

**UNIVERSIDADE PAULISTA**

**APLICAÇÃO DA LÓGICA PARACONSISTENTE ANOTADA  
EVIDENCIAL  $E_{\tau}$  EM ENGENHARIA DE REQUISITOS E  
INTERFACE HOMEM-COMPUTADOR**

Dissertação apresentada ao Programa de  
Pós-Graduação em Engenharia de  
Produção da Universidade Paulista –  
UNIP, para obtenção do título de Mestre  
em Engenharia de Produção.

**LUIZ ROBERTO FORÇAN**

**SÃO PAULO**

**2021**

**UNIVERSIDADE PAULISTA**

**APLICAÇÃO DA LÓGICA PARACONSISTENTE ANOTADA  
EVIDENCIAL  $E_{\tau}$  EM ENGENHARIA DE REQUISITOS E  
INTERFACE HOMEM-COMPUTADOR**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Paulista – UNIP, para obtenção do título de Mestre em Engenharia de Produção.

**Orientador:** Prof. Dr. Jair Minoro Abe

**Área de Concentração:** Gestão de Sistemas de Operação.

**Linha de pesquisa:** Métodos Quantitativos em Engenharia de Produção

**Projeto de Pesquisa:** Processos Decisórios Baseados em Lógicas Não Clássicas

**LUIZ ROBERTO FORÇAN**

**SÃO PAULO**

**2021**

Forçan, Luiz Roberto.

Aplicação da lógica paraconsistente evidencial E em engenharia de requisitos e interface homem-computador / Luiz Roberto Forçan. - 2021.

99 f. : il. + CD-ROM.

Dissertação de Mestrado Apresentada ao Programa de Pós Graduação em Engenharia de Produção da Universidade Paulista, São Paulo, 2021.

Área de concentração: Gestão de Sistemas de Operação.

Orientador: Prof. Dr. Jair Minoro Abe.

1. Scrum. 2. Lógica paraconsistente anotada evidencial E. 3. Tomada de decisão. 4. Escala de usabilidade do sistema.  
I. Abe, Jair Minoro (orientador). II. Título.

**LUIZ ROBERTO FORÇAN**

**APLICAÇÃO DA LÓGICA PARACONSISTENTE ANOTADA  
EVIDENCIAL  $E_{\tau}$  EM ENGENHARIA DE REQUISITOS E  
INTERFACE HOMEM-COMPUTADOR**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Paulista – UNIP, como requisito parcial para obtenção do título de Mestre em Engenharia de Produção.

Aprovado em: \_\_\_\_/\_\_\_\_/\_\_\_\_

**BANCA EXAMINADORA**

---

Prof. Dr. Jair, Minoru Abe  
Orientador / Universidade Paulista

---

Prof. Dr. Pedro Luiz de Oliveira Costa Neto  
Universidade Paulista

---

Profa. Dra. Cristina Corrêa de Oliveira  
Instituto Federal de São Paulo



## **DEDICATÓRIA**

Dedico este trabalho a Deus, pela força, inspiração e iluminação e às seguintes pessoas: minha esposa, Francisca Valnete de França Forçan; minha filha, Isabela Helena França Forçan; minha irmã Sônia Regina Godinho de Lara, por terem me apoiado e compreendido e entenderem minha ausência durante essa jornada.

## **AGRADECIMENTOS**

Agradeço, em primeiro lugar, a DEUS, que colocou, em meu caminho, as pessoas certas, para me ajudarem a alcançar mais um objetivo.

Agradeço, de modo especial, a meus pais, que sempre me incentivaram a estudar e a me desenvolver e me ensinaram a importância de construir e amar uma família.

Agradeço ao meu orientador, Professor Dr. Jair Minoro Abe, pelo apoio e ensinamentos, que tanto contribuíram para a realização deste trabalho.

Agradeço à Professora Cristina Correa de Oliveira, pela excepcional ajuda, dedicação, comprometimento e apoio com que norteou este trabalho, facilitando encontrar o caminho mais adequado.

Agradeço ao Professor Dr. Pedro Luiz de Oliveira Costa Neto, pela excelente qualidade das aulas. Seus conhecimentos sobre Engenharia de Produção ampliaram meu repertório, valorizando ainda mais esta jornada.

Agradeço à Professora Dra. Irenilza de Alencar Nääs, pela ajuda, dedicação, demonstração de apoio e comprometimento com este trabalho, que tornaram esta jornada mais suave.

Agradeço ao Professor Dr. João Gilberto Mendes dos Reis, pela excepcional ajuda. Sem seu apoio, teria sido muito mais difícil vencer esse desafio.

Agradeço ao Professor Dr. Oduvaldo Vendrametto, pela gestão do Programa de Pós-Graduação e por compartilhar suas visões, ampliando as minhas.

Faço um agradecimento especial a todos colegas de mestrado e professores, em especial, a Samira Sestari do Nascimento, pelas inestimáveis contribuições.

Por fim, meu agradecimento a todos colaboradores e facilitadores do curso de Pós-Graduação em Engenharia de Produção da UNIP, em especial, a senhora Márcia Nunes, pelo eficiente auxílio que nos prestou nas questões administrativas.

## EPÍGRAFE

*“Toda teoria é cinza e só é verde a árvore de dourados frutos que é a vida” (Goethe).*

## RESUMO

O presente estudo teve por objetivo analisar a aplicação da Lógica Paraconsistente Evidencial  $E_{\tau}$  como apoio na tomada de decisão, em duas subáreas do processo de desenvolvimento de software: Engenharia de Requisitos e Interface Homem Computador. Dentro da subárea Engenharia de Requisitos, os métodos ágeis utilizam, como técnica para elicitar requisitos, a História de Usuário. Entretanto, desenvolvedores e usuários têm dificuldades para entender essas Histórias, o que pode gerar, entre eles, contradições que resultam em métricas inexatas. Este estudo apresenta um modelo de validação da História do Usuário que utiliza o algoritmo Para-analisador, a fim de contribuir para o aperfeiçoamento do processo de avaliação, estimativa e priorização dessas Histórias. Para chegar a esse modelo, desenvolveu-se uma pesquisa qualitativa, aplicando um *survey*, para avaliar Histórias de Usuário obtidas de um time Scrum em uma empresa de tecnologia. O modelo proposto contempla, como fator de análise, o conceito INVEST: Independente, Negociável, Valiosa, Estimável, Pequena e Testável. O estudo apresenta o resultado da análise dos dados, com base na opinião emitida pelos especialistas. Foi constatado que duas histórias possuíam grau de evidências favoráveis satisfatórias para serem trabalhadas em um Sprint. Porém, conclui-se que nenhuma delas podia ser desenvolvida em um Sprint, devendo, portanto, ser enviada para reavaliação dos especialistas. Dentro da subárea Interface Homem Computador, a opinião da experiência do usuário com determinado produto de software é afetada por sua usabilidade, semelhante à experiência do usuário (UX) que determina se um aplicativo atende às expectativas. Assim, a avaliação da usabilidade é necessária, para melhorar a qualidade do software. A Escala de Usabilidade do Sistema afere a experiência do usuário com o sistema, apresentando 10 questões que se contradizem. O presente trabalho tem como evidenciar um detalhamento das contradições, utilizando a Lógica Paraconsistente Anotada Evidencial  $E_{\tau}$ , por meio da aplicação de um *survey*, em onze profissionais que aferiram um sistema com o qual trabalham. Foi elaborada uma confrontação dos resultados do índice da escala SUS com o algoritmo Para-analisador, aplicando-se a constante de anotação ( $\mu$ ;  $\lambda$ ) da Lógica  $E_{\tau}$ , classificando as questões pares para  $\mu$  e as questões ímpares para  $\lambda$ . O SUS apresentou um resultado com a Escala de Usabilidade do Sistema (76,59), significando uma avaliação positiva da usabilidade. O resultado do Para-analisador exibiu a média global (0,75; 0,25),

apontando um resultado similar ao SUS e, ainda, evidenciando dois itens contraditórios. A avaliação da usabilidade foi aperfeiçoada, demonstrando que os respondentes concordaram em nove atributos da usabilidade. Entretanto, ocorreu uma discordância em um item, uma vez que cinco respondentes atribuíram valores altos a duas questões contraditórias, indicando uma contradição na avaliação. Este resultado pode apontar falta de entendimento do significado das questões do instrumento, o que poderia indicar a necessidade de capacitar os participantes antes do começo do processo. A contribuição principal deste trabalho foi utilizar o Para-analisador junto com a avaliação da usabilidade de um software, pois essa condição permitiu uma análise mais criteriosa, já que ele trabalha com o par  $(\mu; \lambda)$ , o que possibilita aferir os resultados em duas dimensões.

**Palavras chaves:** Scrum. Lógica Paraconsistente Anotada Evidencial  $E_{\tau}$ . Tomada de Decisão. Escala de Usabilidade do Sistema.

## ABSTRACT

This study aimed to analyze the application of the Evidential Paraconsistent Logic  $E_{\tau}$  to support decision making in two sub-areas of the software development process: Requirements Engineering and Human Computer Interface. Within the Requirements Engineering sub-area, agile methods use the User Story as a technique to elicit requirements, however developers and users have difficulties to understand these Stories, and there may be a lot of contradiction between them, resulting in inaccurate metrics. This study presents a User History validation model using the Para-analyzer algorithm, to contribute to the improvement of the evaluation, estimation and prioritization process of these Stories. To arrive at this model, a qualitative research was developed using a survey to evaluate User Stories obtained from a Scrum team in a technology company. The proposed model includes as analysis factors the INVEST concept: Independent, Negotiable, Valuable, Estimable, Small and Testable. This study presents the result of data analysis based on the opinion issued by experts. It was found that two stories had a satisfactory degree of favorable evidence to be worked on in a Sprint. However, it is concluded that two stories could not be developed in a Sprint and, therefore, should be sent for re-evaluation by experts. Within the Human Computer Interface subarea, the opinion of the user's experience with a given software product is affected by its usability, as the user experience (UX) determines whether an application meets users' expectations. So the usability assessment is necessary to improve the quality of the software. The System Usability Scale measures the user's experience with the system, presenting 10 contradictory questions. The present work has as evidence a detail of the contradictions, using the Evidential Annotated Paraconsistent Logic  $E_{\tau}$ , through the application of a survey to eleven professionals who measured a system with which they work. A comparison of the SUS scale index results with the Para-analyzer algorithm was elaborated, applying the pair  $(\mu; \lambda)$ , classifying the even questions for  $\mu$  and the odd questions for  $\lambda$ . The result of the SUS showed a positive evaluation of usability, with the System Usability Scale (76.59). The result of the Para-analyzer showed the global average (0.75; 0.25), indicating a result similar to the SUS and also evidencing two contradictory items. The usability assessment was improved, showing that respondents agreed on nine usability attributes. However, there was a disagreement in one item, since five respondents attributed high values to two contradictory questions, indicating a contradiction in the

assessment. This result may point to a lack of understanding of the meaning of the instrument's questions, which could indicate the need for training of participants before the beginning of the process. The main contribution of this work was to use the Para-analyzer, together with the evaluation of the usability of a software, allowing for a more careful analysis, as it works with the pair  $(\mu; \lambda)$ , making it possible to measure the results in two dimensions.

Keywords: Scrum. Paraconsistent Annotated Evidential Logic  $E_{\tau}$ . Decision Making. System Usability Scale.

## UTILIDADES

Este trabalho contribui para a sociedade nas áreas de conhecimento relacionadas à produção e organização de software, qualidade, redução de custos e desperdícios.

Também evidenciou a importância do uso da Lógica  $E\tau$  como aplicação no auxílio no processo de tomada de decisão em duas subáreas da Engenharia de Software: Engenharia de Requisitos (ER) e Interface Homem Computador (IHC).

Na primeira subárea, apresenta um modelo de validação da História do Usuário, utilizando o algoritmo Para-analisador, baseado na Lógica  $E\tau$ , a fim de ampliar o conhecimento sobre processo de priorização, estimativa e avaliação das Histórias de Usuários. Para cada critério INVEST, o modelo utiliza, como variáveis de entrada, graus de evidência favorável e graus de evidência desfavorável. Sua aplicação permite tomadas de decisão com base em um modelo matemático e pode ser usada como instrumento de suporte para os desenvolvedores de software, Gerente de Projetos, *Product Owner* e outros.

Na subárea Interface Homem Computador (IHC), a principal contribuição desta pesquisa foi avaliar a usabilidade de um software em conjunto com o Para-analisador da Lógica  $E\tau$ , que possibilita análise mais minuciosa das opiniões dos especialistas, sendo possível avaliar os resultados em duas dimensões, ampliando o conhecimento sobre a análise da usabilidade.

**Luiz Roberto Forçan**

**Orientador:** Prof. Dr. Jair Minoro Abe



## LISTA DE ILUSTRAÇÕES E QUADRO

Figura 1 - Cerimônias do Scrum.....	15
Figura 2 - Apresentação do Reticulado $\tau$ .....	33
Figura 3 - Sistema de Análise Paraconsistente.....	35
Figura 4 - Graus de Certeza e Incerteza .....	37
Figura 5 - Amostra da Base de dados formada por $\mu$ e $\lambda$ da História de Usuário A ..	42
Quadro 1 - Fatores e os critérios do INVEST .....	40
Quadro 2 - Histórias de Usuário .....	42

## LISTA DE TABELAS

Tabela 1 - Estados Extremos.....	33
Tabela 2 - Simbolização dos Estados Não-Extremos .....	34
Tabela 3 - Classificação dos Grupos de Especialistas .....	39
Tabela 4 - Normalização dos valores de $\mu$ e $\lambda$ .....	41
Tabela 5 - Normalização de $\mu$ e $\lambda$ .....	43
Tabela 6 - Classificação dos Grupos de usuários.....	44
Tabela 7 - Base de dados formada $\mu$ e $\lambda$ atribuídos pelos usuários .....	44

## LISTA DE ABREVIATURA E SIGLAS

$G_{ce}$ : Grau de Certeza

$G_{in}$ : Grau de Incerteza

## LISTA DE SÍMBOLOS

$\mu$ : Letra grega Mi, grau de evidência favorável.

$\lambda$ : Letra grega Lambda, grau de evidência desfavorável.

$\tau$ : Letra grega Tau, reticulado associado à Lógica  $E\tau$ .

F: Estado Lógico Falso.

T: Estado Lógico Inconsistente.

V: Estado Lógico Verdadeiro.

$\perp$ : Estado Lógico Paracompleto.

## SUMÁRIO

1	INTRODUÇÃO .....	14
1.1.	Apresentação .....	14
1.2.	Justificativa.....	16
1.3.	Objetivo Geral .....	18
1.3.1.	Objetivos Específicos .....	18
1.4.	Estrutura do Trabalho .....	19
2	REFERENCIAL TEÓRICO .....	21
2.1.	Metodologia Ágil.....	21
2.2.	Scrum Framework de Gerenciamento de Projetos .....	22
2.3.	As contradições no desenvolvimento de software .....	27
2.4.	Lógica Paraconsistente Anotada Evidencial $E\tau$ .....	31
3	METODOLOGIA .....	39
3.1.	Engenharia de Requisitos: Aplicação da Lógica $E\tau$ na avaliação da História de Usuário .....	39
3.2.	Metodologia Aplicada na Avaliação da Usabilidade de Software.....	43
4	Resultados e discussão .....	45
4.1	Artigo 1 .....	45
4.2	Artigo 2 .....	45
4.3	Artigo 3 .....	48
5	CONSIDERAÇÕES FINAIS.....	49
6.1.	Sugestões de Trabalhos Futuros .....	51
	REFERÊNCIAS.....	52
	APÊNDICES.....	59
	Apêndice 1 .....	59
	Apêndice 2 .....	68
	Apêndice 3 .....	85

## 1 INTRODUÇÃO

Este capítulo destina-se a introduzir os conceitos de Engenharia de Software utilizados neste trabalho, que contribuíram para o desenvolvimento da pesquisa, a fim de aprimorar duas etapas do processo de desenvolvimento de software, com o apoio da Lógica Paraconsistente Anotada Evidencial  $E\tau$ .

### 1.1. Apresentação

Na competitiva indústria de software, a exigência do mercado por qualidade vem crescendo com a mesma rapidez que a rejeição de softwares de baixa qualidade. (KABIR; REHMAN; MAJUMDAR, 2016).

A qualidade de um software encontra-se diretamente ligada à qualidade do processo de desenvolvimento (SWEBOK, 2014). A Engenharia de Requisitos (ER) de Software (SITTHITHANASAKUL; CHOOSRI, 2016; LORCA; BURROWS; STERLING, 2018), que pertence à grande área da Engenharia de Software, preocupa-se com o chamado triângulo de ferro - prazo, custo e escopo do projeto de software (HAYAT *et al.*, 2019). A ER é composta por Concepção, Elicitação de Requisitos, Elaboração, Especificação, Negociação, Validação e Gerenciamento de Requisitos (PRESSMAN, 2011). Requisito é uma função, serviço ou recurso, para atender uma necessidade ou demanda do usuário de um produto, sendo denominado requisito funcional. (PRESSMAN, 2011). Os requisitos não funcionais são restrições impostas ao software, como tempo de resposta, tempo de processamento, segurança e usabilidade (25000, 2014; VILLAMIZAR *et al.*, 2019).

Categoriza-se o modelo de processo de software em dois grandes grupos: a cascata e o modelo iterativo e incremental (KOMAI; NAKANISHI; SAIDI, 2017), Denominados, no mercado, métodos Ágeis. Eles vêm sendo utilizados pela indústria de software. O método de gerenciamento de projetos mais utilizado, no mercado, é o Scrum (SHARMA; HASTEER, 2016), que se caracteriza como um processo leve, composto por cinco cerimônias, conforme mostra a Figura 1: Lista com as Funcionalidades do Produto (*Product Backlog*); Reunião de Planejamento do Sprint (*Sprint Planning Meeting*); Reunião de Quinze Minutos em Pé (*Daily Scrum*); Reunião da Revisão do Sprint (*Sprint Review Meeting*) e Reunião de Retrospectiva do Sprint (*Sprint Retrospective*).

Figura 1 - Cerimônias do Scrum



Fonte: Autor

O desenvolvimento de interfaces gráficas para sistemas computacionais que permitem interação entre usuário e sistema tornou-se uma das atividades fundamentais do processo de desenvolvimento de software. Nos últimos anos, empresas e instituições de ensino que atuam na área de Interação Humano Computador (IHC) aumentaram seu interesse por pesquisas sobre integração de usabilidade e desenvolvimento ágil de software (CURCIO *et al.*, 2019).

A norma ISO/IEC 25010 (ISO/IEC 25010, 2021) estabelece as características e subcaracterísticas externas e internas da qualidade de um software (SHEN *et al.*, 2018). Nessa norma, a usabilidade divide-se em seis subcaracterísticas: Inteligibilidade, Apreensibilidade, Operacionalidade, Proteção contra Erros do Usuário, Estética da Interface e Acessibilidade (ESAKI; AZUMA; KOMIYAMA, 2013).

O aumento de popularidade e aceitação dos métodos ágeis torna imperativa a necessidade de integrar engenharia de usabilidade nos processos de design e desenvolvimento. (UMAR *et al.*, 2016; DERAMAN; SALMAN, 2019). Usabilidade é o termo mais utilizado, para especificar a qualidade de um software (SAGAR; SAHA, 2017; JAIN; GUPTA; KHANNA, 2018).

Enquanto, no Ágil, o foco recai sobre os requisitos funcionais e, não detalhadamente, na interação com o usuário final, a usabilidade geralmente é tratada em segundo plano (MARQUES *et al.*, 2018). Combinar a validação de requisitos e a usabilidade é um longo caminho no desenvolvimento de um bom produto (DERAMAN; SALMAN, 2019), uma vez que o sucesso e a aceitação do software dependem não apenas das tecnologias usadas, mas de quão bem ele integra os métodos orientados ao usuário (UMAR *et al.*, 2016).

O presente trabalho utiliza a avaliação da usabilidade, para medir o grau de satisfação dos usuários que utilizam diariamente o Sistema de Gestão de Prestadores de Serviço (SGPS) sobre uma proposta de reestruturação interna do SUS, aplicando o algoritmo Para-analisador, utilizando como variáveis de entrada as opiniões emitidas pela avaliação dos usuários do sistema. Para realçar a contradição, a Lógica  $E\tau$  utiliza critérios lógicos, colaborando, assim, no processo decisório.

## 1.2. Justificativa

O sucesso do desenvolvimento de um software baseia-se na satisfação dos requisitos funcionais, dentre eles, a ER, fase inicial na abordagem de desenvolvimento de software, enquanto as Histórias de Usuários (HU) são escritas a partir de suas necessidades (CHOPADE; DHAVASE, 2017).

Cada HU é pormenorizadamente avaliada e mensurada pelo Time e pelo PO. Em caso de divergência ou falta de informação, as contradições são sanadas com o PO (SUTHERLAND, 2015). Os principais problemas encontrados na análise referem-se a falhas no atendimento do conceito INVEST: *Independent, Negotiable, Valuable, Estimable, Small, Testable* (OPAZO; LOPEZ, 2018).

A avaliação da usabilidade mede o grau de satisfação dos usuários que utilizam um sistema em seu dia a dia (OLAVERRI-MONREAL; HASAN; BENGLER, 2013). Ignorar as práticas de qualidade e o processo de avaliação dos aspectos

comportamentais e psicológicos na usabilidade pode acarretar consequências sociais, organizacionais, econômicas e provocar o fracasso do produto, considerando que o desenvolvimento do software deve centrar-se no ser humano (GHANBARI; VARTIAINEN; SIPONEN, 2018).

Diante disso, este trabalho teve, por principal objetivo, realizar um estudo das contradições apresentados na avaliação de HUs e na avaliação da usabilidade de software, aplicando a Lógica Paraconsistente Anotada Evidencial  $E_{\tau}$ , em duas dimensões.

Na etapa de Especificação de Requisitos, este estudo contribui com a melhoria da tomada de decisão no processo de avaliação, priorização e estimativas das HUs, com base em um modelo matemático que serve como instrumento de apoio para desenvolvedores, clientes e gerentes de projetos, entre outros profissionais.

Na etapa usabilidade de software, este trabalho permite avaliar a usabilidade em conjunto com o Para-analisador, analisando, de forma mais detalhada e aprofundada, as contradições, de modo a avaliar o resultado em duas dimensões.

### 1.3. Objetivo Geral

A Engenharia de Software insere-se na grande área da engenharia que estuda as melhores práticas para desenvolver softwares dentro de prazos, custos e qualidades aceitáveis (SOMMERVILLE, 2010).

Por meio de ferramentas, métodos e procedimentos da Engenharia de Software, os profissionais de tecnologia de informação desenvolvem softwares, maximizando a qualidade e minimizando custo, tempo e escopo de projeto.

A Lógica  $E\tau$  aplica-se a esses casos em dois momentos do processo de desenvolvimento:

- na subárea Engenharia de Requisitos (ER) (HORKOFF, 2018), auxilia o processo decisório de avaliação dos membros de uma equipe de Tecnologia da Informação (TI), por meio do algoritmo Para-analisador, de modo a oferecer respostas que ofereçam suporte no processo de tomada de decisão (GHOZALI *et al.*, 2019). Para tanto, utilizam critérios lógicos que permitem uma validação técnica, sendo os parâmetros de entrada definidos pelas opiniões dos especialistas, transformadas em termos matemáticos, consolidando, assim, uma lógica coletiva entre equipe de desenvolvimento e cliente;
- na subárea Interface Homem Computador (IHC), aplicou-se o instrumento SUS, por meio de um *survey*, para aferir a usabilidade do software. Depois de normalizar os dados, o algoritmo Para-analisador possibilitou uma avaliação em duas dimensões, atendendo aos critérios da Lógica  $E\tau$ , o que resultou uma análise mais minuciosa das opiniões dos especialistas.

#### 1.3.1. Objetivos Específicos

Para alcançar o objetivo geral, apresentam-se dois objetivos específicos, que contribuem para gerar conhecimento sobre a aplicabilidade da Lógica  $E\tau$ :

- A) dentro da subárea Engenharia de Requisitos, busca-se atender ao objetivo específico, aplicando a Lógica Paraconsistente Anotada Evidencial  $E\tau$  (Lógica  $E\tau$ ) no aprimoramento do processo de validação de requisitos de software em



projetos Scrum. Atualmente, o método Scrum lida com requisitos que mudam dinamicamente, trazendo desafios para o processo de tomada de decisão na Engenharia de Requisitos. Em muitas situações, as dificuldades que o PO e o Time enfrentam para entender e chegar a um consenso sobre os requisitos de negócios compromete o escopo do projeto e, conseqüentemente, seu tempo e custo.

B) realizado na subárea Interface Homem Computador (IHC), este estudo buscou atender ao objetivo específico da aplicação da Lógica  $E\tau$  em avaliações da qualidade da usabilidade do software em processos de tomada de decisão em produtos de software. A avaliação de usabilidade é uma atividade fundamental na Engenharia de Software, pois minimiza riscos e melhora a qualidade do produto. Os retornos da avaliação de usabilidade são inegáveis. Negligenciar essa avaliação durante a fase de desenvolvimento do produto impacta negativamente o projeto de TI. O uso da Lógica  $E\tau$ , nesse processo, permite realçar a contradição e a inconsistências de dados, colaborando para o processo decisório.

#### **1.4. Estrutura do Trabalho**

Com o objetivo de facilitar a leitura e o entendimento do presente trabalho, será apresentada a estrutura, acompanhada de uma breve descrição do conteúdo de cada capítulo.

O Capítulo I apresenta os conceitos abordados nesta pesquisa sobre a Engenharia de Requisitos e Interface Homem computador. Também traz os objetivos gerais e específicos, a justificativa e a introdução do trabalho, seu contexto e a motivação para desenvolvê-lo.

O Capítulo II expõe a fundamentação teórica sobre a metodologia ágil e o Scrum, o framework ágil mais utilizado do mercado. Apresentam-se os problemas que ocorrem no desenvolvimento de software e, em seguida, a Lógica Paraconsistente Anotada Evidencial  $E\tau$ .

O Capítulo III descreve métodos e ferramentas usados para desenvolver o trabalho.

O Capítulo IV traz os resultados obtidos com a aplicação da Lógica  $E\tau$  , acompanhados da discussão sobre o desenvolvimento do trabalho por meio de três artigos científicos.

O Capítulo V apresenta as considerações finais, as descobertas realizadas após a pesquisa e uma proposta de trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo, por meio de uma pesquisa bibliográfica, apresenta os conceitos que nortearam a realização deste trabalho. A seguir, oferece uma visão geral da metodologia ágil, bem como do uso do *framework* em gerenciamento de projetos; das contradições que ocorrem no desenvolvimento de software; do uso da Escala de Usabilidade do Sistema para mensurar Usabilidade e da Lógica Paraconsistente Anotada Evidencial  $E\tau$ .

### 2.1. Metodologia Ágil

A frequência, cada vez maior, com que as organizações adotam e aplicam as metodologias ágeis (CHOUDHARY; RAKESH, 2016; SATRIA; SENSUSE; NOPRISSON, 2017) demonstra interesse constante em melhorar os processos de desenvolvimento de software, procurando aumentar a qualidade do produto e diminuir o tempo de desenvolvimento, a partir de metodologias de desenvolvimento iterativo e incremental (CASTILLO-BARRERA *et al.*, 2017).

