

Revisão Bibliográfica Sobre o Uso de Base de Conhecimento para Gestão de Dívida Técnica de Requisitos

Deborah de Assis Pereira dos Santos¹, Napoleão Verardi Galegale²;

RESUMO

O tema de dívida técnica vem sendo discutido ao longo dos anos, desde a apresentação desta metáfora por Ward Cunningham (1992). Dentre os tipos de dívida técnica, há a dívida de requisitos. Este artigo tem como objetivo identificar quais são as abordagens das pesquisas sobre o uso de base de conhecimento para gestão de dívida técnica de requisitos, publicadas entre 2015 e 2022. O método utilizado foi revisão da literatura e análise de conteúdo com a aplicação baseada no protocolo PRISMA-P, mediante pesquisa do tipo exploratória. Os resultados revelam que o tema de dívida técnica de requisitos vem sendo pouco discutido, sendo que a principal abordagem das pesquisas são dívida técnica de código e arquitetura. Não foi encontrada nenhuma publicação, dentre o período delimitado, sobre o uso de base de conhecimento para gestão de dívida técnica de requisitos. Em virtude do resultado obtido, este artigo traz a proposta do uso de base de conhecimento para tomada de decisão na gestão de dívida técnica de requisitos, por meio do processo de extração de conhecimento KDD (*Knowledge Discovery in Database*).

Palavras-chave: Dívida Técnica, Dívida de Requisitos, Base de Conhecimento, KDD

ABSTRACT

The issue of technical debt has been discussed over the years, since the introduction of this metaphor by Ward Cunningham (1992). Among the types of technical debt, there is the requirement debt. This article aims to identify what are the approaches of research on the use of knowledge base for requirements technical debt management, published between 2015 and 2022. The method used was literature review and content analysis with the application based on the PRISMA-P protocol, through exploratory research. The results reveal that the issue of technical debt of requirements has been little discussed, and the main research approach is technical debt of code and architecture. No publication was found, within the delimited period, on the use of knowledge base for requirements technical debt management. Due to the result obtained, this article proposes the use of a knowledge base for decision making in the management of technical debt of requirements, through the KDD (*Knowledge Discovery in Database*) knowledge extraction process.

Keywords: Technical Debt, Requirements Debt, Knowledge Base, KDD

1 Introdução

O conceito de dívida técnica (DT) foi introduzido por Ward Cunningham (1992), como uma metáfora ao conceito de dívida financeira. A DT pode incorrer por meio de uma decisão em obter benefícios de curto prazo (aumento da produtividade e/ou menor custo) em troca da qualidade do software. Isto pode ocasionar entregas de artefatos imaturos e incompletos.

Assim como no contexto financeiro, a DT gera juros. Quanto mais tempo se tarda para o pagamento dos juros, maiores estes se tornam, impactando os próximos projetos. Sem contar os altos custos de manutenção dos sistemas. De acordo com Li, Zengyang et. al (2015) há a classificação de 10 tipos de DT que são de: arquitetura, design, código, teste, construção (*build*), documentação, infraestrutura, versionamento, defeito e requisitos.

A literatura apresenta com bastante ênfase estudos sobre o tipo DT de código, porém esse mesmo cenário é antagônico quando se trata de DT de requisitos. O levantamento de requisitos é uma etapa importante no ciclo de desenvolvimento de software, onde as necessidades do usuário são capturadas. Com isso, a equipe de desenvolvimento pode transformar os requisitos em funcionalidades do sistema. Requisitos de software inadequados e mal definidos é um dos fatores que causa falha nos projetos (MCMANUS12, John; WOOD-HARPER, Trevor, 2007).

Somente após 20 anos do surgimento do termo DT que Neil Ernst (2012) mencionou pela primeira vez o conceito de dívida técnica de requisitos, referindo-se à distância entre a especificação ótima de requisitos e a implementação real do sistema. Os requisitos são a validação final do sucesso do projeto, uma vez que são a representação em sistema dos anseios do usuário. Questões técnicas do projeto, como requisitos de software inapropriados ou mal definidos podem responder por 35% das falhas dos projetos (MCMANUS12, John; WOOD-HARPER, Trevor, 2007). Entendendo a relevância dos requisitos como um dos fatores de sucesso de um projeto, é notória a importância em ter conhecimento sobre dívida técnica de requisitos do produto.

Se em cada versão do produto pudesse ser registrado em uma base de dados os requisitos que foram implementados (completos e incompletos) e os que não foram implementados, seria possível ter uma visão sobre a distância do que especificado e o que de fato foi entregue. Por meio de uma base de conhecimento, seria possível acessar estas informações, permitindo a possibilidade de gestão de dívida técnica de requisitos do produto.

As bases de conhecimento podem ser entendidas como ferramentas informacionais abordadas por usuários que buscam informações e que, como tal, devem refletir as necessidades dos usuários (Todd et. al, 1994). O processo de KDD (*Knowledge Discovery in Databases*) foi formalizado em 1989 em referência à procura de conhecimento a partir de bases de dados. O processo de KDD pode ser definido como não trivial em identificar padrões válidos, novos, potencialmente úteis e, em última análise, compreensíveis em dados (Usama Fayyad, et.al 1996).

