Improving the Process of Evaluating User Stories Using the Paraconsistent Annotated Evidential Logic Eτ

Samira Sestari Do Nascimento, Jair Minoro Abe, Luiz Roberto Forçan, Cristina Corrêa de Oliveira, Kazumi Nakamatsu and Ari Aharari

July 16, 2022

Improving the Process of Evaluating User Stories using the Paraconsistent Annotated Evidential Logic Eτ

Samira Sestari do Nascimento[1[0000-0001-7429-6033]], Jair Minoro Abe[1[0000-0003-2088-9065]], Luiz Roberto Forçan[1[0000-0001-7376-830x]], Cristina Corrêa de Oliveira[2[0000-0002-8629-6679]], Kazumi Nakamatsu[3], Aharari Ari[4]

[1] Paulista University, São Paulo, Brazil
[2] Federal Institute of São Paulo, Bragança Paulista, Brazil
[3]University of Hyogo, Kobe, Japan
[4]Sojo University, Kumamoto, Japan

samira.nascimento3@aluno.unip.br,jair.abe@docente.unip.br, luforçan@gmail.com , crisolive@ifsp.edu.br, akamatu@pf7.so-net.ne.jp,aharari@cis.sojo-u.ac.jp

Abstract. Software developers need to be agile to meet users' needs, delivering software on tight, quality deadlines. User history is a technique used in agile methods to elicit requirements. However, this process is performed with the developers and the user, and there may be contradictions between them, resulting in inaccurate metrics. This article presents a model of validation of user history using the Para-analyzer algorithm, based on the Paraconsistent Annotated Evidential Logic Eτ to assist in improving the evaluation, prioritization, and estimation process of user stories.

A survey was conducted with a team of developers working with agile methods. The model uses the degrees of favourable and contrary evidence for each INVEST criterion as input variables. The application of this model allows considering extremely relevant issues when it comes to supporting decision-making based on a mathematical model and serving as a support tool for teams, Product Owners, Project Managers, and others. Four user stories were analyzed by nine experts, who evaluated the criteria for each user story. The interpretation of the evaluations performed by the experts was through the global analysis in the unit square of the Cartesian plane, which indicated the degrees of favourable evidence and contrary evidence for the data used. Two stories that could not be developed in a Sprint were verified and, therefore, should be refactored and resubmitted to the opinion of experts. The other two stories had favourable evidence to be used in a Sprint.

Keywords: User Stories, Paraconsistent Annotated Evidential Logic Eτ, INVEST.

1 Introduction

A requirement is defined as a role, service, or resource to meet a product user's need or demand, as defined by Requirements Engineering, a software engineering subarea. It can be divided into functions, constraints, or business rules regardless of the methodology adopted [1].

In an agile development, user requirements are treated as User Stories (US), which are written requests from the user from the point of view of needs and use [2].

Tasks are prioritized from an available list of software called Product Backlog. The development team (TIME) commits to advancing the cycle in software development, lasting four weeks to run [3].

Due to the constant involvement of the client in the validation and acceptance tests of USs, it becomes a frequent need [4], requiring business knowledge and the domain of the application where the system will be used (Heikkiläetal., 2015). Although it does not have a specific input for the validation process, it is essential to narrow the organizational strategy to the corporate culture (Schwaber & Sutherland, 2020). In the validation step, the developer evaluates whether the USs understand, are absent, or replay information to start a sprint, with the certainty that the issues will be resolved by the product owner (PO) or by the customer himself. The project's success depends on the sprint plan, which verifies the quality of the Time and PO, and USs to ensure requirements are correct, documented, and validated [5].

However, the complexity of USs and time determination are not clearly defined and described, and it is considered risky in the project development plan to understand [6] [7]. Another aspect is the collective consensus. Because the analysis of each US is individual, it requires knowledge of the application and the domain of the business and can lead to inconsistencies in a study among the planning participants [4].

The validation process needs to generate a list of issues and actions agreed upon by the client, PO, and TIME. The main problem with this process is that it does not meet the INVEST criteria. It is an acronym for "Independent, Tradeable, Valuable, Estimable, Small or Small, and Testable Stories" [8]. From this list of problems, it needs to create an action plan to define the overall work plan to be executed and agreed upon by all those involved. The various paths of understanding to arrive at results can upset the plan. A Paraconsistent Annotated Evidential Logic Eτ (logic Eτ) evaluates uncertain. Inconsistent data that stakeholders in certain USs do not understand may express a logical contradiction [9], considering the criteria to be logic Eτ assists in decision-making [10].