A metodologia ágil de software é um método de desenvolvimento iterativo e incremental. Adota uma abordagem mais leve, que abrange a maioria das particularidades do projeto de produção de produtos de software (POPLI; CHAUAHN, 2015).

A utilização da metodologia ágil permite reduzir custos no desenvolvimento de produtos de software e na prestação de serviços. Para alcançar essas vantagens, torna-se necessário, desde o início do projeto, uma metodologia ágil de suporte para a descoberta dos requisitos de Software e da dimensão de sua complexidade, a fim de que seja possível estimar o tempo adequado para o desenvolvimento e sucesso do projeto (CASTILLO-BARRERA *et al.*, 2018).

Desde o final da década de 1990, as metodologias de desenvolvimento ágeis de software vêm ganhando preferência em relação às convencionais. (COCKBURN, 2006; CHOUDHARY; RAKESH, 2016), em razão dos muitos projetos de software frustrados ou entregues com atraso e custo muito maior do que o originalmente previsto, com baixa qualidade e desempenho, mesmo quando faziam parte da equipe de projetos profissionais altamente qualificados e com capacidades técnicas acima da média. Acredita-se que a falha se deve ao fato das técnicas de gerenciamento de

projetos de software utilizadas serem adaptadas de outras disciplinas da área de engenharia, tornando-se ineficientes para o desenvolvimento de software (SOMMERVILLE, 2010).

Esse novo paradigma, com métodos mais leves, simples e ágeis, trouxe para as empresas novos benefícios, como a implementação de soluções mais rápidas, focando apenas nas funções necessárias; melhor qualidade do produto de software; menos tempo gasto com documentação e melhoria da satisfação do cliente, coletando feedback e reagindo rapidamente às mudanças de tecnologia e de software (INAYAT *et al.*, 2015; CHOUDHARY; RAKESH, 2016; FITRIANI; RAHAYU; SENSUSE, 2016).

Existem várias metodologias ágeis: Scrum, Kanban, Extreme Programming (XP), Lean, Crystal, entre outras (COCKBURN, 2006; INAYAT, *et al.*, 2015; FITRIANI, RAHAYU e SENSUSE, 2016). Ela podem ter procedimentos diferentes, no entanto, compartilham e seguem as mesmas premissas e valores do Manifesto Ágil:

- Dar mais importância a indivíduos e interações do que a processos e ferramentas;
- Priorizar o software em funcionamento, ao invés de ter uma documentação abrangente;
- Focar mais na colaboração com o cliente, do que na negociação de contratos;
- Dar mais importância a reagir rapidamente às mudanças do que se preocupar em seguir um plano.

Entre os *frameworks* mais utilizados, está o Scrum, que se define por interações. Em cada iteração é realizado um planejamento, uma análise de requisitos, um projeto, codificação, testes e documentação (SUTHERLAND, 2015; CASTILLO-BARRERA *et al.*, 2017).

## 2.2. Scrum Framework de Gerenciamento de Projetos

Projetos são um conjunto de atividades temporárias, executadas em equipe, com o objetivo de criar um produto, serviço ou solução única (PMI, 2017). Os projetos voltados para a área de Tecnologia da Informação (TI) alcançam seus objetivos por meio da tecnologia (SOMMERVILLE, 2010; PRESSMAN, 2011).

*Scrum* é um *framework* ágil de gestão e planejamento de projetos. Está entre as metodologias ágeis mais adotadas pelas empresas (SHARMA; HASTEER, 2016),

que ajudam times de pessoas e organizações a atingir objetivos e criar valor por meio de soluções adequadas para problemas complexos (CASTILLO-BARRERA *et al.*, 2018).

O *framework Scrum* foi criado e desenvolvido por Jeff Sutherland, para suceder o tradicional método cascata, considerado lento e imprevisível, a fim de gerenciar os projetos de softwares de forma mais ágil, transparente e eficiente, para entregar o produto final como os clientes pedem ou desejam (SUTHERLAND, 2015).

No *Scrum*, os projetos são divididos em ciclos de 2 a 4 semanas, chamados *Sprints* ou uma espécie de miniprojeto. *Sprint* é o cerne do *Scrum* e corresponde a um *Time Box*, no qual se executa um conjunto de atividades. Um *Sprint* começa imediatamente após a conclusão do *Sprint* anterior.

Na fase de Especificação de Requisitos (ER), é criado o *Product Backlog* (PB). Trata-se de uma lista priorizada de funcionalidades de software, constituída de requisitos, melhorias, correções, alterações, funções e demais desejos do cliente para um determinado produto (BARBOSA; SILVA; MORAES, 2016). Essa lista não é fixa; muda constantemente para adaptar-se ao produto, condições de mercado, tecnologia e ambiente onde será utilizado (SCHWABER, 2020).

O *Scrum* cria a lista do PB na forma de Histórias de Usuários (HUs) (LUCASSEN *et al.*, 2015; ORMSBY; BUSBY-EARLE, 2017). HUs são especificações de requisitos ou funcionalidades, escritos de maneira sucinta, como necessita o usuário ou a área de negócio que usa o produto de software (WAUTELET *et al.*, 2016).

Segundo Dalpiaz e Brinkkemper (2018), noventa por cento dos profissionais que utilizam a metodologia ágil, usam HUs para especificar requisitos. Essa popularidade se dá em razão da estrutura simples, porém, rígida, que captura, pela linguagem natural, os termos e jargões da área de negócio ou domínio do software (ZEAARAOUI *et al.*, 2013).

As HUs desempenham um papel importante para determinar quais requisitos de software e correções de bugs devem ser tratados e em qual ordem de prioridade.

Por meio das HUs, realizam-se as estimativas de tempo para a conclusão de cada tarefa e para o total do projeto (CASTILLO-BARRERA *et al.*, 2017; RODEGHERO *et al.*, 2017).

Do *Product Backlog*, seleciona-se as tarefas que farão parte do *Sprint Backlog*, que possui a lista de todas as atividades a serem realizadas no *Sprint*.

A equipe entrega o produto de forma iterativa e incremental, potencializando as oportunidades de *feedback* do cliente que contratou o serviço. As entregas incrementais permitem, após o Sprint, que uma versão do produto em funcionamento esteja sempre disponível para ser apresentada e testada pelo cliente, pois sua satisfação com determinado produto é afetada por sua experiência como usuário (UX), que determina se um aplicativo atende às expectativas. A medição quantitativa da UX depende do atendimento de necessidades psicológicas, que podem ser tratadas nas HUs e que validam a experiência com determinado aplicativo (OLAVERRI-MONREAL; HASAN; BENGLER, 2013).

O *Scrum* Permite incorporar, dentro de sua estrutura, outros processos e práticas existentes, técnicas de trabalho e métodos, para que melhorias possam ser aplicadas constantemente ao produto, a equipe e ao local de trabalho.

O *Scrum* atua sobre a inteligência coletiva de quem o utiliza, orientando seus relacionamentos e interações por meio da vivência dos cinco valores: Respeito, Foco, Comprometimento, Abertura e Coragem. Seus três pilares - inspeção, transparência e adaptação – estabelecem um clima de confiança entre as partes interessadas, que tendem a compartilhar um conhecimento comum sobre o processo. Essa interação faz com que o Time se comprometa com o trabalho e seus desafios, para atingir os objetivos. O foco principal, as partes, está no *Sprint* e em fazer o melhor progresso possível em direção aos objetivos. (SCHWABER, 2020)

O *Scrum* acolhe a incerteza e a criatividade. Coloca uma estrutura em volta do processo de aprendizagem, permitindo que as equipes avaliem o que já criaram e, o mais importante, de que forma o criaram. A estrutura do *Scrum* busca aproveitar a maneira como as equipes realmente trabalham, dando a elas as ferramentas para se auto-organizar e, o mais importante, aprimorar rapidamente a velocidade e a qualidade de seu trabalho. (SUTHERLAND, 2015)

A equipe Scrum é constituída por um pequeno time com membros auto-organizados, cada um com uma função específica. O membro *Scrum Master* é responsável por promover e dar suporte para a equipe sobre a aplicação e entendimento da teoria, prática e aplicação das regras e valores Scrum, promovendo um ambiente onde:

1. Um *Product Owner* ou responsável pelo produto e pelo *Product Backlog* direciona o trabalho para resolução de um problema complexo por meio de uma lista de atividades ou *Product Backlog*. Esse membro da equipe que representa as necessidades e desejos do cliente, em geral, é chamado de *stakeholder* ou usuário chave.

2. O *Scrum Team* ou *Time Scrum*, que realiza e desenvolve uma atividade do *Product Backlog*, transformando-a em um incremento de valor do produto durante um Sprint. Os membros do time de desenvolvimento têm estrutura e capacidade para organizar e gerenciar seu próprio trabalho, trazendo eficácia e eficiência para os objetivos.

3. O *Time Scrum* e os *stakeholders* ou usuários chaves, responsáveis pelo projeto, conferem e aprovam os resultados de um Sprint; refinam e melhoram seus processos para o próximo *Sprint*.

4. Reiniciam a interação do processo no passo 1. (SCHWABER, 2020)

Para conduzir o progresso do projeto em direção às metas e detectar variações ou problemas potencialmente indesejáveis, promove-se, no Scrum, a inspeção por meio de cinco eventos: *A Sprint*, *A Sprint Planning*, *Daily Scrum*, *Sprint Review* e *Sprint Retrospective*. (KNIBERG, 2015)

O trabalho a ser executado em um Sprint inicia com a reunião de Planejamento do Sprint (*Sprint Planning*).

O *Time Scrum* colabora, nessa reunião, para definir todo o plano do trabalho a ser realizado e o objetivo do Sprint, fornecendo incrementos com funcionalidades úteis que atendam à definição de aceitável pela Equipe, que deve trazer valor para as partes interessadas. Selecionam-se, então, os itens do *Product Backlog*, uma lista de requisitos do usuário, para serem trabalhados e desenvolvidos no *Sprint*. Essa lista passa a ser chamada *Sprint Backlog* (CORREIA; GONÇALVES; MISRA, 2019).

Define-se o *minimum viable product* (MVP) do produto o Sprint com o menor conjunto possível de requisitos necessários para formar um pacote do produto, para ser instalado em produção, de modo que o ele comece a produzir valor para a organização e que os usuários comecem a fornecer *feedbacks*. (SUTHERLAND, 2015)

Para cada item do *Sprint Backlog*, são elaboradas sua descrição, priorização, estimativa e valor. O *Time* de desenvolvimento começa a projetar o sistema e a estimar o esforço necessário. A responsabilidade das estimativas é da Equipe de

Desenvolvimento, mas o *Product Owner* pode influenciar, ajudando a entender os detalhes de implementação dos itens selecionados no *product Backlog*, bem como determinar se há muito ou muito pouco trabalho e a renegociar esses itens do produto que podem influenciar na estimativa final. (SUTHERLAND, 2015)

O *Daily Scrum* é um evento de *Time-Box*, uma reunião chave de inspeção e adaptação dentro do Sprint, realizada todos os dias, durante quinze minutos, pela Equipe de Desenvolvimento, podendo também participar o *Scrum Master* e o *Product Owner*. O propósito de a reunião ser diária é inspecionar o progresso das atividades em direção ao Objetivo do Sprint, ajustando o trabalho planejado e adaptando o *Backlog* do Sprint, conforme necessário. A equipe também produz um plano tangível para o próximo dia de trabalho, criando foco e melhoria no processo com auto-organização. (SCHWABER, 2020)

No final do Sprint, é realizada o *Sprint Review*, com o propósito de inspecionar as entregas do Sprint e especificar futuras adaptações. Os membros da equipe discutem sobre o que foi feito. O resultado do Time Scrum é apresentado para as Partes Interessadas e o progresso do Objetivo do Produto, discutido, podendo ocorrer novas inclusões ou alterações de requisitos no *Product Backlog*. O resultado do *Sprint Review* é um *Product Backlog* revisado, que pode ser totalmente ajustado para atender novas oportunidades e definir os prováveis incrementos para o próximo *Sprint*. (SCHWABER, 2020)

O *Sprint Retrospective* é uma oportunidade para a Equipe inspecionar-se e criar um plano para melhorar a qualidade e a efetividade. É responsabilidade do *Scrum Master* garantir que o evento ocorra e que os envolvidos entendam seu propósito. O time de desenvolvimento é encorajado por ele a melhorar o processo de desenvolvimento, tornando-o mais eficiente para o próximo Sprint. Ao longo de um *Sprint Retrospective*, a Equipe identifica e planeja maneiras de aumentar a qualidade do produto, melhorando os processos de trabalho, realizando um refinamento do *product Backlog*, adicionando detalhes e definindo novas estimativas. Trata-se de um processo contínuo do Scrum, no qual o *Product Owner* e o time de desenvolvimento refinam sucessivamente o *product Backlog* (SCHWABER, 2020).



### 2.3. As contradições no desenvolvimento de software

No jogo de *Rugby*, utiliza-se o termo *Scrum* para definir a reunião de jogadores realizada em formato de círculo, para planejar como será realizada a próxima jogada.

O *Sprint* é uma espécie de miniprojeto, definido como um pequeno ciclo que dura de uma a quatro semanas, cujo objetivo é desenvolver um incremento do produto de software.

É tarefa do Scrum Master orientar o Time, bem como garantir que o planejamento do Sprint aconteça e que seus participantes entendam o propósito da construção do incremento do produto. O *Product Owner* (Proprietário do produto), responsável por esclarecer os itens selecionados do *Sprint Backlog*, pode negociar a priorização desses itens com o Time.

Durante a reunião do Planejamento do Sprint, todo o time trabalha para entender, estimar, priorizar e planejar a funcionalidade que será desenvolvida durante o Sprint, com o objetivo de gerar um incremento de produto, que é a soma de todos os itens ou requisitos de software do *Sprint Backlog* concluídos durante o Sprint (CORREIA; GONÇALVES; MISRA, 2019). O planejamento do trabalho começa com o Time selecionando itens da lista de Backlog do produto e priorizando as tarefas que se comprometem a desenvolver dentro de um Sprint. A priorização dessas tarefas, depende de vários fatores, como recursos, tempo, custo, etc (BIK; LUCASSEN; BRINKKEMPER, 2017). Existem métodos usados para priorizar HUs, mas de acordo com Sheemar e Kour (2017), não são eficientes.

Independentemente da metodologia adotada, é necessário entender, conhecer e identificar a complexidade dos requisitos do cliente, para conseguir planejar executar um Sprint de maneira adequada (SOMMERVILLE, 2010; KAMTHAN; SHAHMIR, 2016).

No Scrum, os requisitos dos usuários são apresentados em formato de Histórias dos Usuários (HU), técnica utilizada nos métodos ágeis, para capturar e escrever requisitos funcionais (CHOPADE; DHAVASE, 2017). Em geral, as HUs são capturadas em reuniões entre o Time, o proprietário do produto, o Scrum Master e o usuário, podendo ocorrer entre eles contradições que resultam métricas inexatas (CHAUHAN, 2015). O Time precisa ter bom conhecimento sobre o domínio da aplicação e do negócio onde o sistema será utilizado, a fim de especificar bons

requisitos ou HUs e criar testes de aceitação e validação dos itens do Sprint Backlog (HEIKKILÄ *et al.*, 2015).

Para tanto, existem várias técnicas de elicitación de requisitos, que requerem um comprometimento das partes interessadas, tais como entrevistas com usuários, análise de documentos, aplicação de questionários e brainstorming (JARZĘBOWICZ; POŁOCKA, 2017). Porém, muitas vezes, as partes envolvidas no projeto têm dificuldades de realizar essas técnicas, por falta de tempo ou de disponibilidade (RAHARJANA; SIAHAAN; FATICHAH, 2019).

Para iniciar um Sprint, na etapa de validação da qualidade das HUs, os desenvolvedores avaliam se essas são compreensíveis o suficiente em sua complexidade e se estão bem fundamentadas. Caso verifique alguma incerteza, duplicidade ou falta de informação, o Time esclarece suas dúvidas junto ao *Product Owner* (PO), representante do cliente, ou junto ao próprio cliente. Esse processo de validação é um fator importante para o projeto, pois garante que os requisitos estejam validados, documentados e corretos (MATEEN; ABBAS; AKBAR, 2017).

Um aspecto importante para o sucesso do projeto é a estimativa de tempo e a análise do nível de complexidade que cada HU requer para sua realização (ORMSBY; BUSBY-EARLE, 2017). A captura dessas informações permite que o Time determine o tempo total do projeto (CASTILLO-BARRERA *et al.*, 2017). Porém, essa atividade ainda apresenta problemas, pois consome tempo e gera estimativas insuficientes, devido à complexidade das HUs. Muitas vezes são escritas de maneira incompleta ou muito simples (HESS; DIEBOLD; SEYFF, 2017), com requisitos mal formulados (LISKIN *et al.*, 2014), o que coloca essa atividade entre as mais arriscadas em projetos ágeis de software, prejudicando o planejamento do desenvolvimento de software e, consequentemente, o sucesso do projeto (CASTILLO-BARRERA *et al.*, 2018).

Outro aspecto do processo de validação é o consenso coletivo, dado que a análise de cada HU é individual e o Time precisa ter a experiência, conhecimento e habilidades exigidas pela HU para sua realização. Pode haver contradições de análise entre os membros do Time devido ao conhecimento do domínio e à experiência profissional de cada um (HANNAY; BENESTAD; STRAND, 2019).

A natureza incerta do tamanho de uma HU, bem como o fato de que novas HUs podem ser adicionadas e as já existentes podem ser removidas em um Sprint criam incertezas no projeto Scrum, que levam a uma estimativa pobre de tempo e custo (POPLI; CHAUAHN, 2015).

O processo de validação das HUs pode originar uma lista de problemas. O Time e o PO precisam entrar em acordo, de modo a criar um plano de ação para resolvê-los. Geralmente os problemas encontrados não estão de acordo com o critério do INVEST, um acrônimo para histórias Independente, Negociável, Valiosa, Estimável, Small ou pequenas e Testável (BUGLIONE; ABRAN, 2013).

As decisões que surgirem da reunião de planejamento para entendimento de uma determinada HU dependem do consenso dos envolvidos, que pode ser expresso como diferenças entre os membros do time ou meramente como contradições lógicas.

No Scrum, há várias técnicas para estimar os pontos de uma HU, como o Planning Poker, julgamento de especialistas e analogia com outros projetos. Essa técnica para estimar o tamanho das HUs exige discussões e interação face a face entre os membros do Time (SUTHERLAND, 2015). Apesar de seus muitos benefícios, esse método não é totalmente eficiente, já que o resultado é sempre baseado na observação e experiência de um especialista do Time. A estimativa de esforço pode ser realizada por membros inexperientes da equipe, ainda não familiarizados com o domínio de negócios específicos ou sem experiência profissional adequada para a situação (CHONGPAKDEE; VATANAWOOD, 2017).

## 2.4. Escala de Usabilidade do Sistema

Entre as várias pesquisas disponíveis para avaliar a usabilidade de um produto ou serviço, a Escala de Usabilidade do Sistema (SUS) (BROOKE, 1995) é o questionário padronizado mais utilizado (LEWIS, 2018), por ser rápido para medir a usabilidade (DREW; FALCONE; BACCUS, 2018). Conforme a norma ISO/IEC 25010 (2021), usabilidade e confiabilidade são dois aspectos importantes em um software. A usabilidade é a ideia central da interação homem-computador (OLAVERRI-MONREAL; HASAN; BENGLER, 2013), pois é a experiência do usuário que determina se um aplicativo atende às suas expectativas. Por esse motivo, a avaliação de usabilidade é uma atividade essencial, já que mitiga riscos e melhora a qualidade do produto (DERAMAN; SALMAN, 2019). O SUS possui várias características que tornam sua utilização interessante. Seu uso é gratuito e pode ser utilizada para avaliar quase todo tipo de interface de usuário, tais como sites e aplicações mobile, entre outros. Composto de apenas dez questões padronizadas, torna rápido e fácil para o participante do estudo responder e, para os pesquisadores, calcular a pontuação imediatamente após a conclusão (BROOKE, 1995). Para responder às questões, o usuário seleciona uma resposta, que está atribuída a um modo de distribuição escalar, com valores que variam de 1 a 5, nos extremos, de discordo totalmente até concordo totalmente, conforme mostra o

### Quadro 1.

Quadro 1 - Escala de usabilidade do sistema

Questões
Q1 - Eu acho que gostaria de usar essa aplicação frequentemente.
Q2 - Eu achei essa aplicação desnecessariamente complexa.
Q3 - Eu achei a aplicação fácil para usar.
Q4 - Eu acho que precisaria do apoio de um suporte técnico para usar essa aplicação.
Q5 - Eu achei que as várias funções da aplicação estavam bem integradas.
Q6 - Eu achei que havia muita inconsistência na aplicação.
Q7 - Imagino que a maioria das pessoas possa aprender a utilizar este aplicativo muito rapidamente.
Q8 - Achei a aplicação muito complicada de se usar.
Q9 - Eu me senti muito confiante em utilizar esta aplicação.
Q10 - Eu precisei aprender várias coisas antes que eu pudesse começar a usar essa aplicação.
Critério
( ) Discordo totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo totalmente

Fonte: Brooke (1995)

Para calcular o resultado da pesquisa, utiliza-se uma pontuação única, variando de 0 a 100, procedimento relativamente fácil de entender por pessoas de qualquer área do conhecimento que trabalhem em equipes de projeto.

Para realizar o cálculo do resultado do SUS, a somatória de todas as contribuições da pontuação, para os dez itens, são multiplicados por 2.5, conforme mostra a Equação 1.

Equação 1 - Fórmula do SUS

$$SUS = \left[ \sum_{n=1}^5 (U_{2n-1} - 1 + (5 - U_{2n})) \right] \times 2.5 \quad (1)$$

Fonte: Brooke (1995)

Quanto mais alto os valores dos resultados, maior a satisfação das expectativas do usuário com o produto. Resultados com pelo menos 90 pontos significam que a expectativa, na usabilidade, está acima do desejado. Valores entre 80 e 90 pontos refletem usabilidade excelente. Resultados entre 70 e 80 pontos denotam boa usabilidade, mas apontam que algumas melhorias podem ser realizadas. Os que alcançam resultado entre 60 e 70 pontos são considerados “ok”, entretanto, devem ser fortemente melhorados. Por fim, os abaixo de 60 pontos não apresentam grau de usabilidade satisfatório (BROOKE, 1995).

## 2.5. Lógica Paraconsistente Anotada Evidencial $E_{\tau}$

A lógica clássica baseia-se nos estados zero e um, o que permite trabalhar somente com valores de forma binária (0 ou 1). Na Lógica Aristotélica, uma afirmação pode ser verdadeira ou falsa, atendendo ao princípio da não contradição. A lógica booleana, por exemplo, só admite um único valor para um termo qualitativo que avalia uma determinada situação com resultado certo ou errado (CARVALHO; ABE, 2018).

Os sistemas e computadores do mundo contemporâneo exigem, cada vez mais, aplicações do uso da lógica não clássica no desenvolvimento dos algoritmos computacionais, que admitam tratar mais de dois estados. Um exemplo:

questionando, em uma avaliação qualitativa de temperatura, se o dia está muito quente, pode-se obter os seguintes resultados: concordo totalmente, concordo, indiferente, discordo e discordo totalmente.

A Lógica Paraconsistente Anotada Evidencial  $E\tau$  (Lógica  $E\tau$ ) faz parte das chamadas lógicas não clássicas (ABE; SILVA LOPES; ANGHINAH, 2016), por conter disposições contrárias a alguns dos princípios essenciais da Lógica Aristotélica, por exemplo, o princípio da contradição, já que ela pode conter proposições totalmente contraditórias e ambas serem verdadeiras (ABE, 2016).

Perante a Lógica Aristotélica, qualquer proposição é necessariamente verdadeira ou falsa. Já na Lógica  $E\tau$ , uma sentença e sua negação podem ser, ambas, verdadeiras. Essa lógica surgiu com o reconhecimento, por parte da comunidade científica (ABE, 2015), do filósofo russo Nicolai Alexandrovich Vasilév, nascido em 1910, e do lógico polonês Jan Łukasiewicz Lvov, nascido em 1878, que formulou os primeiros sistemas lógicos polivalentes, considerados predecessores da Lógica Paraconsistente, denominada, também, Lógica Imaginária.

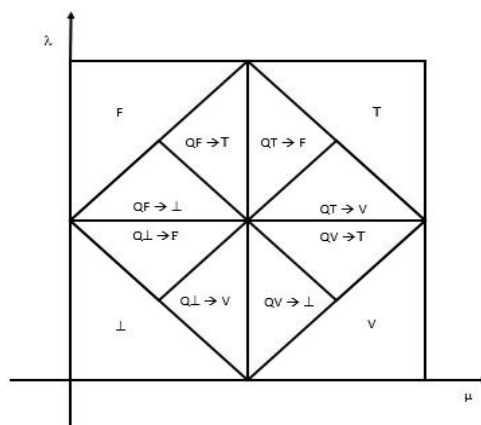
Em 1948, o polonês Stanilaw Jaśkowski, discípulo de Łukasiewicz publicou estudos sobre cálculo proposicional paraconsistente. Na década de 1950, o brasileiro matemático lógico, Newton Carneiro Affonso da Costa e Jaśkowski propuseram a contradição na estrutura lógica e ficaram conceituados como os iniciadores da Lógica Paraconsistente. Em 1954, o professor Newton Carneiro apresentou lógicas de ordem superior (teoria de conjuntos) e predicados, dando uma abordagem tanto matemática quanto filosófica para estudos e sistemas que envolvem uso da lógica e podem abranger contradições (ABE *et al.*, 2011).

O ser humano pode tomar decisões baseado em entendimentos subjetivos ou intuitivos. Ambos não têm medidas exatas, sendo inconsistentes ou contraditórios (SILVA FILHO; TORRES; ABE, 2010). Na Lógica  $E\tau$ , como uma Lógica Não Clássica, apresenta soluções para resolver os problemas gerados pela contradição, pois considera, dentro da sua estrutura, o tratamento de dados contraditórios e incertos, evidenciando uma solução para as circunstâncias não abrangidas pela Lógica Clássica (ABE; AKAMA; NAKAMATSU, 2015).

A linguagem da Lógica  $E\tau$  consiste de expressões onde  $p$  é uma proposição, no sentido habitual, e o par  $(\mu, \lambda)$  chama-se constante de anotação onde  $\mu$

demonstra o grau de evidência favorável e  $\lambda$  o grau de evidência desfavorável, respectivamente, expressa por  $p$ , estando os valores de  $\mu$  e  $\lambda$  limitados entre 0 e 1, conforme Figura 2 (ABE, 2015; AKAMA, 2016).

Figura 2 - Apresentação do Reticulado  $\tau$



Fonte: Abe (2015).

Na apresentação do reticulado, conforme se vê na Figura 2, visualizam-se os estados da decisão que compõem quatro valores limites extremos, representados por meio de seus símbolos, conforme Tabela 1.

