Artigo recebido em 10/07/2026
Publicado em 10/11/2026

Revista SODEBRAS – Volume 21
N° 225 – Janeiro/Dezembro 2026

UMA TÉCNICA PARA ESPECIFICAÇÃO DE HISTÓRIAS DE USUÁRIOS BASEADAS EM PROBLEMAS

PROPOSAL OF A TECHNIQUE FOR SPECIFYING PROBLEM-BASED USER STORIES

Lucimara dos Santos[1]
Maria Claudia F. P. Emer [2]
Paulo Cézar Stadzisz[3]

Resumo – A técnica de Histórias de Usuários tem amplo emprego na especificação de requisitos de software, principalmente nas abordagens ágeis. Ela traz como diferenciais o foco no usuário e a associação das ações requeridas do software aos benefícios advindos destas ações. Entretanto, considerando-se a Abordagem Baseada em Problemas (PBSRS), percebe-se que faltam às Histórias de Usuários informações sobre as causas originais que levam aos requisitos, que são os "problemas" do negócio. Neste artigo, propõe-se a adequação e a extensão das Histórias de Usuários por meio de uma nova técnica denominada HUBP – Histórias de Usuários Baseadas em Problemas. Esta técnica integra as relações entre os domínios de conhecimento dos Problemas, das Necessidades e dos Requisitos e desenvolve-se de forma iterativa, compatível com os métodos ágeis atuais. Como resultado, a técnica permite uma melhor verificação da completude e da corretude da especificação, levando a uma maior probabilidade de sucesso do software desenvolvido em termos de satisfação dos stakeholders.

Palavras-chave: Requisitos de Software. Histórias de Usuários. Engenharia de Software.

Abstract - The User Stories technique is widely employed in software requirements specification, especially in agile approaches. Its main distinguishing features are the focus on the user and the association of the software's required actions with the benefits derived from those actions. However, when considering the Problem-Based Approach (PBSRS), it is evident that the User Stories technique lacks information about the root causes of the requirements, namely, the business problems. Thus, this article proposes an adaptation and extension of the classical User Stories into a new technique called HUBP – Problem-Based User Stories. This technique integrates the relationships among the Problem, Needs, and Requirements knowledge domains. It evolves iteratively to remain compatible with current agile methods. As a result, the technique enables more effective verification of the specification's completeness and correctness, increasing the likelihood of stakeholder satisfaction.

Keywords: Software Requirements. User Stories. Software Engineering.

[1] Mestranda em Computação Aplicada (UTFPR); Graduada em Gestão da Tecnologia da Informação (Centro Universitário Claretiano). Contato: lucimara.santos@hc.ufpr.br.
[2] Doutora em Engenharia Elétrica (UNICAMP); Mestre em Informática (UFPR); Graduada em Matemática e Bacharel em Informática (UNIOESTE). Contato: mcemer@utfpr.edu.br
[3] Doutor em Automática e Informática (UFC/França); Mestre em Engenharia Elétrica e Informática Industrial (UTFPR); Graduado em Processamento de Dados (UFPR). Contato: stadzisz@utfpr.edu.br.

I. INTRODUÇÃO

A Especificação de Requisitos é uma atividade crítica que orienta todo o processo do desenvolvimento de software, definindo as condições e capacidades que o produto deverá alcançar (WASHIZAKI, 2024). A má determinação dos requisitos tende a gerar custos adicionais, atrasos e insatisfação dos clientes (DAVIS et al., 2006). Por outro lado, requisitos bem formulados mitigam os riscos do projeto, levando ao desenvolvimento de softwares que atendam melhor às demandas reais dos clientes.

Para fazer frente aos desafios da Especificação de Requisitos, a Engenharia de Software tem proposto técnicas e abordagens para auxiliar os profissionais da área, destacam-se as abordagens por elicitação e por Histórias de Usuários.