In this article, the questionnaire results were applied to software development experts in different functions within the company, evaluating USs and measuring evidence through the logic Eτ in conjunction with the INVEST criteria. The model of this study assists in the decision-making process by permitting technical validation and through Para-analyzer algorithms that will be represented in the final analysis of USs.

2 Reasons

2.1 User Stories

User stories (US), in agile development, are a simple way to disbelieve the needs of the product owner because they capture the essential elements using the most widespread format. "As a <role>, I want <goal/desire> so that <benefit>" [11].

They express functional requirements using a stakeholder business language because they represent an agreement between the PO and the developers, forming the basis for development. After all, TIME must understand, estimate, and implement all US [12].

2.2 INVEST

The INVEST criterion is an acronym for <I>ndependent, <N>egotiable, <V>aluable, <E>stimable, <S>mall, <T>estable, which should be applied for each US [8], representing an agreement between the PO and TIME to decide whether the US will be assigned in a Sprint.

The main problems encountered in this agreement are communication failures between TIME and the PO [13], [14], and understanding of INVEST [15] from incomplete requirements; formulated or intestable; provided late (scope increase), or not detailed enough to express the smallest significant unit of activity for the user.

Subjectivity in US estimates, which are usually experimentally derived and managed, often without reference to the historical data of the para comparison organization; lack of experience in estimating; or even lack of knowledge of the domain [16], may result in a lack of understanding of the criteria, making it difficult to estimate, as there is no clarity of the outputs that the system should produce and the inputs are not objectively measured [17]

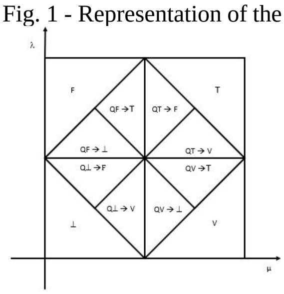
Time estimates for the completion of each task, defined as story points, and for the project total [18] are carried out with the definition of all investment criteria for each US. Although the estimate is important from a commercial point of view, the film is about minimizing errors or failures because the scope of the software project is estimated along with the time and cost [19].

2.3 Paraconsistent Annotated Evidential Logic Eτ

The Paraconsistent Annotated Evidential Logic Eτ belongs to the paraconsistent logic class, not classical, considering the principle of contradiction by obtaining contrary (P) propositions, which are associated with an atomic request, the type of Degree of Favorable Evidence (μ), and the type of Unfavorable Degree of Evidence (λ). The pair (μ, λ), called the annotation constant, where μ and λ [0, 1] e (P) denote a proposition in the usual sense. The annotation μ indicates the degree of favourable evidence, and the λ annotation represents the unfavourable evidence expressed by request (P) τ∈. Logical states are called extremes and are characterized by internal states, as illustrated in **Erro! Fonte de referência não encontrada.**. Non-extreme states are named according to their proximity to extreme logical conditions, as shown in Tab. 1.

The Para-analyzer algorithm consists of information collected through a research form for decision-making analysis [20]. In data processing, the connectives express favourable and unfavourable opinions about the propositions from the experts participating in the decision-making process. Operators (OR) and (AND) correspond to disjunction and conjunction in classical logic. For example, in Logic Eτ, maximization is considered to be (μ 1, λ1) OR (μ 2, λ2) = (Max {μ 1, μ 2}; Min {λ1, λ2}) where Max indicates the maximization of real numbers with the standard order and Min indicates minimizing the actual numbers with a legal order. On the other hand, operation AND is described as (μ 1, λ1) e (μ 2, λ2) = (Min {μ 1, μ 2}; Max {λ1, λ2}) where Min indicates the working of minimizing real numbers with standard order and Max indicates an operation maximizing real numbers with standard order.

Fig. 1 - Representation of the Lattice τ



Source: [1]

Tab. 1 - Symbolization of logical states

Extreme States	Symbol	Non-Extreme States	Symbol
True	V	Quasi-true tending to Inconsistent	$Q_V \rightarrow T$
False	F	Quasi-true tending to Paracomplete	$Q_V \rightarrow \perp$
Inconsistent	T	Quasi-false tending to Inconsistent	$Q_F \rightarrow T$
Paracomplete	⊥	Quasi-false tending to Paracomplete	$Q_F \rightarrow \perp$
		Quasi-inconsistent tending to True	$Q_T \rightarrow V$
		Quasi-inconsistent tending to False	$Q_T \rightarrow F$
		Quasi-paracomplete tending to True	$Q_\perp \rightarrow V$
		Quasi-paracomplete tending to False	$Q_\perp \rightarrow F$

3 Case Study

For simplicity, this study is based on a backlog with only four US; selected requirements gathering is carried out in companies. Therefore, they were removed from previous projects, and the respondents were unaware of them.