- (1;0): evidência favorável total e evidência desfavorável nula (Verdadeiro);
- (0;1): evidência favorável nula e evidência desfavorável total (Falso).
- (1;1): evidência favorável total e evidência desfavorável total (Inconsistente);
- (0;0): evidência favorável nula e evidência desfavorável nula (Paracompleto);

Tabela 1 - Estados Extremos

Item	Símbolo	Estados Extremos
01	V	Verdadeiro
02	F	Falso
03	T	Inconsistente
04	$\perp$	Paracompleto

Fonte: Abe (2015).

Na Lógica  $E_{\tau}$ , os estados lógicos são denominados extremos. Além desses quatro Estados Extremos, a Lógica  $E_{\tau}$  possui oito estados não extremos, denominados de acordo com a proximidade com os estados lógicos extremos. Dessa

forma, contribui com possíveis resultados, permitindo mais possibilidades de análise, para um apoio fundamental no momento da tomada de decisão, conforme Tabela 2.

Tabela 2 - Simbolização dos Estados Não-Extremos

Estados Não-Extremos	Símbolo
Quase verdadeiro tendendo para o inconsistente	$QV \rightarrow T$
Quase verdadeiro tendendo para o paracompleto	$QV \rightarrow \perp$
Quase falso tendendo para o inconsistente	$QF \rightarrow T$
Quase falso tendendo para o paracompleto	$QF \rightarrow \perp$
Quase inconsistente tendendo para o verdadeiro	$QF \rightarrow \perp$
Quase inconsistente tendendo para o falso	$QT \rightarrow F$
Quase paracompleto tendendo para o verdadeiro	$Q\perp \rightarrow V$
Quase paracompleto tendendo para o falso	$Q\perp \rightarrow F$

Fonte: Abe (2015).

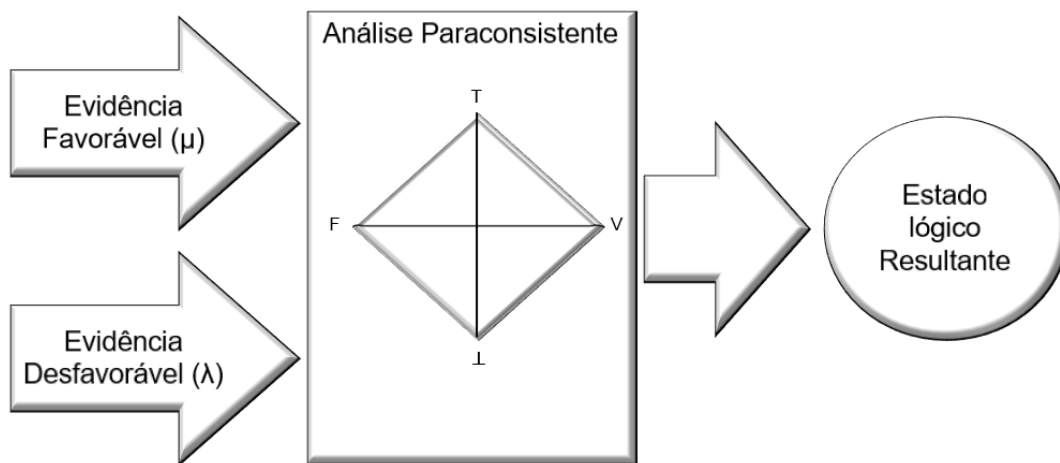
As opiniões coletadas entre os profissionais servem de entrada para o processamento dos dados, onde, nas proposições  $\mu$  e  $\lambda$ , que representam as opiniões favoráveis e desfavoráveis para questão, são empregados os conectivos AND e OR.

Os graus de evidência favorável e de evidência desfavorável são utilizados como dados de entrada do sistema de Análise Paraconsistente. Então aplica-se o uso de técnicas com os conectivos de maximizar e minimizar em suas formas atômicas “A” e “B”, para gerar os estados lógicos extremos ou não-extremos, resultantes de saída, demonstrados nos vértices do reticulado, conforme



Figura 3.

Figura 3 - Sistema de Análise Paraconsistente



Fonte: (ABE, 2015)

Os graus de evidência favorável e de evidência desfavorável no sistema de análise paraconsistente podem ser compreendidos como conhecimento, pois a medida que se adquirem novas evidências, minimizam as contradições e amplia-se a capacidade do sistema de obter uma conclusão mais adequada (ABE; AKAMA; NAKAMATSU, 2015).

Na Lógica Clássica, as ideias de disjunção e conjunção equivalem aos operadores OR e AND. A maximização é representada como  $(\mu_1, \lambda_1) \text{ OR } (\mu_2, \lambda_2) =$

$(\text{Max } \{\mu_1, \mu_2\}; \text{Min } \{\lambda_1, \lambda_2\})$ , onde Max designa a operação de maximização de números reais com a ordem ordinária e Min assinala a operação minimização de números reais com a ordem ordinária.

A utilização da técnica de maximização permite extrair o maior valor entre os graus de evidência favorável ( $\mu_1$  OR  $\mu_2$ ). Da mesma maneira, a utilização da técnica de minimização possibilita obter o menor valor entre os graus de evidência desfavorável ( $\lambda_1$  OR  $\lambda_2$ ), obtendo-se, então, o grau de evidência desfavorável resultante (DA SILVA FILHO, 1999).

A operação AND é especificada como  $(\mu_1, \lambda_1) \text{ AND } (\mu_2, \lambda_2) = (\text{Min } \{\mu_1, \mu_2\}; \text{Max } \{\lambda_1, \lambda_2\})$ , onde Min estabelece a operação de minimização de números reais com a ordem ordinária e Max determina a operação maximização de números reais com a ordem ordinária.

Após aplicar as técnicas de maximização e minimização, pode-se obter os graus de certeza e de incerteza, conforme segue:

- Para calcular o Grau de Certeza utiliza-se a seguinte fórmula:  

$$G_{ce}(\mu, \lambda) = \mu - \lambda \quad (0 \leq \mu, \lambda \leq 1);$$
- Para calcular o Grau de Incerteza, utiliza-se a seguinte fórmula:  

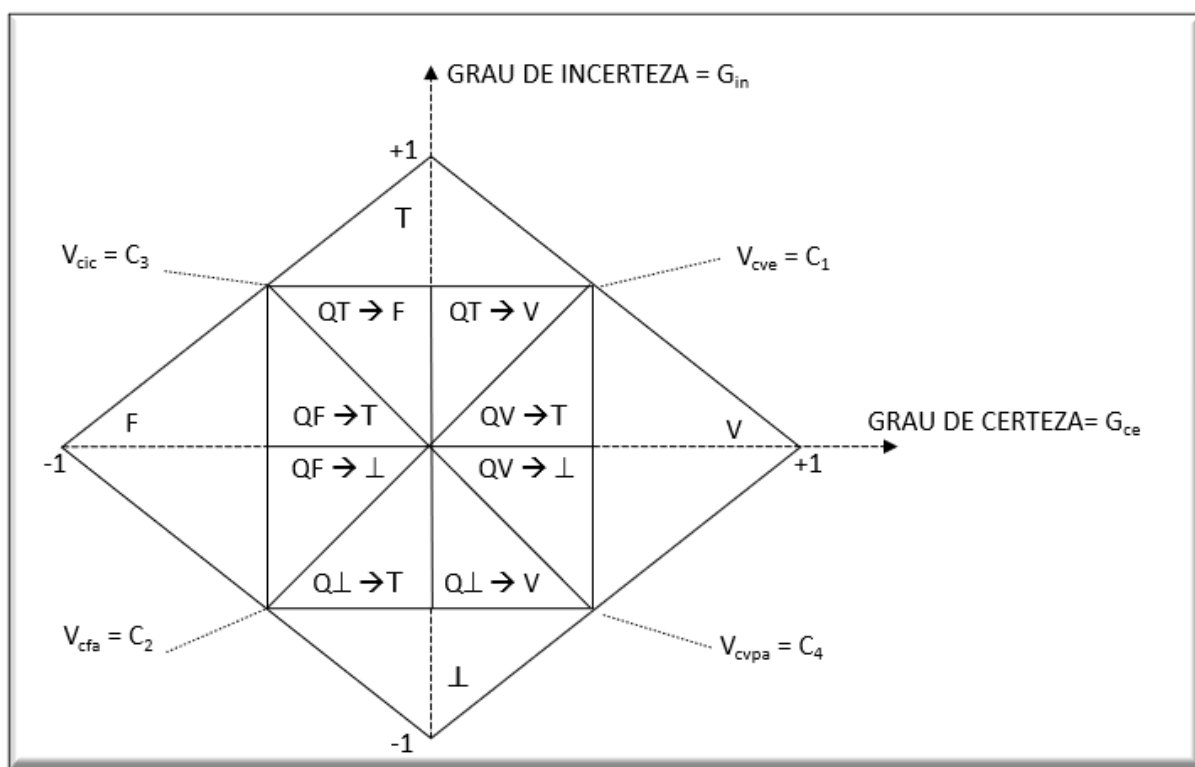
$$G_{in}(\mu, \lambda) = \mu + \lambda - 1 \quad (0 \leq \mu, \lambda \leq 1);$$

Denomina-se Grau de Certeza  $G_{ce}(\mu, \lambda)$  de uma anotação  $(\mu, \lambda)$  a qualquer um dos graus de verdade ou graus de falsidade.

Denomina-se Grau de Incerteza  $G_{in}(\mu, \lambda)$  de uma anotação  $(\mu, \lambda)$  qualquer um dos graus de paracompleteza ou inconsistência.

Os graus de certeza e incerteza permite trabalhar com uma gama maior de faixa de valores - tanto de verdade como de falsidade -, ao invés de considerar somente a verdade ou falsidade absoluta, conforme demonstra a Figura 4.

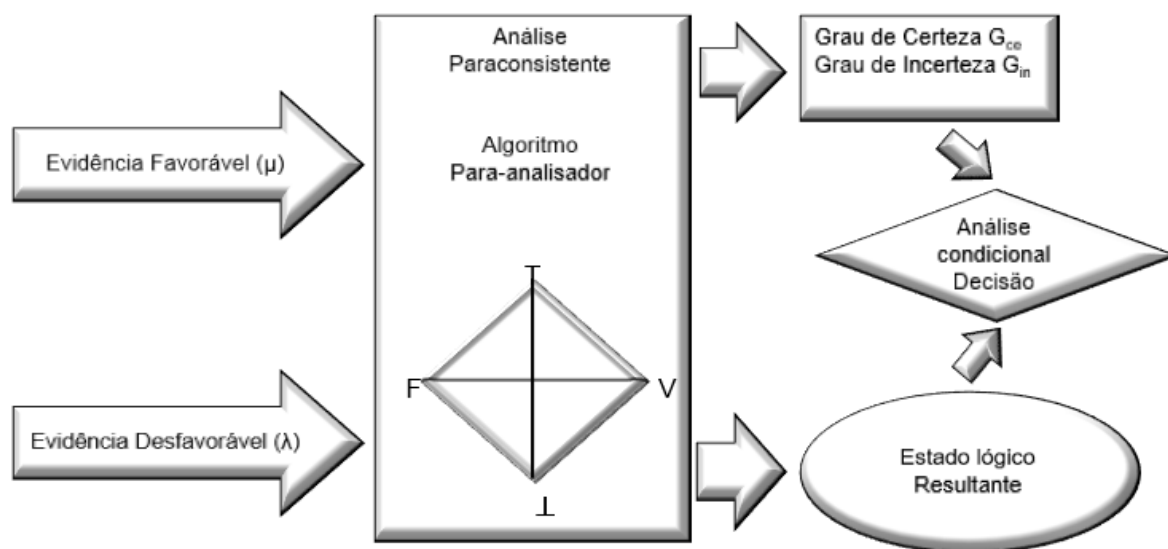
Figura 4 - Graus de certeza e incerteza



Fonte: Abe (2015).

A principal preocupação, em qualquer análise, é saber como medir ou determinar o grau de certeza em relação a uma proposição, se ela é falsa ou verdadeira (ABE *et al.*, 2018). O algoritmo Para-analisador é um conjunto de regras que permite, a partir dos valores de graus de evidência favorável e graus de evidência desfavorável, analisar as proposições, a fim de obter um estado lógico resultante, conforme indica a Figura 5. Ele possibilitou o desenvolvimento e aplicação nas áreas da Saúde (ABE; SILVA LOPES; ANGHINAH, 2016), na Robótica (TORRES; REIS, 2015) e nas Redes Neurais (SOUZA; ABE, 2015).

Figura 5 - Representação do algoritmo Para-analisador

Fonte: (ABE *et al.*, 2011)

### 3 METODOLOGIA

Esta seção apresenta os procedimentos metodológicos utilizados, com o propósito de responder aos seus objetivos. Esta pesquisa é de natureza aplicada, com finalidade exploratória e de tratamento quali-quantitativo, pois objetiva construir um conhecimento sobre a aplicação da Lógica  $E\tau$  em dois subprocessos comuns no desenvolvimento de software: a Engenharia de Requisitos e a Interação Humano-Computador.

#### 3.1. Engenharia de Requisitos: Aplicação da Lógica $E\tau$ na avaliação da História de Usuário

Esta seção mostra os procedimentos metodológicos utilizados na aplicação da Lógica  $E\tau$  na análise das HUs, mediante o processo de avaliação delas para uma equipe de especialistas da área de desenvolvimento de software.

Foi utilizado, como procedimento metodológico, um survey, a fim de coletar as opiniões dos profissionais sobre a avaliação das HUs, seguindo o critério INVEST. Os participantes não foram identificados, mantendo-se o sigilo, sem coleta de dados pessoais.

Foram selecionados nove profissionais da área de Tecnologia da Informação, deste ponto em diante, denominados “especialistas”, que atuam como consultores autônomos em várias empresas de desenvolvimento de softwares na cidade de São Paulo, todos eles trabalhando com desenvolvimento de aplicativos utilizando a metodologia Scrum. Os especialistas foram divididos em três grupos (Tabela 3), adotando-se o cargo como critério de agrupamento, sendo denominados E1, E2, ..., E9 na base de dados. Os cargos dos especialistas são: Analistas de Sistemas Sênior e Pleno, Arquitetos de Software e Scrum Master.

Tabela 3 - Classificação dos Grupos de Especialistas

Grupo	Cargo	Quantidade
A	Arquiteto de Software e Scrum Master	3
B	Analista de Sistema Sênior	3
C	Analista de Sistema Pleno	3

Fonte: Autor

Conforme mostra o Quadro 2, o INVEST é composto de questões que avaliam os requisitos do usuário sob a ótica da independência de requisitos, pois a Engenharia de Requisitos depara-se com requisitos interdependentes, que podem consumir mais de um Sprint. O fator negociável diz respeito à definição de prioridade dos requisitos, pois pode haver divergência de opiniões entre o TIME e o PO em razão do escopo e da interdependência dos requisitos do projeto.

Apesar do fator valor estar relacionado com a geração de valor para o negócio, o TIME e o PO podem ter percepções qualitativas diferentes sobre o valor da HU. O fator estimável refere-se à estimativa de tempo de desenvolvimento do software para que o requisito seja desenvolvido dentro de um Sprint, conforme apresentado no item 2.2.

O fator pequeno (ou *small*) avalia se a HU tem um tamanho ideal para um Sprint ou pode ser dividida em partes menores. Esse fator pode gerar incertezas na estimativa.

O fator testável compreende o processo de homologação do software, pois a realização dessa etapa envolve critérios claros e precisos entre PO e o TIME. Essa avaliação resulta, muitas vezes, em divergência entre os especialistas.

Quadro 2 - Fatores e os critérios do INVEST

<b>Fator</b>	<b>Nome do Fator</b>
F1	Independente
F2	Negociável
F3	Valor
F4	Estimável
F5	Pequeno
F6	Testável

Fonte: Autor

Eles aferiram cada HU, quantificando  $\mu$  e  $\lambda$  de acordo com seus conceitos sobre cada um dos critérios INVEST, por meio de uma escala Likert.

Os valores qualitativos resultantes dessa coleta, foram normalizados para valores quantitativos, por meio de um método de distribuição escalar, com valores, variando de zero a um, entre os extremos “concordo totalmente” e “discordo totalmente”, igualando-se aos critérios numéricos atribuídos.

### Fatores da Lógica $E_{\tau}$ atribuídos aos critérios do INVEST,

Tabela 4 - Normalização dos valores de  $\mu$  e  $\lambda$

<b>Resposta</b>	<b>Grau</b>	<b>Descrição</b>
Concordo totalmente	1,00	Não existe nenhuma dúvida em relação a evidência
Concordo	0,75	Pequena dúvida em relação a evidência
Indiferente	0,50	Dúvida média em relação a evidência
Discordo	0,25	Baixa certeza em relação a evidência
Discordo totalmente	0,00	Quase nenhuma certeza em relação a evidência

Fonte: Autor

A análise dos resultados foi realizada por meio do algoritmo Para-analisador. Para facilitar a mensuração os valores de  $\mu$  e  $\lambda$  foram normalizados (Tabela 4).

Os participantes foram contatados por e-mail, para confirmar que aceitavam a participação, o instrumento de avaliação e o recebimento das HUs.

Quatro HUs, provenientes de levantamento de requisitos realizado em empresas foram selecionadas para avaliação e os respondentes desconheciam as mesmas.

Quadro 3 - Histórias de Usuário

HU	História de usuário	Crítérios de aceitação
A	Como um Cliente, eu quero saber o preço e a disponibilidade de um produto para verificar se quero concluir a compra.	- Produto deve possuir o nome e o preço. - Produto deve estar associado a uma categoria e um fornecedor.
B	Como um Administrador eu quero consultar as categorias de produtos para visualizar as categorias de produto.	- Não listar as categorias que estejam descontinuadas.
C	Como um Administrador eu quero consultar produtos por categoria, eu conheço o código da categoria e quero encontrá-lo no sistema. para descobrir seus produtos para visualizar os produtos de uma categoria informada	- Não listar os produtos que estejam descontinuados. - Deve-se listar todas os produtos mesmo que estejam com o estoque zerado.
D	Como um cliente eu quero que o sistema disponibilize diversas formas de pagamento para poder pagar o meu pedido e o sistema fechar o pedido.	- Cliente pode selecionar uma forma de pagamento para poder efetuar pagamento. - Quando o pagamento for realizado com cartão de crédito, validar com a operadora a operação ou restrição. - Estando tudo ok prosseguir com a finalização do pedido.

Fonte: Autor

Uma base de dados foi elaborada a partir das respostas dos especialistas, com os valores das evidências colhidos de  $\mu$  e  $\lambda$  para cada HU, conforme Quadro 3. A Figura 6 apresenta uma pequena amostra da base de dados da história de usuário A. As bases de dados por HU, que se encontram-se no Capítulo 4, servem de entrada de dados para o algoritmo Para-analisador, desenvolvidos em uma planilha Excel®. Em seguida, foram aplicadas as regras de maximização (OR) e de minimização (AND) da Lógica  $E_{\tau}$  para as opiniões dos especialistas, para cada HU.

Figura 6 - Amostra da Base de dados formada por  $\mu$  e  $\lambda$  da história de usuário A

Fator	Grupo A						Grupo B						Grupo C					
	E1		E2		E3		E4		E5		E6		E7		E8		E9	
	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$
F01	0,25	0,75	0,25	1,00	0,50	0,75	0,25	0,75	0,00	0,75	0,25	0,75	0,25	0,50	0,00	1,00	0,25	1,00
F02	0,50	0,50	0,75	0,00	0,75	0,25	1,00	0,00	0,75	0,00	0,75	0,25	1,00	0,25	0,75	0,25	0,75	0,00
F03	1,00	0,00	1,00	0,25	1,00	0,00	1,00	0,25	0,75	0,25	1,00	0,00	0,75	0,00	1,00	0,25	1,00	0,00
F04	0,75	0,25	0,00	0,75	0,00	1,00	0,25	0,75	0,25	0,75	0,00	1,00	0,25	1,00	0,25	0,75	0,25	1,00
F05	1,00	0,00	0,00	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,00	0,75	0,25	1,00	0,25	0,75	0,25	1,00
F06	1,00	0,00	0,75	0,50	0,75	0,25	1,00	0,00	1,00	0,50	1,00	0,25	1,00	0,75	1,00	0,25	0,75	0,25

Fonte: Autor



### 3.2. Metodologia Aplicada na Avaliação da Usabilidade de Software.

Esta seção apresenta a metodologia utilizada nesta pesquisa, na qual se pretendeu aplicar a Lógica  $E\tau$  nas contradições obtidas dos respondentes na avaliação da usabilidade do SGPS. Os procedimentos utilizados neste estudo tiveram o objetivo de ressaltar as contradições nos resultados.

A presente pesquisa é de natureza aplicada, pois pretende obter o conhecimento por meio de uma aplicação prática e orientada à solução de problemas na usabilidade de software, aspecto determinante na aprovação de um software, trazendo satisfação e melhoria de produtividade para o usuário (Harrison et al., 2013).

Este estudo é exploratório, pois pretende evidenciar a contradição por meio da aplicação da Lógica  $E\tau$  na avaliação de usabilidade, tendo por instrumento o SUS, que utiliza perguntas fechadas e padronizadas, empregando uma escala *Likert* (Stevens, 1936), com valores que variam de um a cinco. No caso, um significa “Discordo totalmente” e cinco, “Concordo totalmente”, equivalendo aos critérios numéricos atribuídos à escala, conforme Tabela 5 (Pereira et al., 2018).

Tabela 5 - Normalização de  $\mu$  e  $\lambda$

Resposta do Usuário	Score SUS	Valor Lógica $E\tau$
Concordo totalmente	5	1,00
Concordo	4	0,75
Indiferente	3	0,50
Discordo	2	0,25
Discordo totalmente	1	0,00

Fonte: Autor

Quanto ao procedimento adotado para esta pesquisa, utilizou-se um *survey*, uma vez que se almeja buscar informações diretamente com um grupo de interesse, com o propósito de registrar as opiniões dos usuários. Utilizou-se um instrumento de avaliação para uma população alvo de onze profissionais da área de Gerência de Prestadores de Serviço, que avaliaram a usabilidade do software em seu cotidiano. Foi mantido o anonimato dos participantes, que receberam o instrumento de avaliação por meio eletrônico. Tendo o cargo como critério de agrupamento, foram separados em três grupos: Product Owner (PO); Analistas de Controle (AC) e Operadores (OP). (Tabela 6). na base de dados, foram denominados  $E_1$ ,  $E_2$ , ...,  $E_{11}$ .

Tabela 6 - Classificação dos Grupos de usuários

Grupo	Ocupação	Número de entrevistados
A	Product Owner	3
B	Analista de Controle	4
C	Operador	4

Fonte: Autores

O SUS foi utilizado para coletar os dados primários. A Tabela 7 apresenta um fragmento da base de dados da avaliação da usabilidade. A atribuição das respostas foi realizada da seguinte forma: as respostas ímpares foram atribuídas para evidência favorável ( $\mu$ ) complementando o valor de  $\lambda$  com  $1 - \mu$  e as respostas pares, para a evidência desfavorável ( $\lambda$ ), complementando o valor de  $\mu$  com  $1 - \lambda$ . O valor resultante da evidência favorável e da evidência desfavorável são números reais, que variam no intervalo fechado  $[0;1]$ , ou seja, podem variar de 0 (ou 0%) a 1 (ou 100%) (ABE, 2015).

Para construção da base de dados, foi utilizada a seguinte regra, explicada neste exemplo: para a questão Q1 do Especialista 1, a resposta foi concordo, o que após a normalização, equivale ao valor 0,75 na Lógica  $E_\tau$ , atribuído a  $\mu$  e com  $\lambda$  complementar 0,25, formando o par (0,75; 0,25). A partir da aplicação dessa regra para todas as respostas dos especialistas, elaborou-se a base de dados, conforme amostra apresentada na Tabela 7. A base completa encontra-se no Capítulo 4.

Tabela 7 – Amostra da base de dados formada  $\mu$  e  $\lambda$  atribuídos pelos usuários

Questão	Product owner						Analista de Controle								Operador							
	E1		E2		E3		E4		E5		E6		E7		E8		E9		E10		E11	
	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ	μ	λ
Q1	0,75	0,25	0,75	0,25	1,00	0,00	1,00	0,00	1,00	0,00	1,00	0,00	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	0,75	0,25

Fonte: Autor

### 3.3. Aspectos Éticos e Legais

Em consonância com as determinações da Resolução 466/12 do Conselho Nacional de Saúde, este estudo foi submetido e aprovado pelo Comitê de Ética em Pesquisa da Unip, sob o nº 5.021.838, CAAE 52164521.0.0000.5512 (Apêndice 4).

Os participantes manifestaram concordância com ele, após o pesquisador explicar o estudo, assegurar a confidencialidade das informações, o anonimato e o direito de retirar-se a qualquer momento.

## **4 RESULTADOS E DISCUSSÃO**

Neste capítulo, para responder os objetivos específicos, são apresentados os resultados em forma de artigos científicos.

### **4.1 Artigo 1**

Os conceitos abordados neste artigo, que se encontra no Apêndice 1, com o título “Modelo de questionário para avaliação paraconsistente da qualidade de um software desenvolvido no Sales Force”, permitiram atender ao objetivo específico (B) da seção 1.3.1.

O texto sugere, como proposta, utilizar o método Paraconsistente de Decisão (MPD), para aprimorar o entendimento na análise de dados capturados em um questionário padronizado SUS - System Usability Scale. A avaliação com a Lógica  $E\tau$  permite aferir a qualidade da usabilidade do Software de Gerenciamento de Prestadores de Serviço, desenvolvido na plataforma Salesforce. Os dados obtidos por meio eletrônico foram normalizados antes da aplicação do algoritmo Para-analisador. Essa Lógica permite analisar as opiniões dos usuários, considerando as contradições e subjetividades inerentes aos valores humanos. Pretendeu-se, com esse algoritmo, obter um consenso das opiniões dos especialistas sobre a usabilidade do software. O estudo abriu a possibilidade de haver contradição em um instrumento de avaliação quantitativo, aprimorando a análise dos dados da usabilidade e contribuindo para melhoria do desenvolvimento de software. Esse artigo fez parte do congresso internacional APMS Conference e foi publicado no IFIP International Federation for Information Processing.

### **4.2 Artigo 2**

Os conceitos abordados nesse artigo presente no Apêndice 2, com o título “Aprimoramento do Processo de Avaliação de Histórias de Usuários utilizando a Lógica Paraconsistente Anotada Evidencial  $E\tau$ ”, permitiram atender ao objetivo específico (A) da seção 1.3.1.

Além de demonstrar a aplicação da Lógica  $E\tau$  no processo de avaliação, priorização e estimativas das HUs, o estudo ampliou o conhecimento sobre a análise das contradições que podem ocorrer entre o PO e o Time, resultando em avaliações discordantes. Para isso, foi realizado um survey, com uma equipe de desenvolvedores que atuam com métodos ágeis. Os graus de evidência favorável e de evidência desfavorável coletados para cada critério INVEST são utilizados como variáveis de entrada para o Para-analisador. Ele destaca a relevância das questões, no apoio na tomada de decisão, com base em um modelo matemático.