As Histórias de Usuários foram propostas por Kent Beck (1999) como alternativa às técnicas tradicionais de especificação de requisitos (WASHIZAKI, 2024). Elas consistem em descrições textuais curtas e informais de funcionalidades ou características do software sob a perspectiva do usuário (BECK, 1999). A ampla adoção das histórias de usuário pode ser atribuída ao foco no usuário e à natureza colaborativa, além de estender a especificação dos requisitos com a indicação dos benefícios de cada um (Cohn, 2004).

Em 2001, o projeto Connextra propôs um *template* de Histórias de Usuário que foi, posteriormente, popularizado por Mike Cohn (2004) no seguinte formato: "Eu como um <papel>, Quero <objetivo>, Para que <benefício>". Esse *template* considera três componentes: o “papel”, que indica uma responsabilidade ou classe de stakeholder; o “objetivo”, que indica uma ação desejada para o software; e o “benefício”, que indica um ganho ou vantagem a ser obtido. Como exemplo, Cohn (2004) descreveu a seguinte história: “Eu, como um candidato a emprego, quero pesquisar por vagas de trabalho para que eu possa encontrar vagas que correspondam às minhas habilidades.”.

Apesar de sua popularidade, a aplicação prática das Histórias de Usuários revela limitações, como a informalidade e a brevidade, que geram ambiguidades, inconsistências e especificações incompletas. A falta de critérios rigorosos pode gerar descrições vagas, dificultar a estimativa de esforço e levar a interpretações incorretas (AMNA; POELS, 2022). O "benefício" esperado pode ser impreciso ou confundir-se com a própria solução funcional do sistema (SILVA, 2024). Essa imprecisão oculta as verdadeiras causas da demanda e compromete a rastreabilidade causal dos requisitos (AMNA; POELS, 2022; WANG et al., 2023).

Neste contexto, este artigo apresenta uma proposta de adequação e extensão das Histórias de Usuários em uma técnica denominada Histórias de Usuários Baseadas em Problemas (HUBP), baseada na abordagem PBSRS – *Problem Based Software Requirements Especification* (SOUZA; STADZISZ, 2016).

A abordagem PBSRS visa contribuir para melhorar a qualidade da atividade de Requisitos de Software, por meio de uma conceituação mais precisa, de modo que o entendimento e a especificação dos requisitos sejam mais efetivos (SILVA, 2024). Essa abordagem envolve o Domínio dos Problemas e o Domínio das Necessidades, além do Domínio dos Requisitos, permitindo que o especialista desenvolva a especificação de maneira mais sistemática e compreensível.

II. PROPOSTA

2.1 Método de Pesquisa

Esta pesquisa insere-se na área de Ciência da Computação, tendo como objeto de estudo a atividade de Engenharia de Requisitos. Quanto à sua natureza, classifica-se como

pesquisa aplicada, orientada à geração de conhecimento prático para a melhoria da especificação de requisitos (GIL, 2017). Em relação aos objetivos, o estudo possui caráter exploratório, visando estruturar uma técnica para minimizar problemas de completude e de corretude nas Histórias de Usuário. A abordagem do problema tem caráter qualitativo, pois concentra-se na compreensão da comunicação e da especificação na equipe de desenvolvimento, sem focar na representatividade estatística (GIL, 2017). Quanto aos procedimentos técnicos, dividiu-se em pesquisa bibliográfica para sustentação teórica e em exemplos de aplicação para análise da proposta (YIN, 2015).

O desenvolvimento da pesquisa estruturou-se em sete etapas: definição do escopo e do problema; levantamento bibliográfico e fundamentação teórica; proposição da técnica de Histórias de Usuário Baseadas em Problemas (HUBP); planejamento e execução da aplicação; e análise qualitativa dos resultados.

2.2 Conceituação Empregada

A técnica proposta para especificação de Histórias de Usuários baseada em Problemas se apoia nos conceitos de Problema, Necessidade e Requisito da PBSRS, agrupados nos seguintes domínios.

- **Domínio dos Problemas**

Este domínio contém o conhecimento das causas-raiz para os requisitos e descreve os problemas dos *stakeholders* que motivam a busca por uma solução de software. Um Problema é definido como "uma obrigação, expectativa ou esperança que, se não atendida, gera uma penalidade para o *stakeholder*, afetando o negócio relacionado". Assim, entende-se que todas as funcionalidades do software serão construídas para minimizar ou resolver algum problema dos *stakeholders*.

