function point?", Together with the Paraconsistent Annotated Logic.

5. REFERENCES

A.R. Hevner, and Scand. J. 2007. A three cycle view of design science research; Inf. Syst. 19: 87–92.

Abe, J. M. 2009. Lógica Paraconsistente Evidencial Et.: Monografia.

Abe, J. M. 2010. Paraconsistent logics and applications. In: 4th International Workshop on Soft Computing Applications. IEEE, p. pp. 11–18.

Abe, J. M. 2011. Lógica Paraconsistente Anotada Evidencial Et.: Santos: Comunicar.

Abe, J. M., Akama, S. and Nakamatsu, K. 2015. Paraconsistent Intelligent-Based Systems - New Trends in the Applications of Paraconsistency. : 1. ed. Switzerland: Springer International Publishing, v. 1, 2015. ISBN DOI: 10.1007/978-3-319-17912-4

Abe, J. M., Lopes, H. , F. S. and Nakamatsu, K. 2013. Paraconsistent artificial neural networks and EEG. International Journal of Knowledge Based Intelligent Engineering Systems, v. 17, n. 2, p. 99-111. ISSN: 13272314. Disponivel em: <http://search.ebscoho

Boehm, B., Valerdi, R., Lane, J. and Brown, A. 2005. Cocomo suite methodology and evolution.: CrossTalk 18(4), 20-25.

Carvalho, R., and A. J. M. 2011. Tomadas de Decisão com Ferramentas da Lógica Paraconstistente Anotada.: São Paulo: Edgard Blucher Ltda.

Clive, D. Wrigley and Albert S. Dexter. 1991. Design science in information systems research., MIS Q. 15,245-257.

Costa, N. C. A., Abe, J.M., Silva Filho, João Inácio da, Leite, C. F. S. and Murolo, A. 1999. Lógica Paraconsistente Aplicada.: 1.ed. São Paulo: Atlas, v. 1. p.214.

IFPUG, CPM 4.3.1, 2010. Unadjusted Functional Size Measurement Method – Counting Practices Manual, International Organization for Standardization, ISO/IEC 20926.

Jones, Capers., 2010. Software Engineering Best Practices.: McGraw Hill.

LeszekMaciaszek and Kang Zhang. 2015. ENASE (Conference), Evaluation of novel approaches to software engineering: 9th International Conference.; Lisbon, Portugal, April 28-30, 2014, Revised selected papers.

PMI. 2017. Project Management Institute, Guide PMBOK®- 6ed; Project Management Institute Inc.

Pressman, Roger S. 2011. Software engineering: a practitioner's approach, 7th Edition.ISBN 0073375977 / 9780073375977 ©: The McGraw-Hill Companies, Inc., New York, NY, EUA.

Vazquez, C. E. , Simões, G. S. and Albert, R. M. 2010. Análise de Pontos de Função - Medição, Estimativas e Gerenciamento de Projetos de Software. 9. ed., São Paulo: Érica.