Foram constatadas, na análise global, duas HUs que não poderiam ser desenvolvidas em um Sprint, portanto, deveriam ser refatoradas e reavaliadas pelos especialistas. As outras duas histórias possuíam grau de evidências favoráveis apropriadas para serem desenvolvidas em um Sprint.

Esse artigo foi aceito para o XL Congresso Encontro Nacional de Engenharia de Produção (ENEGEP 2020), em Foz do Iguaçu, Paraná, Brasil.

A Tabela 8 exibe a base de dados formada pelos graus de evidência favorável e de evidência desfavorável atribuídos pelos especialistas a cada fator de análise da História de Usuário A, descrita no Quadro 3, no Capítulo 3.

Tabela 8 - Base de dados História de Usuário A

Fator	Grupo A						Grupo B						Grupo C					
	E1		E2		E3		E4		E5		E6		E7		E8		E9	
	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$
F01	0,25	0,75	0,25	1,00	0,50	0,75	0,25	0,75	0,00	0,75	0,25	0,75	0,25	0,50	0,00	1,00	0,25	1,00
F02	0,50	0,50	0,75	0,00	0,75	0,25	1,00	0,00	0,75	0,00	0,75	0,25	1,00	0,25	0,75	0,25	0,75	0,00
F03	1,00	0,00	1,00	0,25	1,00	0,00	1,00	0,25	0,75	0,25	1,00	0,00	0,75	0,00	1,00	0,25	1,00	0,00
F04	0,75	0,25	0,00	0,75	0,00	1,00	0,25	0,75	0,25	0,75	0,00	1,00	0,25	1,00	0,25	0,75	0,25	1,00
F05	1,00	0,00	0,00	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,00	0,75	0,25	1,00	0,25	0,75	0,25	1,00
F06	1,00	0,00	0,75	0,50	0,75	0,25	1,00	0,00	1,00	0,50	1,00	0,25	1,00	0,75	1,00	0,25	0,75	0,25

Fonte: Autor

A Tabela 9 exibe a base de dados formada pelos graus de evidência favorável e evidência desfavorável atribuídos pelos especialistas para cada fator de análise da História de Usuário B descrita no Quadro 3, no Capítulo 3.

Tabela 9 - Base de dados História de Usuário B

Fator	Grupo A						Grupo B						Grupo C					
	E1		E2		E3		E4		E5		E6		E7		E8		E9	
	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$
F01	1,00	0,00	0,75	0,00	1,00	0,25	1,00	0,00	0,75	0,00	1,00	0,25	1,00	0,25	0,75	0,25	0,75	0,25
F02	1,00	0,00	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,25	0,75	0,50	0,75	0,25
F03	0,50	0,00	1,00	0,25	1,00	0,25	1,00	0,25	0,75	0,25	0,75	0,25	1,00	0,00	1,00	0,25	1,00	0,25
F04	1,00	0,00	1,00	0,00	1,00	0,00	0,75	0,25	1,00	0,25	0,75	0,00	1,00	0,00	0,75	0,25	0,75	0,25
F05	1,00	0,00	0,75	0,25	0,75	0,00	1,00	0,00	1,00	0,00	0,75	0,25	0,75	0,25	0,75	0,00	1,00	0,00
F06	1,00	0,00	0,75	0,25	0,75	0,50	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,25	0,75	0,25

Fonte: Autor

A Tabela 10 exibe a base de dados formada pelos graus de evidência favorável e evidência desfavorável atribuídos pelos especialistas a cada fator de análise da História de Usuário C, descrita no Quadro 3, Capítulo 3.

Tabela 10 - Base de dados História de Usuário C

Fator	Grupo A						Grupo B						Grupo C					
	E1		E2		E3		E4		E5		E6		E7		E8		E9	
	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$
F01	0,50	0,00	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,50	1,00	0,25	1,00	0,25	0,75	0,50	1,00	0,25
F02	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
F03	1,00	0,00	0,75	0,25	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,25	1,00	0,25	1,00	0,25	1,00	0,25
F04	0,50	0,50	1,00	0,25	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,00	1,00	0,25	0,75	0,25	0,75	0,25
F05	1,00	0,00	1,00	0,25	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,25	0,75	0,25	1,00	0,00
F06	1,00	0,00	0,75	0,25	0,75	0,25	1,00	0,25	1,00	0,25	1,00	0,00	0,75	0,25	1,00	0,25	1,00	0,25

Fonte: Autor

A Tabela 11 exibe a base de dados formada pelos graus de evidência favorável e evidência desfavorável atribuídos pelos especialistas a cada fator de análise da História de Usuário D descrita no Quadro 3, Capítulo 3.

Tabela 11 - Base de dados História de Usuário D

Fator	Grupo A						Grupo B						Grupo C					
	E1		E2		E3		E4		E5		E6		E7		E8		E9	
	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$
F01	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
F02	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
F03	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
F04	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
F05	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
F06	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

Fonte: Autor

Para mais informações sobre resultados e discussão, este artigo encontra-se no Apêndice 2.

### 4.3 Artigo 3

Este artigo, que se encontra no Apêndice 3, com o título “Escala de Usabilidade do Sistema Baseada na Lógica Paraconsistente Anotada Evidencial  $E\tau$ ”, traz os resultados da aplicação da Lógica  $E\tau$  na análise da Escala de Usabilidade do Sistema, mensurando a experiência do usuário final com o sistema. Dessa forma, atende aos objetivos específicos (B). O artigo pretende apresentar um aprofundamento das contradições em uma avaliação de usabilidade de software, aplicando a Lógica Paraconsistente Anotada Evidencial  $E\tau$  e realizando uma pesquisa sobre a avaliação de um sistema com onze profissionais. Foi feita uma comparação dos resultados do algoritmo Para-analisador com a escala SUS, utilizando o par  $(\mu; \lambda)$ , atribuindo as questões pares para  $\mu$  e as questões ímpares, para  $\lambda$ .

O resultado da Escala de Usabilidade do Sistema (76,59) recebeu avaliação positiva em usabilidade. O Para-analisador apresentou, como resultado, a média global (0,75; 0,25), o que demonstra similaridade com o primeiro, salientando dois itens contraditórios. A análise da usabilidade foi aprimorada com essa avaliação, pois houve a concordância dos respondentes em nove características da usabilidade. Entretanto, houve divergência em um item, que apresentou uma contradição na avaliação, sendo duas questões contraditórias avaliadas com valores altos por cinco respondentes. Esse resultado pode indicar carência de treinamento dos participantes antes do início do processo ou falha de percepção na semântica do instrumento. A principal contribuição desse trabalho foi avaliar a usabilidade de um software em conjunto com o Para-analisador, circunstância que permitiu uma análise mais minuciosa, já que ele trabalha com o par  $(\mu; \lambda)$ , o que permite avaliar os resultados em duas dimensões.

Este artigo foi publicado na revista Research, Society and Development Journal - ISSN 2525-3409.

A

Tabela 12 apresenta os valores coletados no SUS e que foram normalizados, aplicando o critério apresentado na Tabela 5 e descrito no Capítulo 3.

Tabela 12 - Base de dados resultado da normalização

Questão	Product Owner						Analista de Controle						Operador									
	E1		E2		E3		E4		E5		E6		E7		E8		E9		E10		E11	
	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$
Q1	0,75	0,25	0,75	0,25	1,00	0,00	1,00	0,00	1,00	0,00	1,00	0,00	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	0,75	0,25
Q2	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,00	1,00	0,00	1,00	0,00	0,25	0,75	0,75	0,25
Q3	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,00	1,00	0,00	1,00	0,00	1,00	0,00	1,00	0,00	0,75	0,25	0,75	0,25
Q4	0,75	0,25	0,75	0,25	1,00	0,00	0,25	0,75	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	0,50	0,50	0,75	0,25	0,75	0,25
Q5	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,00	0,75	0,25	0,75	0,25	0,50	0,50	0,75	0,25	0,75	0,25	0,75	0,25
Q6	0,25	0,75	0,25	0,75	0,25	0,75	0,00	1,00	0,00	1,00	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25
Q7	0,25	0,75	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	0,75	0,25
Q8	0,75	0,25	1,00	0,00	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	0,75	0,25
Q9	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	0,25	0,75	1,00	0,00	0,75	0,25	0,75	0,25	1,00	0,00	1,00	0,00	0,75	0,25
Q10	0,25	0,75	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,00	0,75	0,25	0,75	0,25	1,00	0,00	1,00	0,00	0,75	0,25	0,25	0,75

Fonte: Autor

Mais informações sobre resultados e discussão encontram-se disponíveis no artigo, no Apêndice 3.

## 5 CONSIDERAÇÕES FINAIS

Este estudo cumpriu seu objetivo de evidenciar a relevância e ampliar a possibilidade do uso da Lógica  $E\tau$ , para aprimorar instrumentos utilizados em dois subprocessos de software: a engenharia de requisitos e a Interface Homem Computador. Dessa forma, contribuiu com um modelo para auxiliar no processo decisório nesses dois subprocessos importantes para a Engenharia de Software, com a utilização da Lógica Paraconsistente Anotada Evidencial  $E\tau$  para buscar benefícios no tratamento direto de informações contraditórias, imprecisas e/ou conflitantes, podendo ser manipuladas por meio do algoritmo Para-analisador que lhe concede critérios lógicos e matemáticos na análise dos parâmetros de entrada definidos pela estrutura do pensamento dos especialistas, consolidando as opiniões em uma lógica coletiva.

O diferencial desta pesquisa está no uso da Lógica  $E\tau$  para auxiliar na tomada de decisão desses dois subprocessos, ampliando as possibilidades de análise e, consequentemente, a gestão.

Dentro do subprocesso Engenharia de Requisitos, no modelo Ágil, não se leva em consideração a inconsistência ou a contradição nos processos atuais de validação

de Histórias de Usuários. Na vida real, em razão das condições do ambiente em que foi realizado o levantamento de requisitos, apresentaram-se tanto a contradição como a inconsistência. Independente da vontade dos membros da equipe de desenvolvimento de software ou da área de negócio, as situações de contradição aparecem, demonstrando que a existência de conflitos é inerente do subprocesso da Engenharia de Requisitos. Quanto maior o envolvimento do *Product Owner*, do time de desenvolvimento e dos *stakeholders* para solução dos desejos, necessidades e interesses do cliente, maiores os níveis de contradição, conflito e inconsistência,

O presente trabalho apresenta um modelo para apoiar na tomada de decisão da avaliação das Histórias de Usuários, utilizando a Lógica  $E\tau$ , por meio do algoritmo Para-analisador, aplicando critérios lógicos que permitam uma validação técnica. Os parâmetros de entrada são definidos pelas opiniões dos especialistas, sugerindo que os conceitos da Lógica Paraconsistente Anotada Evidencial  $E\tau$  podem ser aplicados, para apresentar e validar perspectivas de lidar com situações de inconsistência e indefinição. Essas equações ou fatores são importantes para apoiar a tomadas de decisão e influenciam no sucesso e qualidade de um sistema de informações.

Dentro do subprocesso Interface Homem Computador, os processos atuais de avaliação de usabilidade não levam em conta a contradição ou a inconsistência. Dependendo das condições do ambiente em que foi aplicado o questionário, ambas aparecem, demonstrando que a existência de conflitos faz parte do processo de avaliação de usabilidade. Quanto maior o comprometimento dos especialistas com o software para resolução de seus processos de trabalho no dia a dia, maior o nível de contradição, inconsistência e conflitos a que estão sujeitos.

Este trabalho apresenta um modelo paraconsistente para apoiar a tomada de decisão na avaliação da usabilidade utilizando a Lógica  $E\tau$ , aplicando o algoritmo Para-analisador, que utiliza critérios lógicos, que proporcionam uma validação técnica. As opiniões dos especialistas são usadas como parâmetros de entrada. Esses valores são consolidados em uma lógica coletiva, convertida em termos matemáticos. O modelo paraconsistente baseia-se nas perspectivas que proporcionam satisfação aos especialistas, auxiliando no processo decisório de avaliação da usabilidade por meio da Lógica  $E\tau$ . Eles podem ser utilizados, para validar e apresentar uma nova técnica de lidar com opiniões dos especialistas em situações de indefinição ou mesmo inconsistência na utilização do software. As possíveis situações futuras são relevantes para a tomada de decisões que garantam a qualidade da avaliação da usabilidade e



o sucesso de um software. Tanto a análise, quanto seu resultado serão apresentadas em um futuro artigo.

As respostas do instrumento de avaliação da usabilidade de software foram adaptadas para a Lógica  $E\tau$ , com o par  $(\mu, \lambda)$ . Para respostas pares, foram atribuídas para  $\lambda$ . Para as ímpares, foram atribuídas para  $\mu$ , devido ao fato das questões do SUS se contradizarem.

Na avaliação dos usuários, o SGPS obteve, como resultado, uma usabilidade boa. Acredita-se que não foi obtida uma pontuação melhor por falta de discernimento semântico das questões do instrumento de avaliação. Presume-se que seja necessário um treinamento do SUS com os usuários, antes de sua aplicação.

A avaliação crítica da usabilidade do SGPS realizada pelos usuários junto à Lógica  $E\tau$  proporcionou o aprimoramento da análise da usabilidade, pois demonstrou divergência em uma característica e concordância em nove. Portanto, não se pode aplicar a Lógica  $E\tau$  em instrumentos de avaliação com o propósito de explorar as contradições na análise realizada com o público-alvo.

### 6.1. Sugestões de Trabalhos Futuros

Como sugestão de novos projetos, viu-se a necessidade de pesquisar e desenvolver os seguintes temas, para sequência desta dissertação:

- A) Criar uma base histórica das avaliações das Histórias de Usuários, que pode ser alimentada com informações de HUs de vários projetos, para servir de fonte de informação para as reuniões de *Sprint Backlog*.
- B) Explorar a aplicação de outras lógicas não-clássicas para tomada de decisão nos processos de Engenharia de Requisitos e Interface Homem Computador.
- C) Realizar uma pesquisa envolvendo a utilização direta dos métodos ágeis com a usabilidade, explorando a convergência entre as subáreas Engenharia de Requisitos e Interface Homem Computador.
- D) Criar uma ferramenta baseada no estudo realizado nesta dissertação, com a intenção de automatizar o suporte à tomada de decisão com o uso de técnicas de inteligência artificial baseadas na Lógica Paraconsistente Anotada Evidencial  $E\tau$ .

## REFERÊNCIAS

25000, I. E. C. **ISO/IEC 25000:2014 Software engineering - System and software Quality Requirements and Evaluation (SQuaRE)**. [S.l.]. 2014.

ABE, J. M. **Paraconsistent intelligent-based systems: New trends in the applications of paraconsistency**. [S.l.]: Springer, v. 94, 2015.

ABE, J. M. **Tópicos de Sistemas Inteligentes Baseados em Lógicas Não Clássicas**. 1. ed. [S.l.]: Instituto de Estudos Avançados da USP, v. 197 p., 2016.

ABE, J. M.; AKAMA, S.; NAKAMATSU, K. **Introduction to Annotated Logics: Foundations for Paracomplete and Paraconsistent Reasoning**. Cham: Springer International Publishing, 2015. 1–4 p. Disponível em: <[https://doi.org/10.1007/978-3-319-17912-4\\_1](https://doi.org/10.1007/978-3-319-17912-4_1)>.

ABE, J. M.; DA SILVA FILHO, J. I.; CELESTINO, U.; ARAÚJO, H. C. D. **Lógica Paraconsistente Anotada Evidencial Et**. Santos: Comunicar, 2011. ISBN ISBN: 978-85-99561-92-8.

ABE, J. M.; NAKAMATSU, K.; AKAMA, S.; AHRARY, A. **An Expert System for Uncertain, Inconsistent, and Paracomplete Data Decision-Making, ICEST 2018 - 53rd International Scientific Conference on Information, Communication and Energy Systems and Technologies – Sozopol, Bulgaria, June 28 – 30, 2018**. [S.l.]: [s.n.]. jun. 2018.

ABE, J. M.; SILVA LOPES, H. F.; ANGHINAH, R. Paraconsistent Neurocomputing and Biological Signals Analysis. In: ABE, J. M. (Ed.). **Paraconsistent Intelligent-Based Systems**. Intelligent Systems Reference Library. **Springer International Publishing**, p. 94p. 273–306, 2016.

AKAMA, S. (Ed.). **Towards Paraconsistent Engineering**. [S.l.]: Springer International Publishing, v. 110, 2016. Disponível em: <<https://doi.org/10.1007/978-3-319-40418-9>>.

BARBOSA, R.; SILVA, A. E. A.; MORAES, R. **Use of Similarity Measure to Suggest the Existence of Duplicate User Stories in the Srum Process**. 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W). [S.l.]: [s.n.]. jun. 2016. p. 2-5.

BIK, N.; LUCASSEN, G.; BRINKKEMPER, S. **A Reference Method for User Story Requirements in Agile Systems Development**. 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). [S.l.]: IEEE. set. 2017.

BROOKE, J. SUS: A quick and dirty usability scale. **Usability Eval. Ind.**, v. 189, nov. 1995.

CARVALHO, F. R.; ABE, J. M. **A Paraconsistent Decision-Making Method**. [S.l.]: Springer International Publishing, v. 87, 2018.

CASTILLO-BARRERA, F. E.; AMADOR-GARCIA, M.; PEREZ-GONZALEZ, H. G.; MARTINEZ-PEREZ, F. E.; TORRES-REYES, F. J. **Adapting Bloom's Taxonomy for an Agile Classification of the Complexity of the User Stories in SCRUM**. 2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT). [S.l.]: [s.n.]. out. 2018. p. 139-145.

CASTILLO-BARRERA, F. E.; AMADOR-GARCIA, M.; PEREZ-GONZALEZ, H.; MARTINEZ-PEREZ, F. E. **Agile Evaluation of the Complexity of User Stories Using the Bloom's Taxonomy**. 2017 International Conference on Computational Science and Computational Intelligence (CSCI). [S.l.]: [s.n.]. dez. 2017. p. 1047-1050.

CHAUHAN, V. K. **How to reduce user story reopen count in Scrum development?** 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom). [S.l.]: [s.n.]. mar. 2015. p. 2204-2208.

CHONGPAKDEE, P.; VATANAWOOD, W. **Estimating user story points using document fingerprints**. 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS). [S.l.]: [s.n.]. nov. 2017. p. 149-152.

CHOPADE, M. R. M.; DHAVASE, N. S. **Agile software development: Positive and negative user stories**. 2017 2nd International Conference for Convergence in Technology (I2CT). [S.l.]: IEEE. abr. 2017.

CHOUDHARY, B.; RAKESH, S. K. **An approach using agile method for software development**. 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH). [S.l.]: [s.n.]. 2016. p. 155-158.

COCKBURN, A. **Agile Software Development: The Cooperative Game**. [S.l.]: Addison-Wesley Professional, 2006. ISBN ISBN: 978-0-321-48275-4.

CORREIA, A.; GONÇALVES, A.; MISRA, S. Integrating the Scrum Framework and Lean Six Sigma. In: \_\_\_\_\_ **Computational Science and Its Applications – ICCSA 2019**. [S.l.]: Springer International Publishing, 2019. p. 136–149.

CURCIO, K.; SANTANA, R.; REINEHR, S.; MALUCELLI, A. Usability in agile software development: A tertiary study. **Computer Standards & Interfaces**, v. 64, p. 61–77, maio 2019.

DA SILVA FILHO, J. I. **Métodos de Aplicações da Lógica Paraconsistente Anotada com Dois Valores. LPA2v com Construção de Algoritmo e Implementação de Circuitos Eletrônicos.** [S.l.]: São Paulo: [s.n.], 1999.

DERAMAN, A. B.; SALMAN, F. A. Managing usability evaluation practices in agile development environments. **International Journal of Electrical and Computer Engineering (IJECE)**, v. 9, p. 1288, abr. 2019.

DREW, M. R.; FALCONE, B.; BACCUS, W. L. What Does the System Usability Scale (SUS) Measure? In: \_\_\_\_\_ **Design, User Experience, and Usability: Theory and Practice.** [S.l.]: Springer International Publishing, 2018. p. 356–366.

ESAKI, K.; AZUMA, M.; KOMIYAMA, T. Introduction of Quality Requirement and Evaluation Based on ISO/IEC SQuaRE Series of Standard. In: \_\_\_\_\_ **Trustworthy Computing and Services.** [S.l.]: Springer Berlin Heidelberg, 2013. p. 94–101.

FITRIANI, W. R.; RAHAYU, P.; SENSUSE, D. I. **Challenges in agile software development: A systematic literature review.** 2016 International Conference on Advanced Computer Science and Information Systems (ICACISIS). [S.l.]: [s.n.]. 2016. p. 155-164.

GHANBARI, H.; VARTIAINEN, T.; SIPONEN, M. Omission of Quality Software Development Practices: A Systematic Literature Review. **ACM Comput. Surv.**, New York, NY, USA, v. 51, fev. 2018. ISSN ISSN: 0360-0300. Disponível em: <<https://doi-org.ez338.periodicos.capes.gov.br/10.1145/3177746>>.

GHOZALI, R. P.; SAPUTRA, H.; NURIAWAN, M. A.; SUHARJITO; UTAMA, D. N.; NUGROHO, A. Systematic Literature Review on Decision-Making of Requirement Engineering from Agile Software Development. **Procedia Computer Science**, v. 157, p. 274–281, 2019.

HAYAT, F.; REHMAN, A. U.; ARIF, K. S.; WAHAB, K.; ABBAS, M. **The Influence of Agile Methodology (Scrum) on Software Project Management.** 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). [S.l.]: IEEE. jul. 2019.

HEIKKILÄ, V.; PAASIVAARA, M.; LASSENIUS, C.; DAMIAN, D. A Mapping Study on Requirements Engineering in Agile Software Development, ago. 2015.

HESS, A.; DIEBOLD, P.; SEYFF, N. **Towards Requirements Communication and Documentation Guidelines for Agile Teams.** 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). [S.l.]: IEEE. set. 2017.

HORKOFF, J. **The Influence of Agile Methods on Requirements Engineering Courses**. 2018 IEEE 8th International Workshop on Requirements Engineering Education and Training (REET). [S.l.]: IEEE. ago. 2018.

INAYAT, I.; SALIM, S. S.; MARCZAK, S.; DANEVA, M.; SHAMSHIRBAND, S. A systematic literature review on agile requirements engineering practices and challenges. **Computers in Human Behavior**, v. 51, p. 915-929, 2015. ISSN: 0747-5632. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S074756321400569X>>. Computing for Human Learning, Behaviour and Collaboration in the Social and Mobile Networks Era.

JAIN, R.; GUPTA, D.; KHANNA, A. Usability Feature Optimization Using MWOA. In: \_\_\_\_\_ **International Conference on Innovative Computing and Communications**. [S.l.]: Springer Singapore, 2018. p. 453–462.

JARZĘBOWICZ, A.; POŁOCKA, K. **Selecting Requirements Documentation Techniques for Software Projects: a Survey Study**. Proceedings of the 2017 Federated Conference on Computer Science and Information Systems. [S.l.]: IEEE. set. 2017.

KABIR, M. A.; REHMAN, M. U.; MAJUMDAR, S. I. **An analytical and comparative study of software usability quality factors**. 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS). [S.l.]: [s.n.]. 2016. p. 800-803.

KAMTHAN, P.; SHAHMIR, N. **Modeling Negative User Stories is Risky Business**. 2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE). [S.l.]: IEEE. jan. 2016.

KNIBERG, H. **Scrum and XP from the Trenches (Enterprise Software Development)**. 2ª. ed. [S.l.]: lulu.com; Annotated edition, 2015.

KOMAI, S.; NAKANISHI, H.; SAIDI, H. **Guidelines for selecting agile development method in system requirements definition**. 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE). [S.l.]: IEEE. nov. 2017.

LEWIS, J. R. The System Usability Scale: Past, Present, and Future. **International Journal of Human–Computer Interaction**, v. 34, p. 577–590, mar. 2018.

LISKIN, O.; PHAM, R.; KIESLING, S.; SCHNEIDER, K. **Why We Need a Granularity Concept for User Stories**. [S.l.]: [s.n.]. maio 2014. p. 110-125.

LORCA, A. L.; BURROWS, R.; STERLING, L. **Teaching Motivational Models in Agile Requirements Engineering**. 2018 IEEE 8th International Workshop on Requirements Engineering Education and Training (REET). [S.l.]: IEEE. ago. 2018.

LUCASSEN, G.; DALPIAZ, F.; VAN DER WERF, J. M. E. M.; BRINKKEMPER, S. **Forging high-quality User Stories: Towards a discipline for Agile Requirements.** 2015 IEEE 23rd International Requirements Engineering Conference (RE). [S.l.]: IEEE. ago. 2015.

MARQUES, A. B.; FIGUEIREDO, R.; AMORIN, W.; RABELO, J.; BARBOSA, S. D. J.; CONTE, T. **Do Usability and Agility Combine?** Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems. [S.l.]: ACM. out. 2018.

MATEEN, A.; ABBAS, K.; AKBAR, M. A. Robust approaches, techniques and tools for requirement engineering in agile development, p. 100-103, set. 2017.

OLAVERRI-MONREAL, C.; HASAN, A. E.; BENGLER, K. **Semi-automatic user stories generation: To measure user experience.** 2013 8th Iberian Conference on Information Systems and Technologies (CISTI). [S.l.]: [s.n.]. jun. 2013. p. 1-6.

OPAZO, J. G.; LOPEZ, C. **Improving User Stories: A Case Study in the Chilean Banking Industry.** 2018 XLIV Latin American Computer Conference (CLEI). [S.l.]: IEEE. out. 2018.

ORMSBY, M.; BUSBY-EARLE, C. **A Standardized Procedure to Conceptualizing and Completing User Stories.** 2017 International Conference on Computational Science and Computational Intelligence (CSCI). [S.l.]: IEEE. dez. 2017.

PMI. **A guide to the project management body of knowledge (PMBOK guide 6ª. Ed.).** Sixth edition. ed. [S.l.]: Project Management Institute, 2017. ISBN ISBN: 9781628251845.

POPLI, R.; CHAUHAHN, N. **Managing uncertainty of story-points in Agile software.** 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom). [S.l.]: [s.n.]. mar. 2015. p. 1357-1361.

PRESSMAN, R. S. **Software engineering: a practitioner's approach.** 7. ed. [S.l.]: McGraw-Hill Publishing Co., 2011. ISBN ISBN: 0073375977 /9780073375977.

RAHARJANA, I. K.; SIAHAAN, D.; FATICHAH, C. **User Story Extraction from Online News for Software Requirements Elicitation: A Conceptual Model.** 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE). [S.l.]: [s.n.]. jul. 2019. p. 342-347.

RODEGHERO, P.; JIANG, S.; ARMALY, A.; MCMILLAN, C. **Detecting User Story Information in Developer-Client Conversations to Generate Extractive Summaries.** 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE). [S.l.]: [s.n.]. maio 2017. p. 49-59.

SAGAR, K.; SAHA, A. A systematic review of software usability studies. **International Journal of Information Technology**, dez. 2017.