O formato de definição de um problema utilizado é:

<Stakeholder> <Obrigação | Expectativa | Esperança> <Penalização>

Cada problema é definido indicando-se o *stakeholder*, a obrigação, a expectativa ou a esperança que recai sobre ele, e a penalização a que ele está sujeito quando a obrigação não é cumprida.

Como exemplo, considere a informatização do Controle de Estoque de uma empresa em que o Diretor é um dos *stakeholders*. Uma de suas obrigações está relacionada ao desempenho do setor; ou seja, o problema seria a obrigação de "dever garantir o desempenho", conforme exemplificado a seguir.

Problema P1:	"O Diretor deve garantir o desempenho do Setor de Estoque, sob pena de gerar perdas financeiras e afetar o setor de produção."

Problemas também podem definir expectativas e esperanças que recaem sobre os *stakeholders* e que, mesmo não sendo obrigações, acarretam penalidades quando não são cumpridas.

- **Domínio das Necessidades**

Este domínio contém o conhecimento sobre as "necessidades" dos *stakeholders*, visando à solução ou à minimização dos problemas elencados. Uma necessidade descreve algo que o software "entregará" de forma útil aos s*takeholders*. Entretanto, este domínio não define o que o software fará (*i.e.,* requisitos funcionais), nem em quais situações será

utilizado (*i.e.,* casos de uso), tampouco suas características (*i.e.,* requisitos não funcionais) (SILVA, 2024).

As necessidades expressam carências percebidas nos negócios, ou seja, algo que falta aos *stakeholders* envolvidos que ajudaria a resolver algum problema. Assim, a definição das necessidades parte da definição dos problemas e estabelece o que é preciso que o software forneça para minimizá-los ou resolvê-los.

O formato de definição de uma necessidade utilizado é:

<Stakeholder> <Verbo de Necessidade> <Fonte> <Objeto da Necessidade>

Como exemplo, considere o problema P1 de garantia de desempenho do setor de estoque apresentado no tópico anterior, ilustrado a seguir.

Necessidade N1	"O Diretor precisa de um software para ter conhecimento dos itens a serem comprados para repor o nível mínimo de estoque."

- **Domínio dos Requisitos**

Este domínio contém o conhecimento clássico sobre os Requisitos Funcionais e não Funcionais do software, visando prover o que foi especificado no "Domínio das Necessidades". Cada Requisito Funcional é descrito por uma sentença, conforme a norma ISO/IEC/IEEE 29148 (2011), no seguinte formato:

<Sujeito> <Verbo> <Objeto> <Restrição> <Condição>

O sujeito descreve quem executa a ação (*e.g.,* software); o verbo indica a ação; o objeto é o elemento que sofre a ação; a restrição define regras e limitações; e a condição estabelece pré-requisitos.

Como exemplo, considere a necessidade N1 apresentada no tópico anterior, ilustrada a seguir.

Requisito R1	"O Software deverá apresentar um relatório diário contendo a lista dos itens de estoque que devem ser repostos."

2.3 Notação para Histórias de Usuários Baseadas em Problemas

Nesta pesquisa, propõe-se adequar e estender as Histórias de Usuários de acordo com a abordagem PBSRS. Assim, com base nos três domínios de conhecimento da especificação baseada em problemas, propõe-se a seguinte notação para as Histórias de Usuários Baseadas em Problemas.

Eu como <Stakeholder>, Visando resolver <Problema>, Preciso <Necessidade>, Por Meio <Ação do Software>.

Esta notação estende as Histórias de Usuário ao incluir os problemas dos *stakeholders* que a história ajuda a resolver ou a minimizar. Além disso, essa notação substitui o conceito de "Benefício" pelo de "Necessidade" conforme a conceituação da PBSRS. Essa adequação traz como vantagem a substituição de um conceito mais difuso, pois "Benefício" remete a ganho, proveito ou vantagem, por um conceito mais concreto, pois "Necessidade" remete ao que é provido pelo software e é mais mensurável.