A base de conhecimento (*knowledge base*) pode ser compreendida como uma ferramenta que centraliza todos os dados e informações relevantes que a empresa tem sobre as soluções que ela oferece. Um dos grandes benefícios em sua utilização, é identificar desperdício de tempo para solucionar um problema. A base de conhecimento evolui com o ciclo tecnológico (SAVIOTTI, 2009).

Diante deste contexto, este artigo tem como principal objetivo responder a seguinte questão de pesquisa: quais são as abordagens das pesquisas sobre o uso de base de conhecimento para gestão de dívida técnica de requisitos? Para atingir este objetivo principal, foram determinados os seguintes objetivos específicos: realizar a revisão da literatura, complementada com a bibliometria e a análise de conteúdo sobre dívida técnica de requisitos. E a partir desta questão de pesquisa, apresentam-se duas proposições:

P1) não há oportunidade de aprofundamento na literatura quanto ao uso de base de conhecimento para gestão de dívida técnica de requisitos.

P2) há oportunidade de aprofundamento na literatura quanto ao uso de base de conhecimento para gestão de dívida técnica de requisitos.

Este estudo faz parte de uma pesquisa maior que é a oportunidade de desenvolvimento de uma base de conhecimento para gestão de dívida técnica de requisitos, havendo com este artigo o início desse ensejo.

2 Referencial Teórico

O tema sobre dívida técnica (DT) foi introduzido por Ward Cunningham (1992), trazendo uma metáfora ao conceito de dívida financeira. Desde então, diversos autores explanaram sobre esse conceito. De acordo com McConnell (2008), a DT pode ser classificada como intencional e não intencional. A DT intencional é dívida técnica que se assume conscientemente como ferramenta estratégica, onde os benefícios são maiores do que a consequência da DT contraída no momento. Já a DT não intencional é quando não a dívida técnica é contraída de forma acidental, inconscientemente.

Em 2009, Martin Fowler complementou o conceito de McConnell, incluindo mais duas classificações: a DT prudente e imprudente. Desta forma, tem-se formado o quadrante da DT com os tipos: intencional e imprudente, intencional e prudente, não intencional e imprudente, não intencional e prudente. Assim como uma dívida financeira, a DT também pode gerar juros. Se a DT não for corrigida, os juros se acumularão, conseqüentemente a melhoria e a manutenção do software também serão mais difíceis de implementar (Allman, Eric 2012). Incurrendo no aumento do esforço de desenvolvimento e diminuição da produtividade, visto que a dívida deverá ser paga em algum momento (Seaman e Guo, 2011).

A DT, em geral, denota que a equipe de desenvolvimento escolheu um atalho para obter benefícios de curto prazo. Isto pode prejudicar o software a longo prazo, visto que há consideráveis possibilidades de manutenção constante para mantê-lo operante. Se não bem gerenciada, a dívida técnica faz com que os projetos enfrentem problemas técnicos e financeiros relevantes (KRUCHTEN, Philippe, et al., 2012). Uma das características notáveis da dívida técnica é sua natureza interdisciplinar, visto que combina elementos da teoria financeira e da engenharia de software (Ampatzoglou, Areti 2015).

2.1 Dívida Técnica de Requisitos

O conceito de DT pode ser encontrado na literatura com bastante ênfase na parte técnica de desenvolvimento de código. Somente após 20 anos do surgimento do conceito de DT, que foi definido a concepção de dívida técnica de requisitos, por

Neil Ernst (2012). Ele definiu a dívida de requisitos como a distância entre a especificação ótima de requisitos e a implementação real do sistema. A dívida técnica também pode se manifestar na fase de requisitos dos sistemas (ERNST, 2012).

A DT de requisitos ocorre quando é decidido priorizar requisitos que não são necessários e nem entregam maior valor para o cliente (ERNST, 2012). Uma maneira de entender mais sobre a DT de requisitos é olhar a relação entre o valor do produto e a qualidade do software. Se o produto não está entregando valor ao cliente então a alta qualidade é irrelevante. Se a qualidade do software é baixa, ou seja, tem muita dívida técnica, a chance de entrega de um produto com valor é baixa.

Os requisitos são a validação final do sucesso do projeto, uma vez que são a representação em sistema dos desejos do usuário. A engenharia de requisitos é fundamental e central para todo projeto de desenvolvimento de software bem-sucedido. Há diversas razões pelas quais os projetos de software podem falhar, contudo requisitos mal elicitados, documentados, validados e gerenciados contribuem consideravelmente para o fracasso de projetos de software (HUSSAIN, Azham, et al., 2016).