SATRIA, D.; SENSUSE, D. I.; NOPRISSON, H. **A systematic literature review of the improved agile software development**. 2017 International Conference on Information Technology Systems and Innovation (ICITSI). [S.l.]: IEEE. out. 2017.

SCHWABER, K. & S. J. **The scrum guide. The Definitive Guide to Scrum: The rules of the game**. [S.l.]: This publication is offered for license under the Attribution Share-Alike license of Creative Commons. Disponível em <https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100>. Acesso em 01/12/2020, 2020.

SHARMA, S.; HASTEER, N. **A comprehensive study on state of Scrum development**. 2016 International Conference on Computing, Communication and Automation (ICCCA). [S.l.]: IEEE. abr. 2016.

SHEN, P.; DING, X.; REN, W.; YANG, C. **Research on Software Quality Assurance Based on Software Quality Standards and Technology Management**. Busan: IEEE. 2018. p. 385–390.

SILVA FILHO, J. I.; TORRES, G. L.; ABE, J. M. Uncertainty treatment using paraconsistent logic: introducing paraconsistent artificial neural networks. **Amsterdam; Washington, DC: IOS Press**, 2010.

SITTHITHANASAKUL, S.; CHOOSRI, N. **Using ontology to enhance requirement engineering in agile software process**. 2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA). [S.l.]: IEEE. 2016.

SOMMERVILLE, I. **Software Engineering**. [S.l.]: Pearson, 2010.

SOUZA, S.; ABE, J. M. Paraconsistent Artificial Neural Networks and Aspects of Pattern Recognition. In: ABE, J. M. (Ed.). Paraconsistent Intelligent-Based Systems. Intelligent Systems Reference Library. **Springer International Publishing**, p. v. 94p. 207–231., 2015.

SUTHERLAND, J. **Scrum: The Art of Doing Twice the Work in Half the Time**. [S.l.]: Crown Business Publications, 2015.

SWEBOK. **Guide to the Software Engineering Body of Knowledge Version 3.0**. [S.l.]: [s.n.], 2014. ISBN ISBN: 0-7695-5166-1.

TORRES, C. R.; REIS, R. The New Hardware Structure of the Emmy II Robot. In: ABE, J. M. (Ed.). Paraconsistent Intelligent-Based Systems. Intelligent Systems Reference Library. **Springer International Publishing**, p. v. 94p. 87–103., 2015.

UMAR, M. A.; TENUCHE, S. S.; YUSUF, S. A.; ABDULSALAMI, A. O.; KUFENA, A. M. Usability Engineering in Agile Software Development Processes. In: \_\_\_\_\_ **Advances in Systems Analysis, Software Engineering, and High Performance Computing**. [S.l.]: IGI Global, 2016. p. 208–221.

VILLAMIZAR, H.; NETO, A. A.; KALINOWSKI, M.; GARCIA, A.; MENDEZ, D. **An Approach for Reviewing Security-Related Aspects in Agile Requirements Specifications of Web Applications**. 2019 IEEE 27th International Requirements Engineering Conference (RE). [S.l.]: IEEE. set. 2019.

WAUTELET, Y.; HENG, S.; KOLP, M.; MIRBEL, I.; POELMANS, S. **Building a rationale diagram for evaluating user story sets**. 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS). [S.l.]: IEEE. jun. 2016.

ZEAARAOU, A.; BOUGROUN, Z.; BELKASMI, M. G.; BOUCHENTOUF, T. **User stories template for object-oriented applications**. Third International Conference on Innovative Computing Technology (INTECH 2013). [S.l.]: IEEE. ago. 2013.



## APÊNDICES

### Apêndice 1

Este artigo foi escrito seguindo as normas do congresso internacional APMS Conference, no qual foi aprovado.

- FORÇAN, L. R.; ABE, J. M.; LIMA A. L.; NASCIMENTO, S. S., Questionnaire Model for Paraconsistent Quality Assessment of Software Developed in Salesforce. IFIP International Federation for Information Processing., v.591, p.333 - 340, 2020.  
[http://https://link.springer.com/chapter/10.1007%2F978-3-030-57993-7\\_38](http://https://link.springer.com/chapter/10.1007%2F978-3-030-57993-7_38). doi:10.1007 / 978-3-030-57993-7\_38]

## Questionnaire Model for Paraconsistent Quality Assessment of Software Developed in Salesforce

Luiz Roberto Forçan<sup>[0000-0001-7376-830X]</sup>, Jair Minoro Abe<sup>[0000-0003-2088-9065]</sup>,  
Luiz Antônio de Lima<sup>[0000-0003-4228-2387]</sup>, Samira Sestari Nascimento<sup>[0000-0001-7429-6033]</sup>

Paulista University, São Paulo, Brazil  
luforcan@gmail.com.br, jairabe@uol.com.br,  
luiz@wcisp.com.br, samirasestari@gmail.com

**Abstract.** The article suggests the use of the Paraconsistent Decision Method (MPD) to improve the analysis of data captured in a standardized SUS - System Usability Scale questionnaire. The paraconsistent evaluation allows you to measure the usability quality of the provider registration software developed on the Salesforce platform. The data obtained through the questionnaire must be processed to be submitted to the Para-analyzer algorithm of The Paraconsistent Annotated Logic Evidential Et - Logic Et. Logic Et is used as a non-classical logic that allows analyzing the opinions of users considering their uncertainties, inaccuracies, ambiguities, and subjectivities that are inherent to human values. Through the Para-analyzer algorithm, we intend to obtain a consensus of the opinions of the experts on the usability of the software. The model with Logic Et allows being used in addition to the statistical treatment provided by the SUS method, improving the analysis of the data. With the result of this analysis, it is intended to diagnose usability problems contributing to the improvement of software development

**Keywords:** Software Usability Quality, Paraconsistent Logic Annotated Et, Cloud Computing, Salesforce.

### 1 Introduction

To be successful in this competitive market one vital point is the usability assessment. It is essential to measure the degree of user satisfaction that uses the software product developed in the Salesforce tool in your daily life.

There are several approaches to assess the usability of the software: the System Usability Scale (SUS) questionnaire is widely used for usability testing, reliability, learning capacity, appearance, and other aspects. A newer scale for measuring usability is the usability metric for user experience with few items that determine reliability and usefulness.

For the present study, we will use the SUS questionnaire format, and the main advantages of using these instruments for distance evaluation are objectivity in the collection of information, replicability of the instrument in other studies and quantification of results from the participants' responses. [11].

Logic Et because it is a non-classical logic, allows the analysis of the subjective opinions of experts considering their uncertainties, inaccuracies, ambiguities, and subjectivities that are inherent to human values. [2] Other approaches allow you to do this analysis, such as the AHP Analytic Hierarchy Process. It is observed that the AHP method, although with peculiarities that aim to solve these possible problems, is based on the classical methods of decision-making, which consider the human opinions accurate and well defined (false and true). [7] [14].

Considering that it is proposed of the ISO/IEC 25010 standard [12] complements the study as qualitative attributes distributed in eight main characteristics: Functional adequacy, Performance efficiency, compatibility, usability, reliability, safety, maintenance, and portability. The standard recommends the qualitative feature of usability, which is divided into six sub characteristics: Accessibility, Apprehension, Aesthetics of the user interface, Operability, Protection against user errors, and Appropriate Recognition.

This article presents the results of a survey, applying the SUS questionnaire, to the specialists who use the software registration of providers, measuring the favorable and contrary evidence for each question of the questionnaire.

The research presents a model to assist in the decision-making process of evaluating the data collected on usability using Logic Et, through the Para-analyzer algorithm, using logical criteria that enable technical validation, and the input parameters are established by the opinions of the experts, consolidating a collective logic of these opinions, converted into mathematical terms.

## 2 Literature Revision

### 2.1 SalesForce

Salesforce (SF), which stands for "SalesForce" was founded in 1999 in California, USA by Marc Benioff and Parker Harris, and its main predominant point is the customer relationship management or Customer Relationship Management (CRM) system. It proposes to offer a platform in the clouds where the entire computational resource is located. [9] The SF platform has become attractive with the relationship management solution due to outsourcing CRM on the Internet, providing the CRM Web Service where it offers customizable, easy-to-implement software that easily engages the user, ensuring data integrity and security. [15]

### 2.2 Software Usability

According to the iso/IEC 25010 usability standard is the ability of the software to be easily operated, understood, easy to learn, and with an attractive user interface. ISO/IEC25010 [12] defines the following usability characteristics of software product quality: Accessibility, Apprehension, User Interface Aesthetics, Operability, User Error Protection, and Intelligibility or appropriate recognition.

To evaluate the usability of software, questionnaires such as SUMI, SUS, WAMMI, SUPR-Q, CEG, and NPS are used. With these questionnaires, usability, reliability, and

learning capacity and appearance tests are performed in software. For this article, we opted for the System Usability Scale -SUS questionnaire because it is the most popular choice for evaluating software usability.[11]

### 2.3 System Usability Scale (SUS)

The Usability Scale System - SUS [8] questionnaire is used to assess the level of usability of the system. The SUS scale consists of 10 specific questions to allow an analysis of the usability and ease of learning characteristics of a system. Each question has five answer options that follow the 5-point Likert scale: from 1 (I totally disagree) to 5 (I totally agree) where 3 means neutrality. This instrument allows objectivity in the collection of information and evaluation of the results of the experts' answers, through statistical calculations.[6] [11]

### 2.4 Analytical Hierarchical Process (AHP)

The Analytical Hierarchical Process (AHP) method is used to assist in decision making, evaluating multiple objectives and criteria in problems characterized by complexity and subjectivity. Alternatives are evaluated in the face of a complex decision problem. [16]

The AHP assumes that the decision-makers can provide accurate answers by comparing criteria and alternatives. However, due to the uncertainty, incompleteness, and subjectivity of the information, it is difficult to provide accurate judgments.[7] [14]

### 2.5 Paraconsistent Annotated Evidential Logic $E\tau$

Paraconsistent logic is a category on non-classical logics that the Contradiction (or Non-contradiction) Principle is not valid in general. So, in this type of logic, theories based on it, there are propositions  $p$  and  $\neg p$  (the negation of  $p$ ) both true [1]. One category of paraconsistent logic, namely the paraconsistent annotated evidential logic  $E\tau$ , besides being paraconsistent, it is also paracomplete. Such logic is capable of handling inconsistent, imprecise, and paracomplete data.

In everyday life, the data from various sources can be contradictory, making room for uncertainties, resulting in contradictions that prevent decision-making.

In the Paraconsistent annotated evidential Logic  $E\tau$ , each Proposition  $P$  associates a Favorable Degree of Evidence ( $\mu$ ) and a Contrary Degree of Evidence ( $\lambda$ ). The Degrees of Evidence are real values between 0 and 1 that denote, in the case of the Favorable Degree of Evidence, the evidence that proposition  $P$  is true, and in the case of the Contrary Evidence Degree denotes contrary when proposition  $P$  is not true.

It is called Degree of Uncertainty  $G_{in}(\mu, \lambda)$  of an annotation  $(\mu, \lambda)$  to any of the degrees of inconsistency or paracompleteness.

It is called Gce Degree of Certainty  $G_{ce}(\mu, \lambda)$  of an annotation  $(\mu, \lambda)$  to any of the degrees of truth or falsehood. Depending on the Values of the Degrees of Evidence, four extreme states can be: true, false, inconsistent, and paracomplete as shown in Figure 2.

The Para-analyzer algorithm is composed of a set of information collected through a research questionnaire for decision-making analysis [5] The definition of the Paraconsistent Decision Method (MPD), proposed in the studies, reflects the method, through Paraconsistent Logic, to use contradictory information, obtaining results that help in decision making. [1] [4] Figure represents the Unitary Square of the Cartesian Plan (QUPC), the degrees of certainty and contradiction, grouped into twelve states, which graphically presents the Para-analyzer algorithm

Extreme states	Symbol	Non-extreme states	Symbol
True	V	Quasi-true tending to Inconsistent	$QV \rightarrow T$
False	F	Quasi-true tending to Paracomplete	$QV \rightarrow \perp$
Inconsistent	T	Quasi-false tending to Inconsistent	$QF \rightarrow T$
Paracomplete	$\perp$	Quasi-false tending to Paracomplete	$QF \rightarrow \perp$
		Quasi-inconsistent tending to True	$QT \rightarrow V$
		Quasi-inconsistent tending to False	$QT \rightarrow F$
		Quasi-paracomplete tending to True	$Q\perp \rightarrow V$
		Quasi-paracomplete tending to False	$Q\perp \rightarrow F$

Figure 1 - Unitary Square of the Cartesian Plane (Source: [1])

It is observed, then, that with the Para-analyzer algorithm, in addition to the four extreme states, it is possible to determine eight more non-extreme states. The Figure 1 shows the states that represent the non-extreme states in the para-parser algorithm. The Para-analyzer algorithm works with four external control values defined in the application environment, represented in Figure 3:  $V_{cve}$  - Veracity control value,  $0 \leq V_{cve} \leq 1$ .  $V_{cfa}$  - Falsehood control value,  $-1 \leq V_{cfa} \leq 0$ .  $V_{cik}$  - Inconsistency Control Value,  $0 \leq V_{cik} \leq 1$  e  $V_{cpa}$  - Paracompleteza control value,  $-1 \leq V_{cpa} \leq 0$

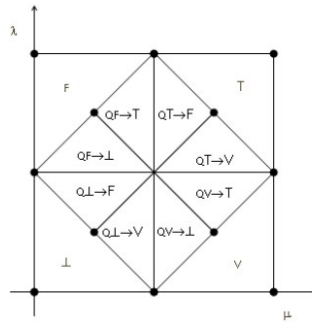


Figure 2 - Lattice  $\tau$  (Source: [2])

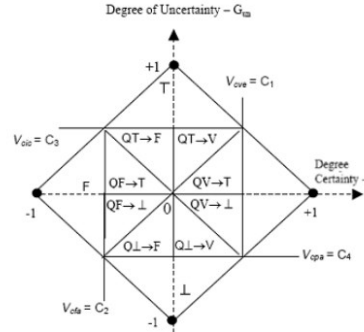


Figure 3 - Control Values (Source:[3])

### 3 Methodology

This research intends to evaluate the usability through the data collected through the AHP questionnaire. The users' responses will be submitted to the algorithm para-analyzer, the results of which will be published in an upcoming article after application and analysis of the results. The objective of this research is exploratory because it is intended to build knowledge about the application of Logic Et in the evaluation of usability, through the process of presenting them to a group of specialists who work with the software registration of providers, is based on the research as a survey.

The target audience of interest in this research is the professionals in the area of registration of providers who work with the software in their daily lives, hereinafter called specialists. Participants are not identified with personal data in this research. The questionnaire will be forwarded to the selected specialists via e-mail and telephone contact for the acceptance of participation.

Nine specialists working in a health care company in the city of São Paulo were selected, divided into three groups, adopting the position as a grouping criterion, according to Table 1.

Like Logic Et, it works with values between 0 and 1. The experts' answers will be mapped to logic Et's values. The value 0 is assigned to the answer "I totally disagree" and the value 1 to the answer "I totally agree", where 0.5 means "Indifferent" and so on, according to Table 2.

**Table 1** - Classification of Expert Groups

Group	Occupation	Number of Interviewees
A	Coordinators	3
B	Senior Registry Analysts	3
C	Junior Registry Analysts	3

Source: Authors

**Table 2** - Conversion of values

Answer	Score	Value Logic Et
I totally agree	5	1
Agree	4	0,75
Indifferent	3	0,5
Disagree	2	0,25
I totally disagree	1	0

Source: Authors

The research instrument was composed of ten closed questions based on the SUS questionnaire, with a Likert scale [10][13] according to Table 3. The questions for usability evaluation were mapped to the factors of Logic Et, according to Table 3.

With the result of the questionnaire, we obtain the agreement or degree of favorable evidence ( $\mu$ ) and the disagreement or evidence contradicts ( $\lambda$ ) for each opinion of a specialist regarding the usability of the software. With the data collected from the groups of specialists, through the questionnaire, it is intended to feed a database to be

6

submitted to the Algorithm Para-analyzer of Logic Er, according to the model in Table 4.

As an example of an application of Logic Er's MPD method, the following simulation is produced, considering that the 9 experts, who are called Expert 1, Expert 2, ..., Expert 9 in the database, answered two questions: Factor 1 (F1) - I think I would like to use this application frequently and F2 - I found this application unnecessarily complicated. Data entries are obtained for the Algorithm Para-analyzer, according to the database shown in Table 4.

By applying the Para-analyzer algorithm, developed in an Excel spreadsheet, the result is obtained according to Figures 4 and 5. The result of the Global Analysis of the Para-Analyzer algorithm (0.87; 0.25) indicates that it is in the "Totally True" region resulting in a viable result. The Para-analyzer algorithm processed a consensus in the opinion of the nine experts, presenting the following results for the factors: F1 (0.75; 0.25) and F2 (1.00; 0.25). It is understood that the usability of the software is within a viable quality standard.

**Table 3 - SUS Questionnaire**

<b>Factor - Question</b> F1 - I think that I would like to use this system frequently. ( ) I totally disagree ( ) Disagree ( ) Indifferent ( ) Agree ( ) I totally agree	<b>Factor - Question</b> F6 - I thought there was too much inconsistency in this system. ( ) I totally disagree ( ) Disagree ( ) Indifferent ( ) Agree ( ) I totally agree
F2 - I found the system unnecessarily complex. ( ) I totally disagree ( ) Disagree ( ) Indifferent ( ) Agree ( ) I totally agree	F7 - I felt very confident using the system. ( ) I totally disagree ( ) Disagree ( ) Indifferent ( ) Agree ( ) I totally agree
F3 - I thought the system was easy to use. ( ) I totally disagree ( ) Disagree ( ) Indifferent ( ) Agree ( ) I totally agree	F8 - I found the system very complicated to use. ( ) I totally disagree ( ) Disagree ( ) Indifferent ( ) Agree ( ) I totally agree
F4 - I think that I would need the support of a technical person to be able to use this system. ( ) I totally disagree ( ) Disagree ( ) Indifferent ( ) Agree ( ) I totally agree	F9 - I would imagine that most people would learn to use this system very quickly. ( ) I totally disagree ( ) Disagree ( ) Indifferent ( ) Agree ( ) I totally agree
F5 - I found the various functions in this system were well integrated. ( ) I totally disagree ( ) Disagree ( ) Indifferent ( ) Agree ( ) I totally agree	F10 - I needed to learn a lot of things before I could get going with this system. ( ) I totally disagree ( ) Disagree ( ) Indifferent ( ) Agree ( ) I totally agree

Source: Adapted from [8]

**Table 4 - Database Model formed  $\mu$  and  $\lambda$  assigned by the experts for each factor of SUS questionnaire**

Factor	Group A						Group B						Group C					
	Expert 1		Expert 2		Expert 3		Expert 4		Expert 5		Expert 6		Expert 7		Expert 8		Expert 9	
	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$
F1	0,75	0,25	1,00	0,25	0,75	0,25	0,50	0,25	0,75	0,25	1,00	0,25	0,75	0,25	0,50	0,25	0,75	0,75
F2	1,00	0,25	0,75	0,25	1,00	0,25	0,75	0,25	1,00	0,00	1,00	0,25	0,75	0,00	0,75	0,25	1,00	1,00

Source: Authors

Factor	Resulting degrees		Decision
	$\mu$	$\lambda$	
F1	0,75	0,25	Not conclusive
F2	1,00	0,25	Viable
Global analysis	0,87	0,25	Viable

Figure 4 - Final result of the questionnaire analysis (Source: Authors)

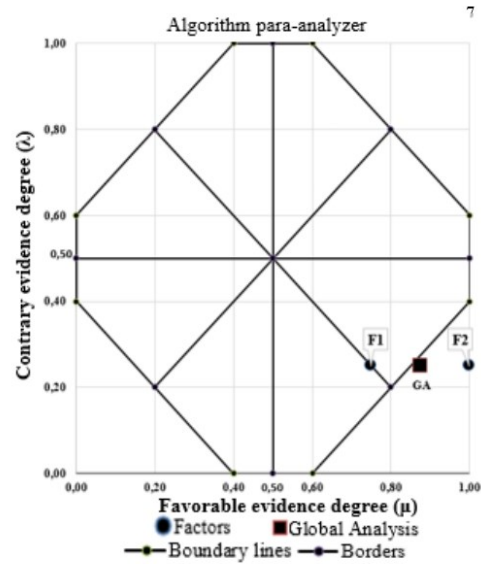


Figure 5 - Overall analysis of questionnaires (Source: Authors)

#### 4 Conclusions

Current usability assessment processes do not take into account inconsistency or contradiction. In a real situation, both contradiction and inconsistency appear due to the conditions of the environment in which the questionnaire was applied.

Therefore, the existence of conflicts is part of the usability evaluation process. The greater the involvement of specialists with the software to solve their work processes daily, the higher the levels of conflicts, contradictions, and inconsistencies to which they will be subject.

The research presents a paraconsistent model to assist in the decision-making process of usability evaluation using Logic Et, through the Para-analyzer algorithm, using logical criteria that allow a technical validation, and the input parameters are established by the opinions of the experts consolidating the input values into a collective logic of all, converted into mathematical terms.

The research presents a paraconsistent model based on the perspectives that provide satisfaction to the specialists, assisting in the decision-making process of usability evaluation using logic Et. That can be used to validate and present a new technique of dealing with expert opinions in situations of blurring or even inconsistency in the use of the software. Possible future situations are relevant for decision making that ensure the quality of usability assessment and the success of software. Both the analysis and the result of this evaluation will be presented in a future article.



## 5 References

1. Abe, J. M.; Da Silva Filho, J. I.; Celestino, U.; Corrêa, de A. H. *Lógica Paraconsistente Anotada Evidencial*. Et. Santos, Editora Comunicar, 2011.
2. Abe, J. M. (2010). Paraconsistent logics and applications. 4th International Workshop on Soft Computing Applications, (pp. 11-18).
3. Abe, J. M., Akama, S., & Nakamatsu, K. (2015). *Introduction to Annotated Logics: Foundations for Paracomplete and Paraconsistent Reasoning*. Cham: Springer International Publishing.
4. Abe, J. M., *Paraconsistent Intelligent-Based Systems - New Trends in the Applications of Paraconsistency*. In: Springer (Ed.), Book Series: Intelligent Systems Reference Library (Vol. 94, p. 306). Switzerland: Springer.
5. Akama, S. (Ed.). (2016). *Towards Paraconsistent Engineering* (Vol. 110). Springer International Publishing.
6. ALAMER, Reem A et al. L3MS: A Lightweight Language Learning Management System using Mobile Web Technologies. p. 1–2, 2015.
7. Bolturk, E., Kahraman, C. A novel interval-valued neutrosophic AHP with cosine similarity measure. *Soft Comput* 22, 4941–4958 (2018). <https://doi.org/10.1007/s00500-018-3140-y>
8. Brooke, J. (1996). SUS: A ‘quick and dirty’ usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 189–194). London, UK: Taylor and Francis.
9. BUYYA, Rajkumar; BROBERG, James; GOSCINSKI, Andrzej (org.). *Cloud Computing: principles and paradigms*. Hoboken, New Jersey: John Wiley & Sons, 2011.
10. Derham, P. A. J. (2011). Using preferred, understood, or effective scales? How to scale presentations effect online survey data collection. *Australasian Journal of Marketing & Social Research*, 19(2), 13-26
11. Jeff Saturo, James R. Lewis. *Quantifying the User Experience (Second Edition)*, 2016. ISBN 978-0-12-802308-2. 2016 Elsevier
12. ISO/IEC 25010:2011. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. <https://www.iso.org/standard/35733.html>. Acesso em 25/05/2020.
13. Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology*, 22(140), 1-55.
14. Mahdi Saeedpoor, Amin Vafadarnikjoo Corrigendum to “Multicriteria renewable energy planning using an integrated fuzzy VIKOR & AHP methodology: The case of Istanbul” [Energy 35(6) (2010) 2517–2527] *Energy*, Volume 79, 1 January 2015, Pages 536-537
15. Manchar, Anuradha & Chouhan, Ankit. (2017). Salesforce CRM: A new way of managing customer relationships in a cloud environment. 1-4. 10.1109/ICECCT.2017.8117887.
16. Vaidya, Omkarprasad & Kumar, Sushil. (2006). Analytic Hierarchy Process: An Overview of Applications. *European Journal of Operational Research*. 169. 1-29. 10.1016/j.ejor.2004.04.028.

## Apêndice 2

Este artigo foi escrito seguindo as normas do congresso XL Encontro Nacional de Engenharia de Produção (ENEGEP 2020), Foz do Iguaçu, Paraná, Brasil, no qual foi aprovado.

- FORÇAN, L. R.; ABE, J. M.; OLIVEIRA, C. C.; NASCIMENTO, S. S., Aprimoramento do Processo de Avaliação de Histórias de Usuários utilizando a Lógica Paraconsistente Anotada Evidencial Et. Contribuições da Engenharia de Produção para a Gestão de Operações.



XL ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO  
 “Contribuições da Engenharia de Produção para a Gestão de Operações Energéticas Sustentáveis”  
 Foz do Iguaçu, Paraná, Brasil, 20 a 23 de outubro de 2020.

XL ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO  
 “Contribuições da Engenharia de Produção para a Gestão de Operações Energéticas Sustentáveis”  
 Foz do Iguaçu, Paraná, Brasil, 20 a 23 de outubro de 2020.

## Aprimoramento do Processo de Avaliação de Histórias de Usuários utilizando a Lógica Paraconsistente Anotada Evidencial Et

**Luiz Roberto Forçan – Universidade Paulista**  
[luforcan@gmail.com](mailto:luforcan@gmail.com)

**Jair Minoro Abe – Universidade Paulista**  
[jairabe@uol.com.br](mailto:jairabe@uol.com.br)

**Cristina Corrêa de Oliveira – Instituto Federal de São Paulo**  
[crisolive@ifsp.edu.br](mailto:crisolive@ifsp.edu.br)

**Samira Sestari do Nascimento – Universidade Paulista**  
[samirasestari@gmail.com](mailto:samirasestari@gmail.com)



*Desenvolvedores de software precisam ser ágeis para atenderem as necessidades dos usuários, entregando software em prazos exíguos e com qualidade. A história do usuário é uma técnica utilizada nos métodos ágeis para elicitar requisitos. Este processo é realizado com o desenvolvedores e o usuário, podendo haver contradições entre eles, resultando em métricas inexatas. Este artigo apresenta um modelo de validação da história do usuário utilizando o algoritmo Para-analisador, baseado na Lógica Paraconsistente Anotada Evidencial Et para auxiliar o aprimoramento do processo de avaliação, de priorização e estimativas das histórias de usuários. Foi realizado um survey com um time de desenvolvedores que trabalham com métodos ágeis. O modelo utiliza como variáveis de entrada, os graus de evidência favorável e contrária para cada critério INVEST. A aplicação deste modelo permite considerar questões extremamente relevantes quando se trata de apoiar a tomada de decisão com base em um modelo matemático, além de servir como instrumento de apoio para os times, Product Owner, Gerente de Projetos e outros. Foram analisadas quatro histórias de usuários por nove especialistas, que avaliaram os critérios para cada história de usuário. A interpretação das avaliações realizadas pelos especialistas deu-se por intermédio da análise global no quadrado unitário do plano cartesiano, que indicou os graus de evidência favorável e evidência contrária dos dados utilizados. Foram constatadas duas histórias que não poderiam ser desenvolvidas em um Sprint e, portanto, deveriam ser refatoradas e submetidas novamente a opinião de especialistas. As outras duas histórias possuíam grau de evidências favoráveis adequadas para serem alocadas em uma Sprint.*

**Palavras-chave:** INVEST, Scrum, Métodos ágeis, Lógica Paraconsistente Anotada Evidencial Et.



## 1. Introdução

Um requisito é definido como uma função, serviço ou recurso atender uma necessidade ou demanda do usuário do produto, conforme a definição da Engenharia de Requisitos (ER), subárea da engenharia de Software. Requisitos podem ser divididos em funções, restrições ou regras de negócio. Independentemente da metodologia adotada, é necessário conhecer, entender e identificar a complexidade dos requisitos para planejamento e execução (SOMMERVILLE, 2010).