Um exemplo dessa notação seria: “Eu como Usuário, Visando resolver a obrigação de manter estoques mínimos, Preciso ter conhecimento do que e quando comprar, Por meio de um alerta diário de reposições necessárias”.

2.4 Canvas para Especificação das Histórias de Usuários Baseadas em Problemas

As Histórias de Usuários Baseadas em Problemas podem ser especificadas textualmente e incorporadas ao *Product Backlog* tal qual as histórias convencionais. Entretanto, dado que as relações causais entre os três domínios de conhecimento (Problemas, Necessidades e Requisitos) frequentemente não serão biunívocas (*i.e.*, um problema pode gerar mais de uma necessidade, que pode gerar mais de um requisito, e estes podem agir para atender mais de uma necessidade e resolver mais de um problema), uma representação visual foi proposta para auxiliar na interpretação da especificação (Figura 1). Assim, um modelo (Canvas de Especificação de Histórias de Usuários) foi proposto, na forma de uma tabela, conforme ilustrado a seguir. Ele permite tabular as histórias na forma de linhas e estabelecer as associações entre os conceitos na forma de colunas, facilitando o gerenciamento da construção gradual da especificação ao longo do desenvolvimento.

Figura 1 – Modelo Canvas proposto para Especificação de Histórias de Usuários.

<table>
<tr><th>Stakeholders</th><th>Problemas</th><th>Necessidades</th><th>Requisitos</th></tr>
<tr><td rowspan="2">Lista dos stakeholders relacionados com o software</td><td>Lista das obrigações, expectativas e esperanças</td><td rowspan="2">Lista das necessidades relacionadas com os problemas de cada stakeholder.</td><td>Lista dos requisitos funcionais e não funcionais</td></tr>
<tr><td>Associação dos problemas aos stakeholders.</td><td>Associação dos requisitos às necessidades.</td></tr>
</table>

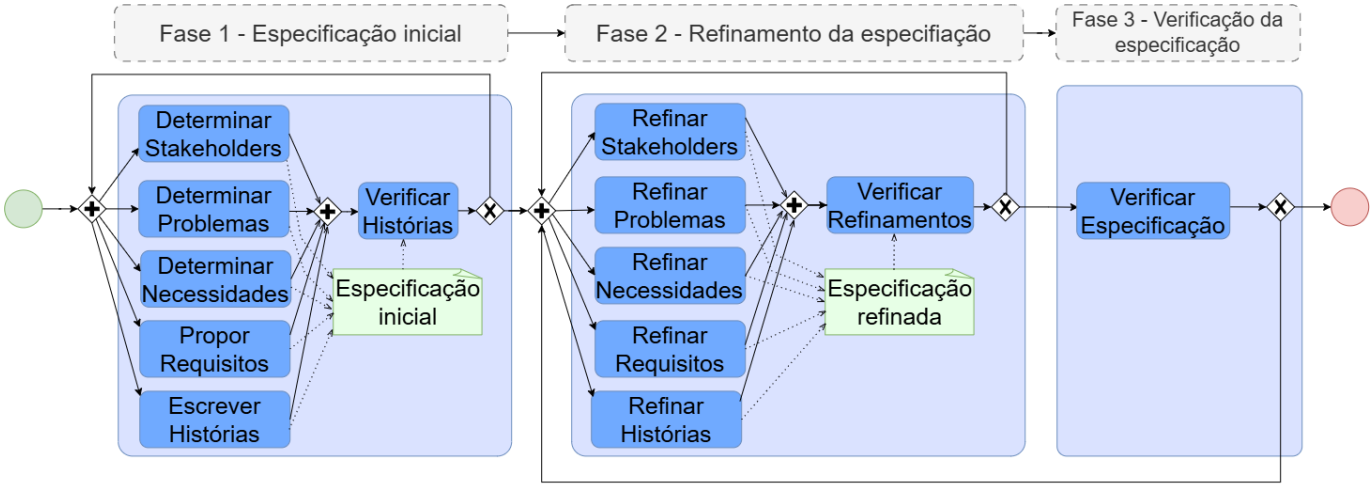
Fonte: Autores.