Tab. 2 - User Stories

US	User Story	Acceptance Criteria
A	As a Customer, I want to know the price and availability of a product to verify that I want to complete the purchase.	The product must have the name and price. The product must be associated with a category and a supplier.
B	As an Administrator, I want to query product categories to view product categories.	Do not list categories that are discontinued.
C	As an administrator, I would want to search for products by category; I know the category code and would like to locate it in the system. to learn about your items and to see the products in a well-informed category.	Do not list products that are discontinued. You must list all products even if they are out of stock.
D	As a customer, I want the system to provide several forms of payment so that I can pay for my order and the system close the order.	Customers can select a payment method to be able to pay. When the payment is made with a credit card, validate the operation or restriction with the operator. If everything is ok, proceed with the finalization of the order.

Source: Authors

Nine participants were selected who are professionals in software development, referred to as “specialists,” who work specifically with the Scrum methodology. These professionals from different software companies in the city of São Paulo were divided into three groups, each adopting the position as a grouping criterion. They are called E1, E2,.., and E9 in the database.

The specialists are represented by the letters E1, E2, and E3. The positions are for specialist E1 (Software Architect and Scrum Master), specialist E2 (Senior System Analyst), and specialist E3 (Full System Analyst).

For the INVEST criteria, they were mapped to F1 (independent) factors; F2 (negotiable); F3 (valuable); F4 (estimable); F5 (small); and F6 (testable). Each expert expressed their opinions on each INVEST criteria, pointing to μ and λ criteria for each US. The values of μ and λ were normalized, according to Tab. 3.

Tab. 3 Normalization of the values of μ and λ

Degree	Per cent (%)	Description
1,00	100	There is no doubt about the evidence
0,75	75	Small doubt regarding evidence
0,50	50	Average doubt in relation to evidence
0,25	25	Low certainty in relation to evidence
0,00	0	Almost no certainty regarding evidence

Source: Authors

Given the reponses of experts, who attributed the values of μ and λ for each US, according to Tab. 4, a database was developed with the factors analyzed.

Tab. 4 Database formed by assigned by experts analysis

	Group A						Group B						Group C					
	E1		E2		E3		E4		E5		E6		E7		E8		E9	
Factor	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ
F1	0,00	1,00	0,25	1,00	0,50	1,00	0,25	1,00	0,25	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,25	1,00
F2	1,00	0,00	1,00	0,25	1,00	0,25	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,25	1,00	0,25	1,00	0,25
F3	1,00	0,00	1,00	0,25	1,00	0,25	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,25
F4	0,00	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,00	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,25	1,00
F5	0,00	1,00	0,00	1,00	0,25	1,00	0,25	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,25	0,75	0,25	0,75
F6	0,00	1,00	1,00	0,50	0,75	0,25	1,00	0,25	1,00	0,50	1,00	0,25	1,00	0,50	1,00	0,25	0,75	0,25

Source: Authors

4 Results

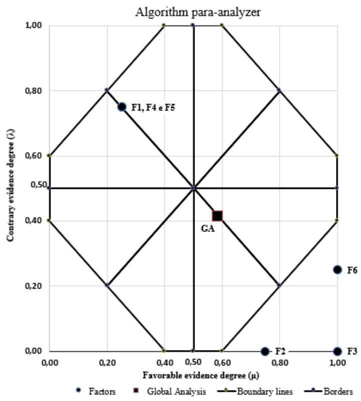
After applying the Para-analyzer algorithm, each track receives a diagnosis, which can be: feasible (which is understood as approved); unfeasible (the item is in disagreement with the evaluation factor); Non-Conclusive (the thing requires further evaluation). For each user story, the following results were obtained:

User Story A

The result of the Global Analysis (GA) of the Para-analyzer algorithm (0.58; 0.42) indicates that it is in the "Quasi-True Tending to Inconsistent" state, resulting in no conclusion. The paraconsistent qualitative evaluation demonstrates that US's excellent quality presentation is no longer an absolute truth. Therefore, it is assumed that its quality is insufficient, requiring that new information be obtained and submitted again for analysis by the specialists, as shown in Fig. 2.