As empresas veem adotando o uso de metodologias ágeis para desenvolvimento de software, buscando diminuir o tempo de desenvolvimento, aumentando a qualidade do produto em um mercado competitivo. O Scrum se destaca entre as metodologias mais utilizadas para alcançar estes objetivos (SUTHERLAND, 2016).

Em um ambiente ágil, os requisitos dos usuários são tratados como histórias dos usuários (HU), que são solicitações por escrito em uma ou mais frases, mostrando uma visão externa do ponto de vista do usuário.

A partir de uma lista de requisitos do usuário, denominada de *Product Backlog* do produto, são priorizadas tarefas que o time de desenvolvimento (TIME) se compromete a desenvolver em uma Sprint, definida como pequeno ciclo de desenvolvimento de software que dura de uma a quatro semanas.

Em razão do envolvimento constante do cliente, com a validação e teste de aceitação das HUs, torna-se uma necessidade frequente (HEIKKILÄ *et al.*, 2015), pois o conhecimento do negócio, do domínio da aplicação ou o conhecimento geral de onde o sistema será utilizado, embora não seja uma entrada tangível no processo de validação, é muito importante, pois está muito ligado à cultura e o processo organizacional da empresa.

Na etapa de validação, os desenvolvedores avaliam se as HUs são claras o suficiente para iniciar uma Sprint. Caso se verifique que existe incerteza, falta de informação ou duplicidade na história de usuário estas dúvidas são sanadas junto ao *Product Owner* (PO), que é o representante do cliente, ou o próprio cliente.

Estudos anteriores aplicaram técnicas diferentes na priorização de requisitos como AHP, Árvore de Pesquisa Binária, Método de Votação Cumulativa (SHEEMAR; KOUR, 2017) Triagem de Requisitos de Priorização Baseada em Lógica Fuzzy Orientada por Especialistas (AASEM; RAMZAN; JAFFAR, 2010) (BABAR; RAMZAN; GHAYYUR, 2011). Outros, como Beck (BECK, 1999), utilizou um jogo para priorizar os requisitos em programação extrema entre os membros do TIME.



Um fator de sucesso para o projeto, na etapa final do planejamento da Sprint, é a validação da qualidade das HUs, pelo TIME e o PO, garantindo que os requisitos estejam corretos, documentados e validados (MATEEN; ABBAS; AKBAR, 2017). No entanto, ser capaz de determinar a complexidade das HUs - e o tempo necessário para cada uma delas - continua sendo uma tarefa laboriosa, visto que elas são muitas vezes escritas de maneira simples, com requisitos mal definidos e estão entre os itens considerados de riscos em projetos ágeis de software, afetando o planejamento do desenvolvimento de software (HANNAY; BENESTAD; STRAND, 2019) (POPLI; CHAUAHN, 2015). Outro aspecto desta validação é o consenso coletivo, dado que a análise de cada HU é individual, podendo haver contradições de análise entre os participantes do planejamento.

O processo de validação deve gerar uma lista de problemas e ações acordadas entre o cliente, o PO e o TIME. Os principais problemas encontrados neste processo são os que não atendem ao critério do INVEST, um acrônimo para histórias Independente, Negociável, Valiosa, Estimável, Small ou pequenas e Testável (BUGLIONE; ABRAN, 2013).

A partir desta lista de problemas deve ser realizado um plano de ação acordado entre todos. Os vários caminhos que podem surgir desta reunião de entendimento, para chegar a um resultado, podem ter o apoio da Lógica Paraconsistente Anotada Evidencial Et (Lógica Et), que trata dados incertos e contraditórios, advindos dos envolvidos, para entendimento de uma determinada HU, podendo ser expressos como contradições lógicas ou meramente como diferenças (CARVALHO; ABE, 2018), trazendo qualidade a tomada de decisão (ABE; AKAMA; NAKAMATSU, 2015).

Este artigo apresenta os resultados de um *survey*, aplicando um questionário estruturado, para os especialistas de desenvolvimento de software, em diferentes funções nas empresas, avaliarem quatro HUs, mensurando as evidências favoráveis e contrárias para cada critério do INVEST.

A pesquisa apresenta um modelo para auxiliar no processo decisório de avaliação das HUs utilizando a Lógica Et, por meio do algoritmo Para-analisador, utilizando critérios lógicos que possibilitam a validação técnica, sendo que os parâmetros de entrada são estabelecidos pelas opiniões dos especialistas, consolidando uma lógica coletiva do time, convertida em termos matemáticos.

## 2. Fundamentação

### 2.1 História do Usuário

As histórias do usuário<sup>1</sup>, ou “user stories”, é uma maneira simples de descrever as necessidades do cliente, ou seja, é apenas um requisito bem expresso, que captura os elementos essenciais: para quem é, qual funcionalidade é esperada do software e por que ela é importante.

<sup>1</sup> Adotou-se o termo História do Usuário para a tradução de user stories, a fim de manter compatibilidade com a tradução





O formato mais difundido para escrever HUs é o "Como um <papel>, eu quero <meta/desejo> de modo que <benefício>"; utilizada para expressar requisitos no desenvolvimento ágil, utilizando uma linguagem do negócio. Porém elas são mais do que apenas um formato curto para capturar requisitos: para desenvolvedores de software, elas são um meio de estruturar requisitos dos POs e *stakeholders*, formando a base para o desenvolvimento, enfatizando a importância e o valor das HUs (DALPIAZ; BRINKKEMPER, 2018).

Os principais problemas na ER são: requisitos incompletos ou ocultos; falhas de comunicação entre o TIME e PO e requisitos instáveis ou imaturos. Isto se dá pelo fato das HUs serem escritas no idioma dos POs e não no tecnicismo da computação, o que pode dificultar a estimativas, pois não há clareza das saídas que o sistema deve produzir e as entradas não são medidas objetivamente (BIK; LUCASSEN; BRINKKEMPER, 2017)

## 2.2 INVEST

Invest é um acrônimo, utilizado pela comunidade ágil, como um conjunto de critérios, ou um check-list, para avaliar a qualidade de uma HU pelo TIME. Caso a HU não atenda estes critérios, o TIME deverá reescrever a HU, junto com o PO, a fim de ambos os lados possam concordar o suficiente para trabalhar de maneira eficaz no desenvolvimento dela.

Esta é uma abordagem de estimativa de esforço, de dimensionamento do produto e prazo. Estes critérios são baseados nos atributos: (I)ndependente, (N)egociável, (V)alor, (E)stimável, (S) pequena e (T)estável, para avaliar e estimar as HUs. A estimativa é importante do ponto de vista comercial, a fim de minimizar erros ou falhas. Estes são, geralmente, consequência de requisitos incompletos, ou fornecidos tardiamente (aumento do escopo) ou não ter o nível certo de detalhes para expressar a menor unidade de atividade significativa para o usuário. Outro objetivo é reduzir a subjetividade nas estimativas de pontos da história, que normalmente são derivadas e gerenciadas de maneira experimental sem referência aos dados históricos da organização para comparação, falta de experiência em estimar ou mesmo falta de conhecimento do domínio (KAMTHAN; SHAHMIR, 2016).

A definição para (I)ndependente é que os requisitos, capturados em uma HU, possam ser implementados no software em qualquer ordem. Quando um requisito, para ser implementado, depende de outros, ele não é independente, e precisa ser revisto pela equipe. (N)egociável é a definição para um contrato aberto entre o PO e o TIME durante o desenvolvimento, cujos os detalhes serão acrescentados posteriormente, pois uma boa HU captura apenas a essência e não os detalhes. (V)aliosa define a importância da HU para o PO. A HU não representa valor para o negócio caso ela não seja

---

do livro (SUTHERLAND, 2016)



valiosa. A HU deve priorizar o negócio, sendo valiosa para o PO. O TIME deve entregar a essência de uma solução completa.

Estimável, definição do (E), consiste em estimar uma HU para que o PO possa classificá-la e agendá-la nas Sprints. Este critério está fortemente relacionado com o entendimento e tamanho da HU. É função da equipe a estimativa, porém ela depende da experiência da equipe: o que é fácil estimar variará dependendo da experiência da equipe no desenvolvimento e no domínio. A comunidade ágil define que HUs pequenas (S de small) e que caibam em uma Sprint são ideais para serem trabalhadas, pois se obtêm estimativas mais precisas.

Por fim, o critério (T)estável implica que toda HU deve ser testável, pois, caso o PO não saiba como testar, pode ser um indicativo de falta de clareza de requisito ou mesmo que a mesma não seja valiosa o suficiente para o negócio.

### 2.3 Lógica Paraconsistente Anotada Evidencial $E\tau$

A Lógica Paraconsistente figura entre as chamadas lógicas não clássicas, por conter disposições contrárias a alguns dos princípios básicos da Lógica Aristotélica, tais como o princípio da contradição, dado que nesta lógica qualquer afirmação é necessariamente verdadeira ou falsa. Segundo a Lógica Paraconsistente, uma sentença e a sua negação podem ser ambas verdadeiras (ABE *et al.*, 2011).

A necessidade de tomada de decisão, em momentos em que há mais de uma opção a seguir, são baseadas em aspectos subjetivos, que não são organizados nem sistemáticos (ABE, 2010).

No cotidiano, a coleta de informações de diversas fontes, pode ser contraditória, abrindo espaço para as incertezas, resultando que contradizem e impedem a tomada de decisão.

Na Lógica Paraconsistente Anotada Evidencial  $E\tau$ , para cada proposição P associa-se um Grau de Evidência Favorável ( $\mu$ ) e um Grau de Evidência Contrária ( $\lambda$ ). Os Graus de Evidências são valores reais entre 0 e 1 que denotam, no caso do Grau de Evidência Favorável, a evidência de que a proposição P seja verdadeira, e no caso do Grau de Evidência Contrária denota contrária quando a proposição P não seja verdadeira.

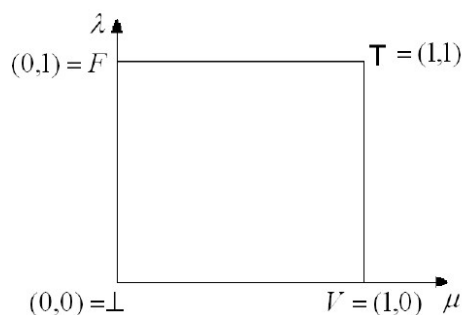
Chama-se Grau de Incerteza  $G_m(\mu, \lambda)$  de uma anotação  $(\mu, \lambda)$  a qualquer um dos graus de inconsistência ou de paracompleteza.

Chama-se Grau de Certeza  $G_{ce}(\mu, \lambda)$  de uma anotação  $(\mu, \lambda)$  a qualquer um dos graus de verdade ou de falsidade.

Dependendo dos valores dos Graus de Evidência, podem-se ter quatro estados extremos: verdadeiro, falso, inconsistente e paracompleto, conforme Figura 1.

- O estado verdadeiro (V) acontece quando  $\mu = 1,0$  e  $\lambda = 0,0$ .
- O estado falso (F) acontece quando  $\mu = 0,0$  e  $\lambda = 1,0$ .
- O estado inconsistente (T) acontece quando  $\mu = 1,0$  e  $\lambda = 1,0$ .
- O estado paracompleto ( $\perp$ ) acontece quando  $\mu = 0,0$  e  $\lambda = 0,0$ .

Figura 1 - Estados Lógica Et

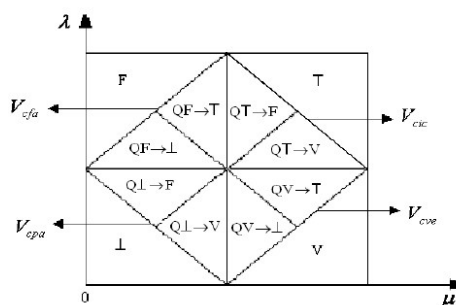


Fonte: (ABE, 2010)

O algoritmo Para-analisador é composto por um conjunto de informações coletadas por meio de formulário de pesquisa para análise da tomada de decisão (AKAMA, 2016). A definição do Método Paraconsistente de Decisão (MPD), proposta nos estudos (CARVALHO; ABE, 2018) (ABE, 2015), reflete o método, por meio da Lógica Paraconsistente, de forma a utilizar informações contraditórias, obtendo-se resultados que auxiliam na tomada de decisão. A Figura 2 representa o Quadrado Unitário do Plano Cartesiano (QUPC), os graus de certeza e de contradição, agrupados em doze estados, que apresenta graficamente o algoritmo Para-analisador.



Figura 2 - Quadrado Unitário do Plano Cartesiano



Fonte: (ABE *et al.*, 2011)

Observa-se, então, que com o algoritmo Para-analisador, além dos quatro estados extremos, é possível determinar mais oito estados não extremos. No Quadro 1 estão as regiões que representam os estados não extremos neste algoritmo.

Quadro 1 - Representação de Estados Não Extremos da Lógica Paraconsistente

Símbolo	Descrição
$QV \rightarrow T$	Quase verdadeiro tendendo para o inconsistente
$QV \rightarrow \perp$	Quase verdadeiro tendendo para Paracompleteza
$QF \rightarrow T$	Quase falso tendendo para o inconsistente
$QF \rightarrow \perp$	Quase falso tendendo para Paracompleteza
$QT \rightarrow V$	Quase inconsistente tendendo a verdadeiro
$QT \rightarrow F$	Quase inconsistente tendendo a falso
$Q\perp \rightarrow V$	Quase Paracompleteza tendendo para verdadeiro
$Q\perp \rightarrow F$	Quase Paracompleteza tendendo para falso

Fonte: (ABE; AKAMA; NAKAMATSU, 2015)

O algoritmo Para-analisador trabalha com quatro valores de controle externos definidos no ambiente de aplicação, representados na Figura 2.

- $V_{cve}$  – Valor de controle de veracidade,  $0 \leq V_{cve} \leq 1$
- $V_{cfa}$  – Valor de controle de falsidade,  $-1 \leq V_{cfa} \leq 0$
- $V_{cic}$  – Valor de controle de inconsistência,  $0 \leq V_{cic} \leq 1$
- $V_{cpa}$  – Valor de controle de paracompleteza,  $-1 \leq V_{cpa} \leq 0$



### 3. Metodologia

Esta seção apresenta os procedimentos metodológicos aplicados nesta pesquisa com o propósito de atingir os resultados finais. Foram estabelecidos no enquadramento metodológico: o objetivo da pesquisa; o tipo de pesquisa; o processo da pesquisa com a definição dos especialistas; a definição do instrumento de coleta e, por fim, a análise dos dados.

O objetivo, da presente pesquisa, é de natureza exploratória, pois se pretende construir um conhecimento sobre a aplicação da Lógica Et na avaliação das HUs, mediante o processo de apresentação delas para um grupo de especialistas da área de desenvolvimento de software.

A natureza deste artigo é fundamentada na pesquisa com *survey*, pois se busca informações com um grupo de interesse. O público alvo de interesse nesta pesquisa são os profissionais da área de desenvolvimento de software, doravante denominados de especialistas, que trabalham especificamente com a metodologia Scrum. Este tipo de pesquisa busca obter dados primários sobre as HUs por meio da aplicação de um questionário. Os participantes não são identificados com dados pessoais nesta pesquisa, mantendo-se o sigilo deles.

Foram selecionados nove especialistas que expressaram suas opiniões sobre cada um dos critérios INVEST, apontando  $\mu$  e  $\lambda$  para cada HU. Os cargos dos participantes são: Analistas de Sistemas Sênior e Pleno, Arquitetos de Software e Scrum Master. Estes profissionais, que trabalham em diferentes empresas de software na cidade de São Paulo, foram divididos em três grupos, adotando o cargo como critério de agrupamento, conforme Quadro 2. Eles são denominados E1, E2, ..., E9 na base de dados.

Quadro 2 - Classificação dos Grupos de Especialistas

Grupo	Cargo	Quantidade Entrevistados
A	Arquiteto de Software e Scrum Master	3
B	Analista de Sistema Sênior	3
C	Analista de Sistema Pleno	3

Fonte: Autores

Os critérios do INVEST foram mapeados para os fatores da Lógica Et, conforme Quadro 3.



XL ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO  
 “Contribuições da Engenharia de Produção para a Gestão de Operações Energéticas Sustentáveis”  
 Foz do Iguaçu, Paraná, Brasil, 20 a 23 de outubro de 2020.

Quadro 3 - Fatores e os critérios do INVEST

<b>Fator</b>	<b>Nome do Fator</b>
<b>F1</b>	Independente
<b>F2</b>	Negociável
<b>F3</b>	Valor
<b>F4</b>	Estimável
<b>F5</b>	Pequena
<b>F6</b>	Testável

Fonte: Autores

Os valores de  $\mu$  e  $\lambda$  foram normalizados, conforme Tabela 1, para facilitar a quantificação pelos especialistas.

Tabela 1- Normalização dos valores de  $\mu$  e  $\lambda$

<b>Grau</b>	<b>Porcentagem (%)</b>	<b>Descrição</b>
<b>1,00</b>	<b>100</b>	Não existe nenhuma dúvida em relação a evidência
<b>0,75</b>	<b>75</b>	Pequena dúvida em relação a evidência
<b>0,50</b>	<b>50</b>	Dúvida média em relação a evidência
<b>0,25</b>	<b>25</b>	Baixa certeza em relação a evidência
<b>0,00</b>	<b>0</b>	Quase nenhuma certeza em relação a evidência

Fonte: Autores

Os participantes receberam a mensagem por e-mail e contato telefônico para o aceite de participação; recebimento das HUs e o instrumento de avaliação.

Foram selecionadas quatro HUs para avaliação, selecionadas de levantamento de requisitos realizado em empresas. Elas foram retiradas de projetos anteriores e os respondentes desconheciam as mesmas, conforme Quadro 4.



XL ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO  
 “Contribuições da Engenharia de Produção para a Gestão de Operações Energéticas Sustentáveis”  
 Foz do Iguaçu, Paraná, Brasil, 20 a 23 de outubro de 2020.

Quadro 4 - Histórias do Usuário

HU	História de usuário	Crítérios de Aceitação
A	Como um Cliente, eu quero saber o preço e a disponibilidade de um produto para verificar se quero concluir a compra.	- Produto deve possuir o nome e o preço. - Produto deve estar associado a uma categoria e um fornecedor.
B	Como um Administrador eu quero consultar as categorias de produtos para visualizar as categorias de produto.	- Não listar as categorias que estejam descontinuadas.
C	Como um Administrador eu quero consultar produtos por categoria, eu conheço o código da categoria e quero encontrá-lo no sistema, para descobrir seus produtos para visualizar os produtos de uma categoria informada	- Não listar os produtos que estejam descontinuados. - Deve-se listar todas os produtos mesmo que estejam com o estoque zerado.
D	Como um cliente eu quero que o sistema disponibilize diversas formas de pagamento para poder pagar o meu pedido e o sistema fechar o pedido.	- Cliente poder selecionar uma forma de pagamento para poder efetuar pagamento. - Quando o pagamento for realizado com cartão de crédito, validar com a operadora a operação ou restrição. - Estando tudo ok prosseguir com a finalização do pedido.

Fonte: Autores

Dada as respostas dos especialistas, que atribuíram os valores de  $\mu$  e  $\lambda$  para cada HU, conforme Quadro 4, foi elaborada uma base de dados com os fatores analisados. A Figura 3 representa um pequeno excerto da base de dados da história de usuário A.

O Para-analisador, desenvolvido em uma planilha Excel, foi alimentado com os dados coletados dos grupos de especialistas para cada HU.

Como exemplo o especialista 1 do grupo A emitiu a seguinte opinião sobre o fator F1:

- $\mu$  = Evidência favorável = 0,50
- $\lambda$  = Evidência Contrária = 0,75
- Formando o par  $(\mu ; \lambda)$  ou (0,50 ; 0,75)



XL ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO  
 “Contribuições da Engenharia de Produção para a Gestão de Operações Energéticas Sustentáveis”  
 Foz do Iguaçu, Paraná, Brasil, 20 a 23 de outubro de 2020.

Figura 3 - Base de dados formada  $\mu$  e  $\lambda$  atribuídos pelos especialistas para cada fator de análise das HUs

Fator	Grupo A						Grupo B						Grupo C					
	E1		E2		E3		E4		E5		E6		E7		E8		E9	
	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$
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

Fonte: Autores

#### 4. Resultados

Após a aplicação do algoritmo Para-analisador, cada faixa recebe um diagnóstico, podendo ser:

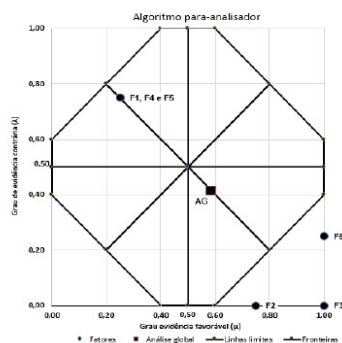
- Viável – Que é entendido como aprovado;
- Inviável – O item está em desacordo com o fator de avaliação;
- Não Conclusivo – O item necessita novas avaliações.

Para cada história de usuário obteve-se os seguintes resultados:

##### História de Usuário A

Conforme a Figura 4, o resultado da Análise Global do algoritmo Para-analisador (0,58 ; 0,42) indica que este se encontra na região “Quase Verdadeiro Tendendo ao inconsistente” resultando não conclusivo. A avaliação qualitativa paraconsistente demonstra que já não é uma verdade absoluta de que a HU apresenta uma ótima qualidade. O algoritmo Para-analisador processou, matematicamente, um consenso na opinião dos 9 especialistas, apresentando os seguintes resultados para os fatores: F1 - Independente (0,25 ; 0,75), F4 - Estimável (0,25 ; 0,75), F5 - Pequena (0,25 ; 0,75) resultaram não conclusivo. Os fatores F2 - Negociável (0,75; 0), F3 - Valiosa (1,00 ; 0) e F6 - Testável (1,00 ; 0,25) resultaram viáveis. Entende-se que esta HU não é independente, não pode ser estimada, está muito grande e não pode ser testada. Supõe-se que a qualidade dela está insuficiente, necessitando que novas informações sejam obtidas e submetida novamente para análise dos especialistas.

Figura 4 - Análise da História de Usuário A

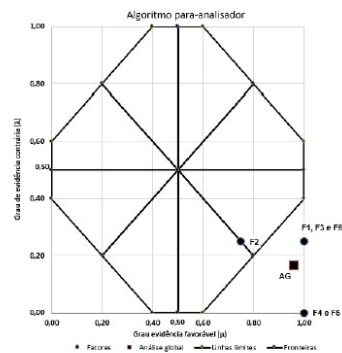


Fonte: Autores

### História de Usuário B

Conforme a Figura 5 o resultado da Análise Global do algoritmo Para-analisador (0,96 ; 0,17) indica que este se encontra na região “Totalmente Verdadeiro” resultando viável. A avaliação qualitativa paraconsistente demonstra que a HU apresenta uma qualidade adequada. O algoritmo Para-analisador processou um consenso na opinião dos 9 especialistas, apresentando os seguinte resultados para os fatores: F1 - Independente (1,00 ; 0,25), F3 - Valiosa (1,00 ; 0,25), F4 - Estimável (1,00 ; 0,00), F5 - Pequena (1,00 ; 0,00) e F6 - Testável (1,00 ; 0,25) apresentaram resultados viáveis. O fator F2 - Negociável (0,75 ; 0,25) apresenta o resultado não conclusivo. Entende-se que esta HU está dentro de um padrão de qualidade viável e pode fazer parte de uma Sprint.

Figura 5 - Análise da História de Usuário B



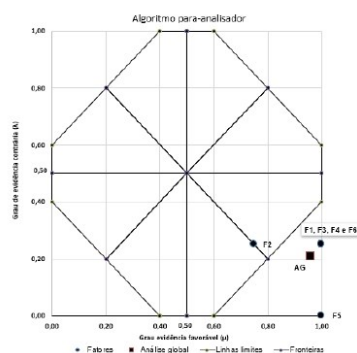
Fonte: Autores

### História de Usuário C



Conforme a Figura 6, o resultado da Análise Global do algoritmo Para-analisador (0,96 ; 0,21) indica que este se encontra na região “Totalmente Verdadeiro” resultando viável. A avaliação qualitativa paraconsistente demonstra que a HU apresenta uma qualidade viável. O algoritmo Para-analisador processou um consenso na opinião dos 9 especialistas, apresentando os seguinte resultado para os fatores: F1 - Independente (1,00 ; 0,25), F3 - Valiosa (1,00 ; 0,25), F4 - Estimável (1,00 ; 0,25), F5 - Pequena (1,00 ; 0,00) e F6 - Testável (1,00 ; 0,25) resultaram viáveis. O fator F2 - Negociável (0,75 ; 0,25) resultou não conclusivo. Entende-se que esta HU está dentro de um padrão de qualidade viável e pode fazer parte de uma Sprint.

Figura 6 - Análise da História de Usuário C

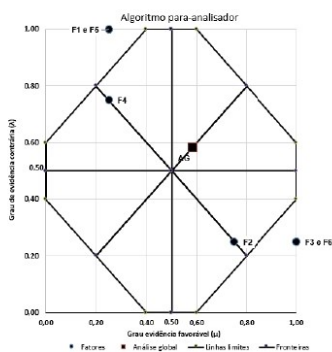


Fonte: Autores

### História de Usuário D

Conforme a Figura 7, o resultado da Análise Global do algoritmo Para-analisador (0,59 ; 0,59) indica que este encontra-se na região “Inconsistente tendendo ao falso” apresentando resultado não conclusivo. A avaliação qualitativa paraconsistente demonstra que já não é uma verdade absoluta a conclusão de que esta HU apresenta uma qualidade viável. O algoritmo Para-analisador processou um consenso na opinião dos 9 especialistas, apresentando os seguintes resultados para os fatores: F1 - Independente (0,25 ; 1,00) e F5 - Pequena (0,25 ; 1,00) resultaram inviáveis. Os fatores F3 - Valiosa (1,00 ; 0,25) e F6 - Testável (1,00 ; 0,25) resultaram viáveis. Os fatores F2 - Negociável (0,75 ; 0,25) e F4 - Estimável (0,25 ; 0,75) resultaram não conclusivos. Entende-se que esta HU depende de outras e está muito grande. Supõe-se que a qualidade desta HU está insuficiente, necessitando que novas informações sejam obtidas e submetida novamente para análise dos especialistas.