2.5 Descrição da Técnica Proposta

A técnica proposta neste artigo é organizada em três fases, conforme ilustrado na Figura 2 e descrito a seguir.

A aplicação da técnica avança de forma sequencial, da Fase 1 à Fase 3, em paralelo ao processo de desenvolvimento. Nas descrições a seguir, o Scrum será considerado como referência de processo.

Figura 2 – Fluxo de Atividades da Técnica de Especificação de Histórias de Usuários.



Fonte: Autores.

Fase 1 - Especificação Inicial de Histórias de Usuários

Essa fase ocorre no início do processo de desenvolvimento e integra a atividade de especificação do *Product Backlog* realizada pelo PO, conforme previsto no Scrum (SCHWABER; SUTHERLAND, 2020). Nessa fase, o PO construirá uma especificação "inicial" do software que será, posteriormente, refinada nas Sprints do processo iterativo. As histórias podem inicialmente apresentar caráter épico, abstrato ou incompleto e serão detalhadas mais à frente. As atividades que compõem essa fase são:

- **Determinação dos *Stakeholders***

O PO deverá determinar o conjunto de *stakeholders* que serão considerados. Em geral, os *stakeholders* são definidos de acordo com os papéis ou responsabilidades-chave dos profissionais no negócio que será informatizado.

- **Análise e determinação dos problemas dos *Stakeholders***

O PO irá determinar quais são os "problemas" dos *Stakeholder* a serem resolvidos pelo software. Os problemas decorrem das responsabilidades dos *stakeholders* nos negócios e, assim, o PO deverá interagir com eles para identificar seus problemas.

- **Análise e determinação das necessidades dos *Stakeholders***

O PO identificará as "necessidades" relacionadas aos problemas. Elas expressam o que falta aos *stakeholders* para a solução de seus problemas e o que será provido pelo software. A determinação das necessidades exige interações do PO com os *stakeholders* para analisar o que falta a eles para resolverem seus problemas considerando a "visão do software".

- **Análise e proposição dos requisitos para o Software**

O PO deve propor os requisitos funcionais e não funcionais do software, interagindo com os *stakeholders*. Cada requisito deve estar associado à necessidade que o originou. É comum que uma única necessidade dê origem a mais de um requisito, uma vez que os requisitos são um detalhamento técnico de como o software irá entregar o que a necessidade especifica.

- **Escrita das Histórias de Usuários baseadas em Problemas**

A escrita das histórias pode ser feita tanto textualmente, com a notação proposta, quanto de forma estruturada, por meio do Canvas de Especificação.

- **Verificação das Histórias de Usuários**

O PO verificará a completude e corretude da especificação. A completude é alcançada quando cada problema possui pelo menos uma necessidade e um requisito associados, indicando que todos os problemas foram cobertos. A corretude refere-se a especificações desassociadas, ou seja, requisitos que não produzem efeitos sobre nenhuma necessidade, necessidades que não colaboram para nenhum problema ou problemas que não recaem sobre nenhum *stakeholder*.

Fase 2 - Refinamento da Especificação de Histórias de Usuários

A Fase 2 da técnica envolve o refinamento das Histórias de Usuário especificadas, ao longo de todas as iterações (*i.e.*, Sprints). Esta fase alinha-se com a atividade de "Refinar e Ajustar o *Product Backlog*" prevista no processo Scrum (SCHWABER; SUTHERLAND, 2020). As atividades que compõem essa Fase 2 são:

- **Refinamento dos *Stakeholders***

Ao longo dos Sprints, o PO pode constatar a necessidade de incluir ou excluir *stakeholders*. A inclusão decorreria de uma falha na análise dos interessados realizada na Fase 1 ou de uma má interpretação das responsabilidades de um stakeholder. A exclusão pode decorrer de responsabilidades que extrapolem o escopo do software ou se confundem com outras responsabilidades.

- **Refinamento dos Problemas**

Com o avanço das Sprints, o PO pode perceber como os problemas estão sendo cobertos pelo software. Assim, o PO pode refinar os problemas, incluindo, excluindo, conjugando ou decompondo problemas ou, ainda, ajustando a terminologia e descrição dos problemas.