Fig. 2 - Analysis of User a Story A

Factor	μ and λ	Decision
F1	(0,25; 0,75)	not conclusive
F4	(0,25; 0,75)	not conclusive
F5	(0,25; 0,75)	not conclusive
F2	(0,75; 0)	feasible
F3	(1,00; 0)	feasible
F6	(1,00; 0,25)	feasible



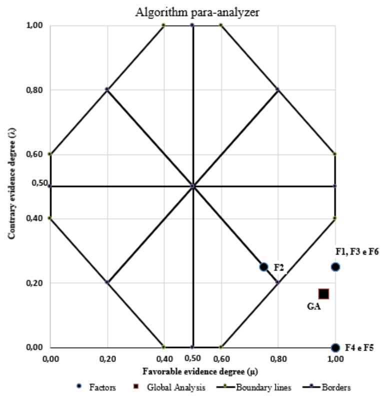
Source: Authors

User Story B

The result of the GA of the Para-analyzer algorithm (0.96; 0.17) indicates that it is in the "Totally True" state, resulting in viability. Furthermore, the paraconsistent qualitative evaluation demonstrates that US presents an adequate quality. Therefore, it is understood that this US is within a possible quality standard and may be part of a Sprint, as seen in Fig. 3.

Fig. 3 - Analysis of User a Story B

Factor	μ and λ	Decision
F1	(1,00; 0,25)	feasible
F3	(1,00; 0,25)	feasible
F4	(1,00; 0)	feasible
F5	(1,00; 0)	feasible
F6	(1,00; 0,25)	feasible
F2	(0,75; 0,25)	not conclusive



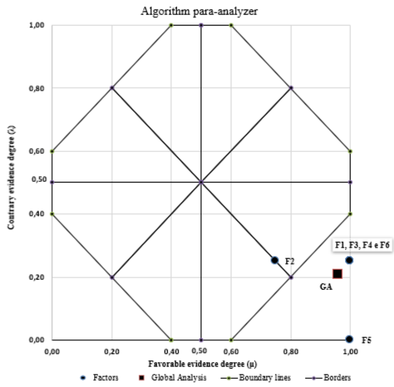
Source: Authors

User Story C

The result of the GA of the Para-analyzer algorithm (0.96; 0.21) indicates that it is in the "Totally True" state, resulting in a viable. The paraconsistent qualitative evaluation demonstrates that US is feasible. It is understood that this US is within a quality standard and can be part of a sprint to conform to Fig. 4.

Fig. 4 - Analysis of User a Story C

Factor	μ and λ	Decision
F1	(1,00; 0,25)	feasible
F3	(1,00; 0,25)	feasible
F4	(1,00; 0,25)	feasible
F5	(1,00; 0,00)	feasible
F6	(1,00; 0,25)	feasible
F2	(0,75; 0,25)	not conclusive



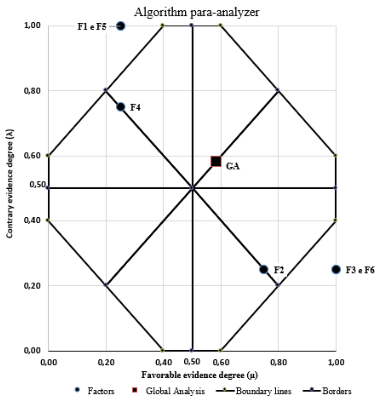
Source: Authors

User Story D

The result of the GA of the Para-analyzer algorithm (0.59; 0.59) indicates that it is in the "Inconsistent" state, tending to falsity present inconclusive results. The paraconsistent qualitative evaluation demonstrates that it is no longer an absolute truth to conclude that this US has a viable quality. It is understood that this US depends on others and is very large. Therefore, it is assumed that the quality of this US is insufficient, requiring that new information be obtained and submitted for analysis by the specialists.

Fig. 6 - Analysis of User a Story D

Factor	μ and λ	Decision
F1	(0,25; 1,00)	infeasible
F5	(0,25; 1,00)	infeasible
F3	(1,00; 0,25)	feasible
F6	(1,00; 0,25)	feasible
F2	(0,75; 0,25)	not conclusive
F4	(0,25; 0,75)	not conclusive



Source: Authors

5 Conclusion

Current US validation processes do not take into account inconsistency or contradiction. In an actual situation, contradictions appear due to the conditions of the environment in which the requirements were surveyed. These situations of contradiction occur regardless of the will of the software development team members or the company's business area team. Therefore, the conflicts are part of the US validation. The bigger involvement of stakeholders, customers, Product Owners, and the development team to resolve the client's interests needs, and desires, bigger the levels of conflicts, contradictions, and inconsistencies.