Figura 7 - Análise da História de Usuário D



Fonte: Autores

O resultado das HUs estabelece a avaliação e processamento dos critérios INVEST, o grau de resultados individuais  $\mu$  e  $\lambda$  definindo as decisões analíticas conforme Figura 8.

Figura 8 - Resultado Final das HUs

Fator	HU A			HU B			HU C			HU D		
	Graus Result		Decisão	Graus Result		Decisão	Graus Result		Decisão	Graus Result		Decisão
	$\mu$	$\lambda$		$\mu$	$\lambda$		$\mu$	$\lambda$		$\mu$	$\lambda$	
F1-Independente	0,25	0,75	NÃO CONCLUSIVO	1,00	0,25	VIÁVEL	1,00	0,25	VIÁVEL	0,25	1,00	INVIÁVEL
F2-Negociável	0,75	0,00	VIÁVEL	0,75	0,25	NÃO CONCLUSIVO	0,75	0,25	NÃO CONCLUSIVO	0,75	0,25	NÃO CONCLUSIVO
F3-Valor	1,00	0,00	VIÁVEL	1,00	0,25	VIÁVEL	1,00	0,25	VIÁVEL	1,00	0,25	VIÁVEL
F4-Estimável	0,25	0,75	NÃO CONCLUSIVO	1,00	0,00	VIÁVEL	1,00	0,25	VIÁVEL	0,25	0,75	NÃO CONCLUSIVO
F5-Pequena	0,25	0,75	NÃO CONCLUSIVO	1,00	0,00	VIÁVEL	1,00	0,00	VIÁVEL	0,25	1,00	INVIÁVEL
F6-Testável	1,00	0,25	VIÁVEL	1,00	0,25	VIÁVEL	1,00	0,25	VIÁVEL	1,00	0,25	VIÁVEL
Análise global	0,583	0,417	NÃO CONCLUSIVO	0,958	0,167	VIÁVEL	0,958	0,208	VIÁVEL	0,583	0,583	NÃO CONCLUSIVO

Fonte: Autores

## 5. Conclusão

Os processos atuais de validação de HUs não levam em consideração a inconsistência ou a contradição. Em uma situação real, tanto a contradição como a inconsistência, aparecem em razão das condições do ambiente em que foi realizado o levantamento de requisitos. Estas situações de contradição, aparecem independente da vontade dos membros da equipe de desenvolvimento de software ou por parte da equipe da área de negócio da empresa. Portanto, a existência de conflitos faz parte do processo de validação das HUs. Quanto maior for o envolvimento dos *stakeholders*, clientes, *Product Owner* e o time de desenvolvimento para resolução dos interesses, necessidade e desejos do cliente, maior serão os níveis de conflitos, contradições e inconsistências a que estarão sujeitos.

A pesquisa apresenta um modelo para auxiliar no processo decisório de avaliação das HUs utilizando a





Lógica Et, por meio do algoritmo Para-analisador, utilizando critérios lógicos que possibilitam uma validação técnica, sendo que os parâmetros de entrada são estabelecidos pelas opiniões dos especialistas consolidando uma lógica coletiva do TIME convertida em termos matemáticos.

A análise baseada nestas perspectivas proporciona satisfação aos especialistas e *stakeholders* pois o modelo paraconsistente maximiza todas as opiniões dos especialistas e cria um consenso matemático sobre estas opiniões.

Esta pesquisa demonstra que os conceitos da Lógica Paraconsistente Anotada Evidencial Et podem ser utilizadas para validar e apresentar perspectivas de lidar com situações de indefinição e inconsistência. Estas situações ou fatores são relevantes para tomadas de decisão, e influenciam na qualidade e sucesso de um sistema de informações.

## REFERÊNCIAS

- AASEM, M.; RAMZAN, M.; JAFFAR, A. **Analysis and optimization of software requirements prioritization techniques**. Karachi: IEEE. 2010. p. 1–6.
- ABE, J. M. **Paraconsistent logics and applications**. 4th International Workshop on Soft Computing Applications. jul. 2010. p. 11-18.
- ABE, J. M. Paraconsistent Intelligent-Based Systems - New Trends in the Applications of Paraconsistency. In: SPRINGER **Book Series: Intelligent Systems Reference Library**. Switzerland: Springer, v. 94, 2015. p. 306. ISBN: 978-3-319-19721-0.
- ABE, J. M. et al. **Lógica Paraconsistente Anotada Evidencial Et**. Santos: Comunicar, 2011.
- ABE, J. M.; AKAMA, S.; NAKAMATSU, K. Introduction to Annotated Logics: Foundations for Paracomplete and Paraconsistent Reasoning. Cham: Springer International Publishing, 2015. p. 1–4. ISBN: 978-3-319-17912-4. Disponível em: <[https://doi.org/10.1007/978-3-319-17912-4\\_1](https://doi.org/10.1007/978-3-319-17912-4_1)>.
- AKAMA, S. (Ed.). **Towards Paraconsistent Engineering**. [S.l.]: Springer International Publishing, v. 110, 2016. ISBN: 978-3-319-40417-2. Disponível em: <<https://doi.org/10.1007/978-3-319-40418-9>>.
- BABAR, M. I.; RAMZAN, M.; GHAYYUR, S. A. K. **Challenges and future trends in software requirements prioritization**. International Conference on Computer Networks and Information Technology. [S.l.]: [s.n.]. jul. 2011. p. 319-324.
- BECK, K. **Extreme Programming Explained: Embrace Change**. USA: Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN: 0201616416.
- BIK, N.; LUCASSEN, G.; BRINKKEMPER, S. **A Reference Method for User Story Requirements in Agile Systems Development**. 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). [S.l.]: [s.n.]. set.



XL ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO  
 “Contribuições da Engenharia de Produção para a Gestão de Operações Energéticas Sustentáveis”  
 Foz do Iguaçu, Paraná, Brasil, 20 a 23 de outubro de 2020.

2017. p. 292-298.

BUGLIONE, L.; ABRAN, A. **Improving the User Story Agile Technique Using the INVEST Criteria**. 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement. [S.l.]: [s.n.]. out. 2013. p. 49-53.

CARVALHO, F. R.; ABE, J. M. A Paraconsistent Decision-Making Method. In: \_\_\_\_\_ **Smart Innovation, Systems and Technologies**. [S.l.]: Springer International Publishing, v. 87, 2018.

DALPIAZ, F.; BRINKKEMPER, S. **Agile Requirements Engineering with User Stories**. 2018 IEEE 26th International Requirements Engineering Conference (RE). [S.l.]: [s.n.]. ago. 2018. p. 506-507.

HANNAY, J. E.; BENESTAD, H. C.; STRAND, K. Agile Uncertainty Assessment for Benefit Points and Story Points. **IEEE Software**, v. 36, p. 50-62, jul. 2019. ISSN: 1937-4194.

HEIKKILÄ, V. T. et al. **A Mapping Study on Requirements Engineering in Agile Software Development**. 2015 41st Euromicro Conference on Software Engineering and Advanced Applications. [S.l.]: [s.n.]. 2015. p. 199-207.

KAMTHAN, P.; SHAHMIR, N. **Modeling Negative User Stories is Risky Business**. 2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE). [S.l.]: [s.n.]. jan. 2016. p. 236-237.

MATEEN, A.; ABBAS, K.; AKBAR, M. A. **Robust approaches, techniques and tools for requirement engineering in agile development**. 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI). [S.l.]: [s.n.]. 2017. p. 100-103.

POPLI, R.; CHAUAHN, N. **Managing uncertainty of story-points in Agile software**. 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom). [S.l.]: [s.n.]. mar. 2015. p. 1357-1361.

SHEEMAR, H.; KOUR, G. **Enhancing User-Stories Prioritization Process in Agile Environment**. 2017 International Conference on Innovations in Control, Communication and Information Systems (ICICCI). [S.l.]: [s.n.]. ago. 2017. p. 1-6.

SOMMERVILLE, I. **Software Engineering**. 9th. ed. USA: Addison-Wesley Publishing Company, 2010. ISBN : 0137035152.

SUTHERLAND, J. **A arte de fazer o dobro na metade do tempo**. Tradução de Natalie Gerhardt. 2. ed. São Paulo: LeYa, 2016. ISBN 978-8544104514.

### **Apêndice 3**

Este artigo foi escrito seguindo as normas da revista Research, Society and Development Journal - ISSN 2525-3409, no qual foi aprovado.

- FORÇAN, L. R.; ABE, J. M.; NASCIMENTO, S. S.; OLIVEIRA C. C., Escala de Usabilidade do Sistema Baseada na Lógica Paraconsistente Anotada Evidencial  $E_{\tau}$ .

## Escala de Usabilidade do Sistema Baseada na Lógica Paraconsistente Anotada Evidencial Er

System Usability Scale Based on Paraconsistent Annotated Evidential Logic Er

Escala de Usabilidad del Sistema Basada en la Logica Paraconsistente Anotada Evidencial Er

**Luiz Roberto Forçan**

ORCID: <https://orcid.org/0000-0001-7376-830X>  
Universidade Paulista, Brasil  
E-mail: [luforcan@gmail.com](mailto:luforcan@gmail.com)

**Jair Minoro Abe**

ORCID: <https://orcid.org/0000-0003-2088-9065>  
Universidade Paulista, Brasil  
E-mail: [jairabe@uol.com.br](mailto:jairabe@uol.com.br)

**Samira Sestari do Nascimento**

ORCID: <https://orcid.org/0000-0001-7429-6033>  
Universidade Paulista, Brasil  
E-mail: [sestarisamira@gmail.com](mailto:sestarisamira@gmail.com)

**Cristina Corrêa de Oliveira**

ORCID: <https://orcid.org/0000-0002-8629-6679>  
Instituto Federal de São Paulo, Brasil  
E-mail: [crisolive@ifsp.edu.br](mailto:crisolive@ifsp.edu.br)

### Resumo

A opinião da experiência do usuário utilizando um software afeta diretamente a qualidade dele. A Escala de Usabilidade do Sistema mensura a experiência do usuário final com o sistema, que apresenta questões que se contradizem. O objetivo deste artigo é demonstrar um aprofundamento das contradições, aplicando a Lógica Paraconsistente Anotada Evidencial Er, com a realização de uma pesquisa com onze profissionais que avaliaram um sistema. Foi realizada uma comparação dos resultados da escala SUS com o algoritmo Para-analisador, utilizando o par  $(\mu; \lambda)$ , atribuindo as questões pares para  $\mu$  e as questões ímpares para  $\lambda$ . O primeiro resultado apresentou uma avaliação positiva da usabilidade, com a Escala de Usabilidade do Sistema (76,59). O segundo resultado, realizado com o Para-analisador, apresentou a média global (0,75; 0,25), que aponta uma similaridade com o primeiro, destacando dois itens contraditórios. A avaliação aprimorou a análise da usabilidade, pois houve a concordância dos respondentes em nove características da usabilidade. Porém apresentou uma divergência em um item, visto que cinco respondentes avaliaram duas questões contraditórias com valores altos, apresentando uma contradição na avaliação. Este resultado pode evidenciar uma falha de percepção na semântica do instrumento, que poderia indicar uma necessidade de treinamento dos participantes antes do início do processo. A principal contribuição desta pesquisa foi avaliar a usabilidade de um software em conjunto com o Para-analisador, possibilitando uma análise mais detalhada, pois ele trabalha com o par  $(\mu; \lambda)$ , sendo possível avaliar os resultados em duas dimensões.

**Palavras-chave:** Escala de Usabilidade do Sistema; Lógica Paraconsistente Anotada Evidencial Er; Contradição; Análise bidimensional.

### Abstract

The evaluation of the user experience using a software directly affects its quality. The System Usability Scale measures the end user's experience with the system, which presents contradictory issues. The objective of this article is to demonstrate a deepening of those contradictions, using the Evidential Annotated Paraconsistent Logic Er, with the application of a survey for eleven professionals who evaluated a system. A comparison of the results of the above scale was performed with the Para-analyzer algorithm, using the pair  $(\mu, \lambda)$ , assigning the even questions to  $\mu$  and the odd questions to  $\lambda$ . The first result showed a positive assessment of usability, with the System Usability Scale (76.59). The second result, performed with the Para-analyzer, presented the global average (0.75, 0.25), which indicates a similarity with the first, highlighting two contradictory items. The evaluation improved the usability analysis, as respondents agreed on nine usability characteristics. However, there was a divergence in one item, as five respondents rated two questions with high values, presenting a contradiction in the assessment. This result may show a perception failure in the instrument's semantics, which could indicate a need for training of participants before the beginning of the process. The main contribution of this research was to evaluate the usability of a software in conjunction with the Para-analyzer, enabling a more detailed analysis, as it works with the pair  $(\mu, \lambda)$ , being possible to evaluate the results in two dimensions.

**Keywords:** System Usability Scale; Paraconsistent Logic Annotated Er; Contradiction; Two-dimension analysis.

### Resumen

La evaluación de la experiencia del usuario al utilizar un software afecta directamente a su calidad. La escala de usabilidad del sistema mide la experiencia del usuario final con el sistema, que presenta problemas contradictorios. El objetivo de este artículo es demostrar una profundización de esas contradicciones, utilizando la Lógica Paraconsistente Anotada Evidencial Et, con la aplicación de una encuesta a once profesionales que evaluaron un sistema. Se realizó una comparación de los resultados de la escala anterior con el algoritmo Para-analizador, utilizando el par  $(\mu, \lambda)$ , asignando las preguntas pares a  $\mu$  y las preguntas impares a  $\lambda$ . El primer resultado mostró una valoración positiva de la usabilidad, con la Escala de usabilidad del sistema (76,59). El segundo resultado, realizado con el Paraanalizador, presentó el promedio global (0,75, 0,25), lo que indica similitud con el primero, destacando dos ítems contradictorios. La evaluación mejoró el análisis de usabilidad, ya que los encuestados coincidieron en nueve características de usabilidad. Sin embargo, hubo una divergencia en un ítem, ya que cinco encuestados calificaron dos preguntas con valores altos, lo que presenta una contradicción en la evaluación. Este resultado puede mostrar una falla de percepción en la semántica del instrumento, lo que podría indicar la necesidad de capacitar a los participantes antes del inicio del proceso. El principal aporte de esta investigación fue evaluar la usabilidad de un software en conjunto con el Para-analizador, permitiendo un análisis más detallado, ya que trabaja con el par  $(\mu, \lambda)$ , siendo posible evaluar los resultados en dos dimensiones.

**Palabras clave:** Escala de Usabilidad del Sistema; Lógica Paraconsistente Anotada Evidencial Et; Contradicción; Análisis Bidimensional.

## 1. Introdução

A competitividade da indústria de software obriga os desenvolvedores a aprimorarem o processo de produção de software a fim de obter vantagens competitivas, pois a qualidade de um software está diretamente ligada ao modelo de processo de software. Avaliação da usabilidade mede o grau de satisfação dos usuários que utilizam um sistema no seu dia a dia, denominado Sistema de Gestão de Prestadores de Serviço (SGPS).

A norma ISO/IEC 25010 (ISO/IEC 25010, 2021), define características e subcaracterísticas, internos e externos da qualidade de um software e seu uso. A Usabilidade se divide em seis subcaracterísticas: Inteligibilidade, Apreensibilidade, Operacionalidade, Proteção contra Erros do usuário, Estética da Interface e Acessibilidade (Esaki et al., 2013).

Existem diversos instrumentos de avaliação da usabilidade (Jeff Sauro, 2016); a Escala de Usabilidade do Sistema (SUS) (Brooke, 1995) avalia a experiência do usuário com poucos itens, sendo aplicados em testes de usabilidade, capacidade de aprendizado, entre outros aspectos. As principais vantagens do SUS são a objetividade na coleta de dados primários resultante da interação dos usuários com o sistema; na replicabilidade do instrumento; e na facilidade de quantificação e análise dos resultados.

A Lógica Paraconsistente Anotada Evidencial Et (Lógica Et), permite a análise das opiniões subjetivas dos usuários, considerando suas incertezas e contradições, que são inerentes aos valores humanos (Abe, 2015). Esta lógica faz parte do grupo das lógicas não clássicas.

A pesquisa apresenta uma proposta de reestruturação interna do SUS, utilizando o algoritmo Para-analisador, sendo que as variáveis de entrada foram estabelecidas pelo julgamento dos usuários do sistema. A Lógica Et utiliza critérios lógicos que permite evidenciar a contradição, auxiliando no processo decisório.

O procedimento deste estudo foi um *survey* com onze profissionais da área administrativa, de uma empresa de assistência médica privada, utilizando o SUS, a fim de mensurar a usabilidade do Sistema de Gestão de Prestadores de Serviços, apresentando os resultados com o SUS, comparando-os com uma adaptação interna, tendo como base a Lógica Et, atribuindo as respostas ao par  $(\mu, \lambda)$ , a fim de ressaltar as contradições.

## 2. Referencial Teórico

Os seguintes tópicos estão fundamentados na teoria sobre Avaliação da Qualidade de Software, Sistema de Gestão de Prestadores de Serviço, Escala do Usabilidade de Sistema, assim como a aplicação da Lógica Paraconsistente Anotada Evidencial Et.



## 2.1. Qualidade de Software

Qualidade é um termo relativo, para uma pessoa um produto pode ter qualidade, para outra, a qualidade pode estar ausente (Denning, 2016). No desenvolvimento de software, a qualidade e a confiabilidade são dois tópicos importantes (ISO/IEC 25010, 2021), pois as métricas mensuram as características internas e externas de um software após uma avaliação subjetiva realizado por humanos reagindo a artefatos.

Devido ao fato do desenvolvimento de software ser centrado no ser humano, ignorar as práticas de qualidade e o processo de avaliação dos aspectos psicológicos e comportamentais na usabilidade podem acarretar em consequências sociais, organizacionais, econômicas e provocar o fracasso do produto de software (Ghanbari et al., 2018).

A qualidade do software precisa ser garantida pelo padrão de qualidade do software, avaliação da qualidade do software e mecanismo de gerenciamento do processo de testes de software (Shen et al., 2018).

A norma ISO/IEC 25040 apresenta uma descrição do processo de aferição da qualidade de um software, estabelecendo os requisitos para esta atividade (ISO/IEC 25040, 2021). As atividades de avaliação da qualidade de um software devem ser projetadas antecipadamente, considerando-se a disponibilidade de recursos, ferramentas de software e instrumentos de avaliação. O plano de avaliação também deve incluir o propósito da avaliação, os padrões, os métodos e ferramentas (Brooke, 1995).

## 2.2 Sistema de Gestão de Prestadores de Serviço

O Sistema de Gestão de Prestadores de Serviços (SGPS) foi desenvolvido utilizando a metodologia de gerenciamento ágil Scrum (Kniberg, 2015) com o objetivo de automatizar o *workflow* de contratação de prestadores de serviço, a fim de realizar o gerenciamento eletrônico de dados reduzindo os custos do processo e melhorando a eficiência operacional seguindo a abordagem de resolução de problemas (Laudon, 2014).

Três profissionais daquela área desempenharam o papel de *Product Owners*, participando do desenvolvimento, junto com a área de Tecnologia da Informação. Para Manchar & Chouhan (2017), para o desenvolvimento do SGPS foi utilizado a ferramenta Salesforce®, que tem como característica principal a facilidade da gestão de relacionamento com clientes *Customer Relationship Management* (Buyya et al., 2011).

## 2.3 Escala de Usabilidade do Sistema

A Escala de usabilidade do sistema (SUS) é utilizada na análise da usabilidade, pois tem se mostrado um instrumento flexível, não sendo afetado por mudanças de palavras e linguagem (Wahyuningrum et al., 2020).

O SUS é composto por 10 questões padronizadas (Brooke, 1995), onde, para cada uma das questões, o usuário elege uma resposta que está atribuída a um modo de distribuição escalar, com valores variando de um a cinco, nos extremos entre: "Concordo Totalmente" até "Discordo Totalmente", conforme pode ser visto na Tabela 1.

Confiabilidade e usabilidade são dois tópicos importantes em um software conforme a norma ISO/IEC 25010 (2021). A usabilidade é a ideia central da interação homem-computador, pois a experiência do usuário combina aspectos como eficiência e eficácia com critérios adicionais como qualidade, facilidade de uso e atratividade.

Equação 1 – Fórmula do SUS

$$SUS = \left[ \sum_{n=1}^5 (U_{2n-1} - 1 + (5 - U_{2n})) \right] \times 2.5 \quad (1)$$

Fonte: Brooke (1995)

O resultado final do SUS é a soma de todas as contribuições da pontuação para os dez itens multiplicados por 2.5,

conforme mostrado em Equação 1, onde  $U_i$  se refere à classificação do  $i$ -ésimo item, variando entre 0 e 100, onde valores mais altos refletem maior satisfação do usuário.

Resultados com pelo menos 90 pontos representa que a expectativa, na usabilidade, está acima do esperado pelo usuário; valores entre 80 e 90 pontos representam usabilidade excelente. Os resultados que atingem entre 70 e 80 pontos são os que apresentam boa usabilidade, mas que apresentam algumas melhorias a serem realizadas. Os que atingem resultado entre 60 e 70 pontos são considerados “ok” mas devem ser fortemente melhorados, e por fim, os abaixo de 60 pontos não apresentam grau de usabilidade aceitável (Brooke, 1995).

Tabela 1 – Escala de usabilidade do sistema

Questões	Critério
Q1 - Eu acho que gostaria de usar essa aplicação frequentemente.	( ) Discordo totalmente ( ) Discordo Indiferente ( ) Concordo ( ) Concordo totalmente
Q2 - Eu achei essa aplicação desnecessariamente complexa.	
Q3 - Eu achei a aplicação fácil para usar.	
Q4 - Eu acho que precisaria do apoio de um suporte técnico para usar essa aplicação.	
Q5 - Eu achei que as várias funções da aplicação estavam bem integradas.	
Q6 - Eu achei que havia muita inconsistência na aplicação.	
Q7 - Imagino que a maioria das pessoas possa aprender a utilizar este aplicativo muito rapidamente.	
Q8 - Achei a aplicação muito complicada de se usar.	
Q9 - Eu me senti muito confiante em utilizar esta aplicação.	
Q10 - Eu precisei aprender várias coisas antes que eu pudesse começar a usar essa aplicação.	

Fonte: Brooke (1995)

## 2.4 Lógica Paraconsistente Anotada Evidencial $E\tau$

A tomada de decisão envolve a congruência das opiniões dos membros participantes, na seleção de opções alternativas (Abe et al., 2011). Em geral, a tomada de decisão é baseada num processo cognitivo que envolvem aspectos subjetivos e intuitivos. A intuição é um ato inconsistente e involuntário e a subjetividade não tem uma medida perfeita. A coleta de informações advém de diversas fontes, podendo ser muitas vezes contraditórias, abrindo lacunas para as incertezas e contradições que impedem, muitas vezes, a tomada de decisão (Akama, 2016).

A Lógica Paraconsistente Anotada Evidencial  $E\tau$  (Lógica  $E\tau$ ) é denominada como uma técnica da inteligência artificial, pois pertence as chamadas lógicas não clássicas, em razão de possuir premissas contraditórias como o princípio da contradição (Abe, 2015), que diferentemente da lógica clássica, uma sentença e a sua negação podem ser ambas verdadeiras.

A Lógica  $E\tau$  permite tratar dados subjetivos do mundo real em dados precisos com saídas numéricas (Abe et al., 2015). Uma de suas vantagens é realizar a tradução da linguagem natural (termos linguísticos) utilizada na comunicação cotidiana em expressões matemáticas (Abe., 2015). Isso é obtido por meio de propriedades de anotações reticuladas, que cada preposição  $p$  associa, no senso comum, um par  $(\mu; \lambda)$ , representado pelas letras gregas  $\mu$  ( $\mu$ ) e  $\lambda$  ( $\lambda$ ), descrito da seguinte maneira:  $p(\mu; \lambda)$  denominado constante de anotação, que pertence ao produto Real Cartesiano  $[0, 1] \times [0, 1]$ . Intuitivamente,  $\mu$  representa o grau de evidência favorável expresso em  $p$ , e  $\lambda$  representa o grau de evidência desfavorável expresso por  $p$ . Os valores de  $\mu$  e  $\lambda$  estão limitados entre 0 e 1 (Abe, 2015).

Na Lógica  $E\tau$ , os estados da decisão possuem quatro valores limites extremos: verdadeiro, falso, inconsistente e paracompleto (Da Silva Filho et al., 2008).

- O estado verdadeiro (V) acontece quando  $\mu = 1,0$  e  $\lambda = 0,0$
- O estado falso (F) acontece quando  $\mu = 0,0$  e  $\lambda = 1,0$

- O estado inconsistente (T) acontece quando  $\mu = 1,0$  e  $\lambda = 1,0$
- O estado paracompleto ( $\perp$ ) acontece quando  $\mu = 0,0$  e  $\lambda = 0,0$

Além dos quatro estados extremos, possui também oito estados não extremos, conforme Tabela 2, que são denominados de acordo com a proximidade com os estados lógicos extremos conforme Figura 1, como possíveis resultados para um suporte fundamental no momento da tomada de decisão permitindo mais possibilidades de análise (Abe, 2010).

Figura 1. Representação do Reticulado  $\tau$

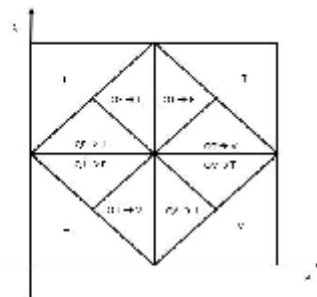


Tabela 2. Simbolização dos Estados Lógicos

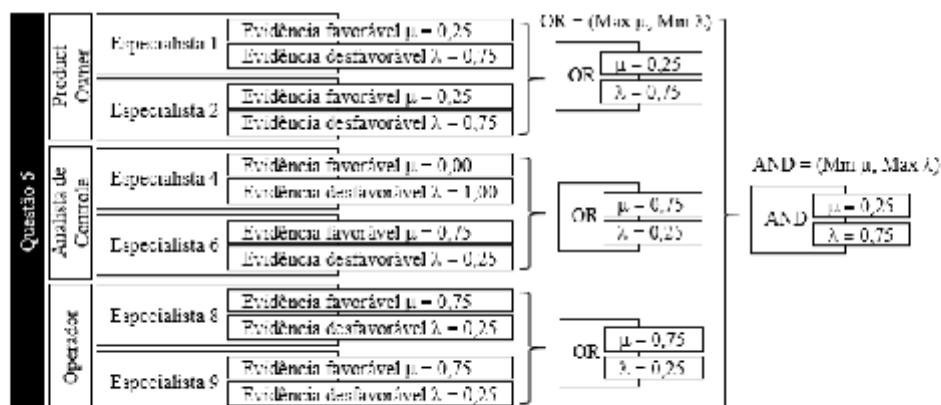
Estados extremos	Símbolo	Estados não extremos	Símbolo
Verdadeiro	V	Quase-Verdade tendendo a Inconsistente	$Q_V \rightarrow T$
Falso	F	Quase-Verdade tendendo a Paracompleto	$Q_V \rightarrow \perp$
Inconsistente	T	Quase-Falso tendendo a Inconsistente	$Q_F \rightarrow T$
Paracompleto	$\perp$	Quase-Falso tendendo a Paracompleto	$Q_F \rightarrow \perp$
		Quase-Inconsistente tendendo a Verdadeiro	$Q_T \rightarrow V$
		Quase-Inconsistente tendendo a Falso	$Q_T \rightarrow F$
		Quase-Paracompleto tendendo a Verdadeiro	$Q_{\perp} \rightarrow V$
		Quase-Paracompleto tendendo a Falso	$Q_{\perp} \rightarrow F$

Fonte: Abe (2015).