- **Refinamento das Necessidades**

Ao longo das Sprints, a especificação das necessidades pode precisar de refinamentos em razão de ajustes nos problemas ou do conhecimento adquirido com o avanço do desenvolvimento, levando à criação, exclusão, junção ou decomposição de necessidades, bem como a ajustes nas respectivas descrições.

- **Refinamento dos Requisitos**

Os requisitos especificados podem ser refinados pelo PO à medida que as Sprints avançam, para incluir novas exigências funcionais e não funcionais. Também os refinamentos podem surgir a partir do detalhamento das exigências e das adaptações requeridas em razão do *feedback* dos *stakeholders*.

- **Refinamento das Histórias**

As histórias de usuários descritas no Canvas são refinadas simultaneamente ao refinamento dos *stakeholders*, Problemas, Necessidades e Requisitos. Assim, o PO poderia atualizar as histórias no *Product Backlog* ao final de cada Sprint ou sempre que ocorresse algum refinamento em qualquer componente do Canvas.

- **Verificação dos Refinamentos**

Esta atividade é análoga à "Verificação das Histórias de Usuários" da Fase 1, na qual se avaliam a completude e a corretude das Histórias de Usuários.

Fase 3 - Verificação da Especificação de Histórias de Usuários

Concluídas as Sprints, verifica-se a qualidade da especificação examinando os pontos seguintes:

- **Completude e corretude da especificação:** verifica-se se a especificação está completa e correta, de forma análoga ao que foi proposto em cada Sprint.
- **Testes de Aceitação:** envolvem a constatação das funcionalidades e também avaliam em que grau estas funcionalidades entregam o que foi especificado nas Necessidades e em que grau resolvem os problemas dos *stakeholders*.
- **Independência Funcional das Histórias de Usuários:** avaliação das dependências funcionais entre as histórias. Caso uma história apresente relações cruzadas que interfiram na resolução dos problemas, caracterizam-se acoplamentos e reduções de coesão (SOUZA, 2016, p. 78).
- **Grau de satisfação:** verificação da efetividade das funcionalidades no atendimento aos problemas e às necessidades.

III. RESULTADOS

3.1 Aplicação

Esta seção ilustra um caso de especificação de um software de *HelpDesk* cujo papel é permitir que usuários de informática de uma empresa solicitem auxílio a técnicos de suporte. A determinação dos *stakeholders* levou à definição dos seguintes papéis: Usuário de Informática, Técnico de Suporte e Gerente de Suporte, o que permitiu preencher a primeira coluna do Canvas (Figura 3). A seguir, determinaram-se os três problemas relevantes dos *stakeholders* (P1 a P3), descritos nas respectivas colunas do Canvas e relacionados aos *stakeholders* pelo símbolo "X". A partir dos problemas, foi determinado o que o software deverá prover, indicado na coluna de necessidades (N1 a N3) para cada par <Stakeholder, Problema>. Por fim, foram definidos os requisitos (R1 a R3) para cada necessidade, especificados nas colunas mais à direita.

Figura 3 – Especificação das Histórias de Usuários no Canvas.

Stakeholders	Problemas			Necessidades	Requisitos		
	P1 - Notificar a empresa em caso de falha em equipamentos e sistemas de informação	P2 - Receber as notificações de falhas em equipamentos e sistemas de informação	P3 - Notificar o Usuário de Informática sobre a resolução das falhas		R1 - Permitir criar uma notificação de falha	R2 - Permitir visualizar as notificações de falhas	R3 - Permitir criar uma notificação de resolução das fahas
S1 - Usuário de Informática	X			N1 - Poder criar uma notificação de falha	X		
S2 - Analista de Suporte		X		N2 - Ter conhecimentos sobre as notificações		X	
S3 - Analista de Suporte			X	N3 - Ter conhecimento sobre o status das notificações			X

Fonte: Autores.