The research presents a model to assist in the decision-making process of evaluation of USs using the Logic Eτ, through the Para-analyzer algorithm, using logical criteria that enable technical validation. The input parameters are established by the experts' opinions, consolidating a collective logic of TIME based on mathematical terms.

The analysis based on these perspectives satisfies experts and stakeholders because the paraconsistent model maximizes all expert opinions and creates a mathematical consensus on these opinions.

This research demonstrates that the concepts of Evidence Noted Paraconsistent Logic Eτ could be used to validate and present perspectives on dealing with situations of uncertainty and inconsistency. These situations or factors are relevant for decision-making and influence the quality and success of an information system.

Bibliography

[1] R. S. Pressman, Software engineering: a practitioner's approach, 7 ed., McGraw-Hill Publishing Co., 2011.

[2] M. R. M. Chopade and N. S. Dhavase, "Agile software development: Positive and negative user stories," in 2017 2nd International Conference for Convergence in Technology (I2CT), 2017.

[3] R. Barbosa, A. E. A. Silva and R. Moraes, "Use of Similarity Measure to Suggest the Existence of Duplicate User Stories in the Srum Process," in 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W), 2016.

[4] V. T. Heikkilä, D. Damian, C. Lassenius and M. Paasivaara, "A Mapping Study on Requirements Engineering in Agile Software Development," in 2015 41st Euromicro Conference on Software Engineering and Advanced Applications, 2015.

[5] A. Mateen, K. Abbas and M. A. Akbar, "Robust approaches, techniques and tools for requirement engineering in agile development," in 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017.

[6] J. E. Hannay, H. C. Benestad and K. Strand, "Agile Uncertainty Assessment for Benefit Points and Story Points," IEEE Software, vol. 36, pp. 50-62, 7 2019.

[7] R. Popli and N. Chauahn, "Managing uncertainty of story-points in Agile software," in 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015.

[8] L. Buglione and A. Abran, "Improving the User Story Agile Technique Using the INVEST Criteria," in 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, 2013.

[9] F. R. Carvalho and J. M. Abe, A Paraconsistent Decision-Making Method, vol. 87, Springer International Publishing, 2018.

[10] J. M. Abe, Paraconsistent Intelligent-Based Systems - New Trends in the Applications of Paraconsistency, vol. 94, Springer, Ed., Switzerland: Springer, 2015, p. 306.

[11] J. Sutherland, Scrum: The Art of Doing Twice the Work in Half the Time, 2 ed., São Paulo: Crown Business Publications, 2015.

[12] F. Dalpiaz and S. Brinkkemper, "Agile Requirements Engineering with User Stories," in 2018 IEEE 26th International Requirements Engineering Conference (RE), 2018.

[13] A. Hess, P. Diebold and N. Seyff, "Towards Requirements Communication and Documentation Guidelines for Agile Teams," in 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), 2017.

[14] O. Liskin, R. Pham, S. Kiesling and K. Schneider, “Why We Need a Granularity Concept for User Stories,” 2014.

[15] J. G. Opazo and C. Lopez, “Improving User Stories: A Case Study in the Chilean Banking Industry,” in 2018 XLIV Latin American Computer Conference (CLEI), 2018.

[16] P. Kamthan and N. Shahmir, “Modeling Negative User Stories is Risky Business,” in 2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE), 2016.

[17] N. Bik, G. Lucassen and S. Brinkkemper, “A Reference Method for User Story Requirements in Agile Systems Development,” in 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), 2017.

[18] F. E. Castillo-Barrera, M. Amador-Garcia, H. Perez-Gonzalez and F. E. Martinez-Perez, “Agile Evaluation of the Complexity of User Stories Using the Bloom’s Taxonomy,” in 2017 International Conference on Computational Science and Computational Intelligence (CSCI), 2017.

[19] F. Hayat, A. U. Rehman, K. S. Arif, K. Wahab and M. Abbas, “The Influence of Agile Methodology (Scrum) on Software Project Management,” in 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2019.

[20] S. Akama, Ed., Towards Paraconsistent Engineering, vol. 110, Springer International Publishing, 2016.

[21] H. Sheemar and G. Kour, “Enhancing User-Stories Prioritization Process in Agile Environment,” in 2017 International Conference on Innovations in Control, Communication and Information Systems (ICICCI), 2017.