O algoritmo Para-analisador auxilia a tomada de decisão, pois ele admite a evidência favorável e desfavorável, advindas de um instrumento de avaliação que capta opiniões, auxiliando a tomada de decisão (Carvalho & Abe, 2018). A Figura 1 apresenta o Quadrado Unitário do Plano Cartesiano (QUPC), sendo que o par  $(\mu; \lambda)$  representa a evidência favorável e desfavorável. Os graus de certeza e de contradição são calculados pelos operadores OR e AND, conforme pode ser visto na Figura 2, dado que o OR representa a maximização das opiniões dos especialistas, e o AND representa a minimização das opiniões dos especialistas. O cômputo final, denominada Análise Global, é a média dos graus resultantes, que pode apresentar doze estados, sendo que são quatro extremos e oito não extremos (Abe et al., 2015).



Figura 2 - Lógica Et evidenciando a contradição



Fonte: Autores

### 3. Metodologia

A presente pesquisa é de natureza aplicada, com objetivo exploratório e de abordagem quali-quantitativa (Pereira et al., 2018), pois se pretende aplicar Lógica Et na captação de contradições dos respondentes na avaliação da usabilidade do SGPS. Esta seção apresenta os procedimentos aplicados neste estudo, com o propósito de ressaltar as contradições nos resultados. Foram estabelecidos, no enquadramento metodológico, o tipo de pesquisa; o objetivo da pesquisa; o procedimento da pesquisa com a definição dos usuários; a definição do instrumento e coleta da avaliação de usabilidade e, finalmente, a análise dos dados.

Esta pesquisa é de natureza aplicada, pois busca-se gerar conhecimento para a aplicação prática e dirigida à solução de problemas na usabilidade de software, pois está é um fator preponderante na aceitação de um software, pois sem ela o usuário não tem como desenvolver suas tarefas adequadamente (Harrison et al., 2013).

O objetivo deste estudo é exploratório, pois almeja-se ressaltar a contradição, com a Lógica Et, na avaliação de usabilidade aplicando o SUS; sendo que as perguntas são fechadas, utilizando uma escala *Likert* (Stevens, 1936), com valores normalizados variando de cinco a um, nos extremos com: “Concordo totalmente” a “Discordo totalmente”, correspondendo aos critérios numéricos atribuídos a escala, conforme Tabela 3.

Tabela 3 - Normalização de  $\mu$  e  $\lambda$ 

Resposta	Score SUS	Valor Lógica Et
Concordo totalmente	5	1,00
Concordo	4	0,75
Indiferente	3	0,50
Discordo	2	0,25
Discordo totalmente	1	0,00

Fonte: Autores

O procedimento adotado por este artigo, foi um *survey*, pois busca-se informações com um grupo de interesse, a fim de capturar a percepção dos usuários, aplicando um instrumento de avaliação para a população alvo de onze profissionais da área de Gerência de Prestadores de Serviço, que emitiram suas opiniões sobre a usabilidade do software utilizado em sua rotina diária. O anonimato dos participantes, que receberam o instrumento de avaliação por meio eletrônico, foi mantido. Eles foram organizados em três grupos, adotando o cargo como critério de classificação sendo o Product Owner (PO), os Analistas de Controle (AC) e os Operadores (OP), conforme Tabela 4. Eles foram denominados E<sub>1</sub>, E<sub>2</sub>, ..., E<sub>11</sub> na base de dados.

Tabela 4 - Classificação dos Grupos de usuários

Grupo	Ocupação	Número de entrevistados
A	Product Owner	3
B	Analista de Controle	4
C	Operador	4

Fonte: Autores

A coleta de dados primários foi realizada com o SUS. A Tabela 5 apresenta uma pequena amostra da base de dados da avaliação da usabilidade, sendo que as respostas foram atribuídas da seguinte forma: ímpares para evidência favorável ( $\mu$ ) complementando o valor de  $\lambda$  com  $1 - \mu$ , e as pares foram atribuídas para a evidência desfavorável ( $\lambda$ ), complementando o valor de  $\mu$  com  $1 - \lambda$ . A evidência favorável e a evidência desfavorável são valores que pertencem ao intervalo  $[0,1]$ , e resultam no grau de certeza (Abe, 2015).

Por exemplo, na questão Q1 do Especialista 1, foi respondido concordo, que após a normalização equivale ao valor 0,75 na Lógica Et, atribuído a  $\mu$  e com  $\lambda$  complementar 0,25, formando o par (0,75; 0,25). A partir da aplicação desta regra elaborou-se a Tabela 5:

Tabela 5 - Base de dados formada  $\mu$  e  $\lambda$  atribuídos pelos usuários

Question	Product owner						Analista de Controle						Operador									
	E1		E2		E3		E4		E5		E6		E7		E8		E9		E10		E11	
	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$
Q1	0.75	0.25	0.75	0.25	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.75	0.25	1.00	0.00	0.75	0.25	0.75	0.25

Fonte: Autores

#### 4. Resultados e Discussão

O resultado do SUS foi calculado aplicando a Equação 1, conforme pode ser conferido na coluna SUS Score da Tabela 6, sendo que a população alvo foi constituída por onze usuários que avaliaram o SGPS.

Tabela 6 - SUS Cálculo da pontuação total

Especialista	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Score
E1	4	2	4	2	4	4	2	2	4	4	60,0
E2	4	2	4	2	4	4	4	1	4	2	72,5
E3	5	2	4	1	4	4	4	1	4	2	77,5
E4	5	2	4	4	4	5	4	2	4	2	65,0
E5	5	1	5	1	5	5	5	1	2	1	82,5
E6	5	2	5	2	4	2	4	2	5	2	82,5
E7	5	1	5	1	4	2	5	1	4	2	90,0
E8	4	1	5	2	3	2	4	2	4	1	80,0
E9	5	1	5	3	4	2	5	1	5	1	90,0
E10	4	4	4	2	4	2	4	2	5	2	72,5
E11	4	2	4	2	4	2	4	2	4	4	70,0

Fonte: Autores

Obteve-se a média de 76,59 pontos para a usabilidade que equivale a um bom resultado geral; sendo que 27% responderam “Ok”, 36% responderam Bom e 36% responderam Excelente, conforme Tabela 7.

Tabela 7 - Resultado da Avaliação

Resultado SUS	Total de Avaliação
Ok	27%
Bom	36%
Excelente	36%

Fonte: Autores

Os valores coletados no SUS foram normalizados, aplicando o critério apresentado na Tabela 3, descrito na metodologia, resultando a Tabela 8.

Tabela 8 – Resultado normalização

	Product Owner						Analista de Controle						Operador									
Questão	E1		E2		E3		E4		E5		E6		E7		E8		E9		E10		E11	
	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$	$\mu$	$\lambda$
Q1	0,75	0,25	0,75	0,25	1,00	0,00	1,00	0,00	1,00	0,00	1,00	0,00	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	0,75	0,25
Q2	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,00	1,00	0,00	1,00	0,00	0,25	0,75	0,75	0,25
Q3	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,00	1,00	0,00	1,00	0,00	1,00	0,00	1,00	0,00	0,75	0,25	0,75	0,25
Q4	0,75	0,25	0,75	0,25	1,00	0,00	0,25	0,75	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	0,50	0,50	0,75	0,25	0,75	0,25
Q5	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,00	0,75	0,25	0,75	0,25	0,50	0,50	0,75	0,25	0,75	0,25	0,75	0,25
Q6	0,25	0,75	0,25	0,75	0,25	0,75	0,00	1,00	0,00	1,00	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25
Q7	0,25	0,75	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	0,75	0,25
Q8	0,75	0,25	1,00	0,00	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	1,00	0,00	0,75	0,25	0,75	0,25
Q9	0,75	0,25	0,75	0,25	0,75	0,25	0,75	0,25	0,25	0,75	1,00	0,00	0,75	0,25	0,75	0,25	1,00	0,00	1,00	0,00	0,75	0,25
Q10	0,25	0,75	0,75	0,25	0,75	0,25	0,75	0,25	1,00	0,00	0,75	0,25	0,75	0,25	1,00	0,00	1,00	0,00	0,75	0,25	0,25	0,75

Fonte: Autores

Entende-se que a usabilidade está dentro de um padrão de qualidade aceitável, pois observa-se que a satisfação dos usuários, Q1, apresenta o resultado totalmente verdadeiro, assim como, o resultado da complexidade, Q2, apresenta resultado na região verdadeira (0,75; 0,25), de acordo com a Tabela 9.

Tabela 9 - Resultado da análise do questionário

Fator	Grau resultante		Decisão
	$\mu$	$\lambda$	
Q1	1,00	0,00	Aceitável
Q2	0,75	0,25	Aceitável
Q3	0,75	0,25	Aceitável
Q4	0,75	0,25	Aceitável
Q5	0,75	0,25	Aceitável
Q6	0,25	0,75	Não Aceitável
Q7	0,75	0,25	Aceitável
Q8	1,00	0,00	Aceitável
Q9	0,75	0,25	Aceitável
Q10	0,75	0,25	Aceitável
<b>Análise Global</b>	<b>0,75</b>	<b>0,25</b>	<b>Aceitável</b>

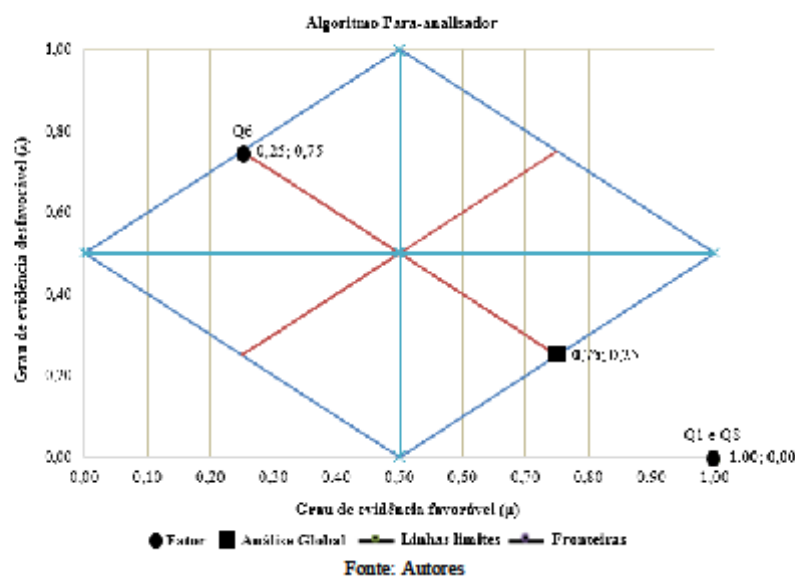
Fonte: Autores

A resposta sobre Facilidade de uso, Q3, apresenta resultado na região verdadeira (0,75; 0,25) significando que o usuário tem uma facilidade de uso, corroborando com a Q4, que significa que o usuário não apresenta dificuldade na utilização do SGPS, portanto não necessita de suporte técnico frequentemente.

As questões Q5, Q7 e Q9 apresentam resultados na região verdadeira (0,75; 0,25) nos itens de integração, facilidade de aprendizado e confiança de uso respectivamente.

As questões Q8 (Complexidade de uso) e Q10 (Treinamento intenso), apresentam valores acima da média, sinalizando que o usuário respondeu corretamente o instrumento, pois ele não contradisse as respostas anteriores. Logo o resultado encontra-se na região verdadeira (0,75; 0,25) do QUPC, indicando satisfação dos usuários nestes itens.

Figura 3 - Análise global do SUS



Os resultados de  $(\mu; \lambda)$  demonstram que os onze avaliadores apresentaram concordância em 9 fatores, sendo que dois estão na região totalmente verdadeira, ou seja, estado extremo do verdadeiro (1,00; 0,00). Sete fatores encontram-se na região limítrofe do verdadeiro, (0,75; 0,25), portanto as respostas para os itens Q1 a Q5 e Q7 a Q10 são representados por um único ponto, na Figura 2, devido a repetição de valores. Os resultados, do SUS e do Para-analisador, são convergentes, apresentando resultados similares. A Análise Global, resultante do Para-analisador, apresenta o par  $(\mu; \lambda)$  igual a (0,75; 0,25), que representa um consenso na opinião dos usuários, representado no gráfico por um quadrado preto preenchido, que se encontra na região limítrofe do estado verdadeiro do QUPC.

Foi detectada uma contradição na resposta da questão Q6, apresentando um resultado de (0,25; 0,75), localizado na região limítrofe Falsa no QUPC. Esta questão contradiz a resposta Q5, pois cinco respondentes avaliaram as duas questões com valores máximos, que pode indicar uma ausência de percepção na semântica das questões. Desta forma, pode ser constatado, na Figura 2, a contradição das respostas ao utilizar a Lógica Et e os seus operadores, que, diferentemente da análise aritmética do SUS, avalia uma única dimensão, não destacando a contradição apresentada.

A principal contribuição desta pesquisa foi a utilização de mais de uma dimensão para avaliar a usabilidade em conjunto com a Lógica Et, auxiliando em uma análise mais abrangente da usabilidade, em conjunto com valores extremos e não-extremos propostos (Abe et al., 2015).

## 5. Conclusão

Foi realizada uma adaptação do instrumento de avaliação da usabilidade de software, alicerçada pela Lógica Et, com o par  $(\mu, \lambda)$ , sendo que as respostas pares foram atribuídas para  $\lambda$ , e as respostas ímpares foram atribuídas para  $\mu$ . Estas atribuições foram realizadas pelo fato das questões do SUS contradizerem umas as outras.

O processo de avaliação do SGPS obteve uma média de avaliação dos usuários, considerando a usabilidade boa. Não foi obtida uma pontuação mais alta, provavelmente, por falta de discernimento semântico das questões, ao utilizar o instrumento de avaliação. Pressupõe-se que seja necessário um treinamento do SUS, antes da sua aplicação, com os usuários da área de Gerência de Prestadores de Serviço.

A avaliação crítica dos usuários, que utilizam o SGPS, propiciou o aprimoramento da análise da usabilidade, com a Lógica Et, que apresentou a concordância em nove características, e a divergência em uma característica. Portanto, é possível utilizar a Lógica Et em instrumentos de avaliação, a fim de explorar contradições na análise realizada com o público-alvo.

Recomenda-se que trabalhos futuros explorem outros instrumentos de avaliação, apoiados pela Lógica Et.

## Referências

- Abe, J. M. (7 de 2010). Paraconsistent logics and applications. *4th International Workshop on Soft Computing Applications*, (pp. 11–18). doi:10.1109/SOFA.2010.5565631
- Abe, J. M. (2015). *Paraconsistent Intelligent-Based Systems - New Trends in the Applications of Paraconsistency* (Vol. 94). (Springer, Ed.) Switzerland: Springer. doi:https://doi-org.ez338.periodicos.capes.gov.br/10.1007/978-3-319-19722-7
- Abe, J. M., Akama, S., & Nakamura, K. (2015). *Introduction to Annotated Logics: Foundations for Paraconsistent and Paraconsistent Reasoning*. Cham: Springer International Publishing. doi:10.1007/978-3-319-17912-4\_1
- Abe, J. M., Da Silva Filho, J. I., Celestino, U., & Corrêa, d. A. (2011). *Lógica Paraconsistente Anotada Evidencial Et. Comunicar*.
- Akama, S. (2016). *Towards Paraconsistent Engineering*. Springer International Publishing. doi:10.1007/978-3-319-40418-9
- Brooke, J. (11 de 1995). SUS: A quick and dirty usability scale. *Usability Eval. Ind.*, 189.
- Buyya, R., Broberg, J., & Goscinski, A. (2011). *Cloud Computing: principles and paradigms*. John Wiley & Sons.
- Carvalho, F. R., & Abe, J. M. (2015). *A Paraconsistent Decision-Making Method* (Vol. 87). Springer International Publishing. doi:https://doi.org/10.1007/978-3-319-74110-9
- Chen, C., Lin, S., Shoga, M., Wang, Q., & Boehm, B. (2018). How Do Defects Hurt Qualities? An Empirical Study on Characterizing a Software Maintainability Ontology in Open Source Software. (pp. 226–237). Lisbon: IEEE. doi:10.1109/QRS.2018.00036
- Da Silva Filho, J. I., Abe, J. M., & Torres, G. L. (2008). *Inteligência artificial com as redes de análises paraconsistentes*. LTC-Livros Técnicos e Científicos Editora S.A, São Paulo, 313.
- Denning, P. J. (8 de 2016). Software quality. *Communications of the ACM*, 59, 23–25. doi:10.1145/2971327
- Esaki, K., Azuma, M., & Komiyama, T. (2013). Introduction of Quality Requirement and Evaluation Based on ISO/IEC SQuaRE Series of Standard. *Em Trustworthy Computing and Services* (pp. 94–101). Springer Berlin Heidelberg. doi:10.1007/978-3-642-35795-4\_12
- Ghanbari, H., Vartiainen, T., & Siponen, M. (2 de 2018). Omission of Quality Software Development Practices: A Systematic Literature Review. *ACM Comput. Surv.*, 51. doi:10.1145/3177746
- Harrison, R., Flood, D., & Duce, D. (2013). Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Science*, 1, 1. doi:10.1186/2194-0827-1-1

ISO/IEC 25010. (15 de 1 de 2021). Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. Fonte: <https://www.iso.org/standard/35733.html>.

ISO/IEC 25040. (15 de 1 de 2021). Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process*. Fonte: <https://www.iso.org/standard/35765.html>.

Jeff Sauro, J. R. (2016). *Quantifying the User Experience*. Elsevier.

Kniberg, H. (11 de 2015). *Scrum and XP from the Trenches (Enterprise Software Development)*. lulu.com; Annotated edition.

London, J. P. (2014). *Management information systems: Managing the Digital Firm* (13th ed.). Pearson.

Manchar, A., & Chouhan, A. (2 de 2017). Salesforce CRM: A new way of managing customer relationship in cloud environment., (pp. 1-4). doi:10.1109/ICECCT.2017.8117887

Nakai, H., Tsuda, N., Honda, K., Washizaki, H., & Fukazawa, Y. (2016). Initial Framework for Software Quality Evaluation Based on ISO/IEC 25022 and ISO/IEC 25023. (pp. 410-411). Vienna: IEEE. doi:10.1109/QRS-C.2016.66

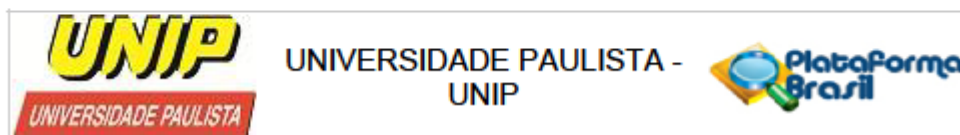
Pereira, A. S., Shituka, D. M., Pereira, F. J., & Shituka, R. (2018). Metodologia da pesquisa científica. Fonte: <http://repositorio.ufam.br/handle/1/15824>

Shan, P., Ding, X., Ran, W., & Yang, C. (2018). Research on Software Quality Assurance Based on Software Quality Standards and Technology Management. (pp. 385-390). Busan: IEEE. doi:10.1109/SNPD.2018.8441142

Wahyuningrum, T., Kartiko, C., & Wardhana, A. C. (10 de 2020). Exploring e-Commerce Usability by Heuristic Evaluation as a Complement of System Usability Scale. *2020 International Conference on Advancement in Data Science, E-learning and Information Systems (ICADEIS)*. IEEE. doi:10.1109/icadeis49811.2020.9277343



## Apêndice 4



### PARECER CONSUBSTANCIADO DO CEP

#### DADOS DO PROJETO DE PESQUISA

**Título da Pesquisa:** APLICAÇÃO DA LÓGICA PARACONSISTENTE EVIDENCIAL E EM SUBPROCESSOS DE ENGENHARIA DE SOFTWARE: ENGENHARIA DE REQUISITOS E INTERFACE HOMEM-COMPUTADOR

**Pesquisador:** LUIZ ROBERTO FORCAN

**Área Temática:**

**Versão:** 1

**CAAE:** 52164521.0.0000.5512

**Instituição Proponente:** ASSOCIACAO UNIFICADA PAULISTA DE ENSINO RENOVADO OBJETIVO-

**Patrocinador Principal:** Financiamento Próprio

#### DADOS DO PARECER

**Número do Parecer:** 5.021.838

**Apresentação do Projeto:**

adequada

**Objetivo da Pesquisa:**

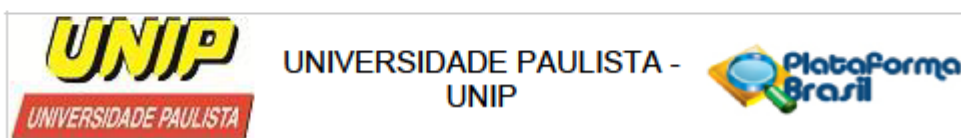
O objetivo desta pesquisa é realizar um estudo da avaliação de histórias de usuário e na avaliação da usabilidade de um software, aplicando a Lógica Paraconsistente Anotada Evidencial contribuindo na melhoria da tomada de decisão na etapa de análise do processo e da usabilidade de software.

**Avaliação dos Riscos e Benefícios:**

A pesquisa não fere princípios éticos. Os participantes de pesquisa serão convidados a participar da pesquisa no momento que antecede a aplicação do questionário, assim, sendo informados sobre os objetivos do questionário, juntamente com as informações de como o questionário e pesquisa serão desenvolvidos, sendo que o mesmo somente será realizado com o Termo de Consentimento Livre e Esclarecido devidamente lido e assinado, conforme Anexo 1 e Anexo 2, ficando uma via assinada para o pesquisador e outra para o participante de pesquisa. O anonimato dos participantes de pesquisa será resguardado, mantendo-se o sigilo, sem a coleta de dados pessoais sensíveis, a partir do posicionamento ético do pesquisador a fim de garantir a manutenção dos cuidados éticos da pesquisa.

O entrevistado não precisa se identificar com nenhum dado pessoal, bastando apenas emitir sua

**Endereço:** Rua Dr. Bacelar, 1212 4º andar  
**Bairro:** Vila Clementino **CEP:** 04.026-002  
**UF:** SP **Município:** SAO PAULO  
**Telefone:** (11)5586-4086 **E-mail:** cep@unip.br



Continuação do Parecer: 5.021.838

opinião sobre o assunto em questão.

**Comentários e Considerações sobre a Pesquisa:**

não há

**Considerações sobre os Termos de apresentação obrigatória:**

adequado

**Recomendações:**

não há

**Conclusões ou Pendências e Lista de Inadequações:**

não há

**Considerações Finais a critério do CEP:**

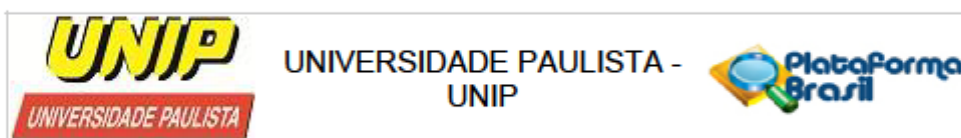
Ressalta-se que cabe ao pesquisador responsável encaminhar os relatórios parciais e finais da pesquisa, por meio da Plataforma Brasil, via notificação do tipo "relatório" para que sejam devidamente apreciadas pelo CEP, conforme Norma Operacional CNS nr 001/12, item XI.2.d.

**Este parecer foi elaborado baseado nos documentos abaixo relacionados:**

Tipo Documento	Arquivo	Postagem	Autor	Situação
Informações Básicas do Projeto	PB_INFORMAÇÕES_BÁSICAS_DO_PROJETO_1833461.pdf	28/09/2021 18:05:00		Aceito
Brochura Pesquisa	questionarioUsabilidade.docx	28/09/2021 18:02:37	LUIZ ROBERTO FORCAN	Aceito
Brochura Pesquisa	questionarioEngenhariaRequisitos.docx	28/09/2021 18:02:25	LUIZ ROBERTO FORCAN	Aceito
Projeto Detalhado / Brochura Investigador	projeto_Pesquisa.docx	28/09/2021 18:01:04	LUIZ ROBERTO FORCAN	Aceito
Brochura Pesquisa	projetoDetalhadoBrochuraInvestigador.docx	28/09/2021 18:00:40	LUIZ ROBERTO FORCAN	Aceito
TCLE / Termos de Assentimento / Justificativa de Ausência	tcle_IHC.doc	28/09/2021 17:59:08	LUIZ ROBERTO FORCAN	Aceito
TCLE / Termos de Assentimento / Justificativa de	tcle_eng_Requisitos.doc	28/09/2021 17:58:44	LUIZ ROBERTO FORCAN	Aceito

Endereço: Rua Dr. Bacelar, 1212 4º andar  
 Bairro: Vila Clementino CEP: 04.026-002  
 UF: SP Município: SAO PAULO  
 Telefone: (11)5586-4086 E-mail: cep@unip.br





Continuação do Parecer: 5.021.838

Ausência	tcle_eng_Requisitos.doc	28/09/2021 17:58:44	LUIZ ROBERTO FORCAN	Aceito
Declaração de concordância	frm_intencao_de_pesquisa.pdf	28/09/2021 15:50:29	LUIZ ROBERTO FORCAN	Aceito
Declaração de Pesquisadores	termo_de_compromisso_do_pesquisador.pdf	28/09/2021 15:49:04	LUIZ ROBERTO FORCAN	Aceito
Solicitação Assinada pelo Pesquisador Responsável	carta_de_apresentacao.pdf	28/09/2021 15:47:33	LUIZ ROBERTO FORCAN	Aceito
Orçamento	orcamento.pdf	28/09/2021 15:43:02	LUIZ ROBERTO FORCAN	Aceito
Cronograma	cronograma.pdf	28/09/2021 15:41:39	LUIZ ROBERTO FORCAN	Aceito
Folha de Rosto	folhaDeRosto.pdf	28/09/2021 15:40:19	LUIZ ROBERTO FORCAN	Aceito

**Situação do Parecer:**

Aprovado

**Necessita Apreciação da CONEP:**

Não

SAO PAULO, 06 de Outubro de 2021

---

**Assinado por:**  
**Bettina Gerken Brasil**  
**(Coordenador(a))**

**Endereço:** Rua Dr. Bacelar, 1212 4º andar  
**Bairro:** Vila Clementino  
**UF:** SP **Município:** SAO PAULO  
**Telefone:** (11)5586-4086 **CEP:** 04.026-002  
**E-mail:** cep@unip.br