Cada linha do Canvas representa uma história de usuário, como no caso da primeira que teria a seguinte composição textual: "Eu como Usuário de Informática, Visando resolver a obrigação de notificar a empresa em caso de falha de equipamentos e sistemas de informação, Preciso poder criar uma notificação de falha, Por meio da ação do software de criar uma notificação de falha".

3.2 Contribuições da Pesquisa

As contribuições desta pesquisa foram: uma nova notação, um modelo de representação (Canvas) e uma técnica de especificação de Histórias de Usuários Baseadas em Problemas. As principais contribuições destes resultados são:

- **Originalidade de conceitos**: as proposições desta pesquisa diferenciam-se por empregar conceitos que estendem aqueles clássicos da engenharia de requisitos, mas que, ao mesmo tempo, são compreensíveis e enriquecem a capacidade de expressão dos requisitos;
- **Amplitude da abordagem**: as proposições proporcionam maior alcance na análise de requisitos, explorando as relações causais entre a demanda real e os requisitos do software;
- **Sistemática da abordagem**: as proposições estabelecem conceitos, formas de expressão e de verificação e um procedimento que sistematizam o desenvolvimento da especificação; e
- **Compatibilidade com outras abordagens**: as proposições puderam ser bem integradas a processos iterativos, notadamente aos métodos ágeis, enriquecendo-os.

IV. CONCLUSÃO

Este artigo apresentou proposições para uma abordagem de especificação de Histórias de Usuários Baseadas em Problemas, incluindo conceitos, uma notação, um modelo Canvas e uma técnica de especificação. Embora a pesquisa ainda não tenha desenvolvido estudos de caso que apresentem resultados comprobatórios, a abordagem tem se mostrado funcional nos exemplos analisados, além de se integrar adequadamente aos processos iterativos, como o Scrum. O Canvas, em especial, mostrou-se uma ferramenta útil para a composição de histórias de usuário e para o mapeamento das interrelações entre os conceitos que as compõem. Como trabalhos futuros, esta pesquisa buscará aprofundar a validação dos resultados por meio de estudos aplicados com profissionais da área de software, em casos reais de desenvolvimento de software.

V. REFERÊNCIAS

AMNA, Anis R.; POELS, Geert. *Systematic literature mapping of user story research.* **IEEE Access**, v. 10, p. 51723-51746, 2022.

BECK, Kent. *Embracing change with extreme programming*. **Computer**, v. 32, n. 10, p. 70-77, 1999.

COHN, Mike. ***User Stories Applied: For Agile Software Development***. Boston: Addison-Wesley, p. 145, 2004.

DAVIS, Alan et al. Identifying and classifying software requirements anomalies. Journal of Systems and Software, v. 79, n. 9, p. 1364-1376, 2006.

GIL, Antonio C. ***Como elaborar projetos de pesquisa***. 6. ed. São Paulo: Atlas, 2017.

ISO/IEC/IEEE 29148 (2011). ***Systems and software engineering — Life cycle processes — Requirements Engineering***. Genebra: ISO/IEC/IEEE, 2011.

SILVA, Andressa H. I.; STADZISZ, Paulo C. *Uma Abordagem para Associação entre Necessidades do Cliente e Requisitos de Software.* **Revista SODEBRAS**, 2024.

SCHWABER, Ken; SUTHERLAND, Jeff. *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game.* **Scrum Guides**, 2020.

SOUZA, Rafael G. M.; STADZISZ, Paulo C. *Problem Based Software Requirements Specification*, **Revista Eletrônica de Sistemas de Informação**, 2016.

WASHIZAKI, Hironori (Ed.). Guide to the Software Engineering Body of Knowledge (SWEBOK Guide). Version 4.0. Piscataway: IEEE Computer Society, 2024.

YIN, Robert K. ***Estudo de caso: planejamento e métodos***. 5. ed. Porto Alegre: Bookman, 2015.

WANG, Chunhui et al. User story clustering in agile development: a framework and an empirical study. Scientific Reports, v. 13, n. 1, art. 17621, p. 1-13, 2023.

VII. COPYRIGHT

Direitos autorais: os autores são os únicos responsáveis pelo material incluído no artigo.