A engenharia de requisitos está na base de todo projeto de software bem-sucedido. Há muitas razões para falhas em projetos de software; entretanto, processos de requisitos mal desenhados contribuem imensamente para a razão pela qual os projetos de software falham. A falha do projeto de software geralmente é cara e arriscada. O custo das falhas de um projeto é alto. Além disso, representam um desafio no ambiente de mercado competitivo, podendo afetar a imagem da empresa, a geração de receita e diminuir a satisfação percebida de clientes. A engenharia de requisitos é o alicerce sobre o qual se baseia o sucesso dos projetos de software (HUSSAIN, Azham; MKPOJIOGU, Emmanuel OC, 2016).

Mesmo após definidos e aprovados, os requisitos podem sofrer alterações durante o desenvolvimento do sistema. Um gerenciamento de mudanças de requisitos de software adequado não só minimiza o aumento de custo, tempo e recursos de projeto de software, mas também tornar o projeto bem-sucedido (Muhammad Wasim Bhatti, et al. 2010). Os requisitos são uma parte importante para o sucesso em projetos de software. Por conseguinte, ter dívida técnica de requisitos, pode contribuir para fracasso destes projetos.

Há a definição de 3 tipos de dívida técnica de requisitos, segundo Valentina Lenarduzzi e Davide Fucci (2019): **Tipo 0 - *Incomplete user's needs***, representa a dívida contraída ao negligenciar as necessidades de alguns stakeholders ou de um grupo específico de stakeholders. **Tipo 1 - *Requirements smells***, representa indicadores de má qualidade na especificação de requisitos, prejudicando negativamente a compreensão, manutenção e qualidade dos artefatos. São utilizados para inspecionar a qualidade das especificações de requisitos em linguagem natural. **Tipo 2 - *Mismatch implementation***, representa a dívida incorrida quando os desenvolvedores implementam uma solução incompatível com o objetivo e expectativa do usuário. Este tipo de dívida pode incorrer também quando os requisitos mudam e o desenvolvimento não foi alterado.

Ao lidar com a dívida técnica de requisitos, nos projetos de software, seria interessante poder registrar o conhecimento obtido. Essa prática adquirida pode ser útil como referência para futuros casos, a fim de até mesmo evitá-los.

2.2 Descoberta de conhecimento em Base de Dados (KDD)

O processo de KDD (*Knowledge Discovery in Databases*) foi formalizado em 1989 em referência à procura de conhecimento a partir de bases de dados. Surgiu como uma solução para descobrir informações e conhecimento de dados em rápido crescimento, permitindo a execução de processos interativos e iterativos para obter novas informações úteis dos dados.

O termo KDD refere-se ao processo geral de descoberta de conhecimento útil a partir de dados. O processo de KDD pode ser definido como não trivial em identificar padrões válidos, novos, potencialmente úteis e, em última análise, compreensíveis em dados (Usama Fayyad, et.al 1996). As principais etapas do KDD realizam a preparação e seleção de dados, limpeza de dados, transformação de dados, mineração de dados e interpretações, conforme detalhado abaixo:

- **Etapa de Seleção**, consiste em selecionar um conjunto de dados que farão parte da análise. As fontes de dados podem ser variadas e os dados podem ter formatos diferentes (estruturados, semiestruturados e não-estruturados).
- **Etapa de Processamento**, consiste em fazer a verificação da qualidade dos dados armazenados. É feito um processo de limpeza na base de dados, corrigindo e/ou removendo dados inconsistentes, dados ausentes ou incompletos, identificação de anomalias.
- **Etapa de Transformação**, consiste em aplicar técnicas de transformação. Os dados ficam disponíveis agrupados em um mesmo local para a aplicação dos modelos de análise.
- **Etapa de Mineração de Dados**, consiste em construir modelos ou aplicar técnicas de mineração de dados.
- **Etapa de Interpretação e Avaliação**, consiste em avaliar o desempenho do modelo. A validação pode ser feita de várias maneiras, como por exemplo, utilizar medidas estatísticas, passar pela avaliação dos profissionais de negócio, dentre outras.

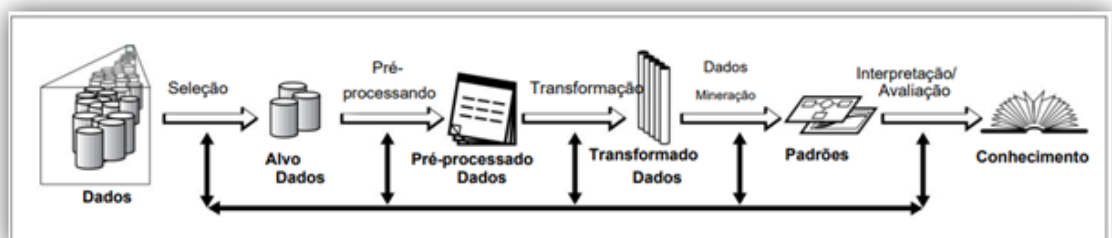


Figura 1 - Etapas do KDD (Usama Fayyad, et.al 1996)

O produto esperado resultante do processo KDD é o conhecimento. Com esse resultado é possível criar uma base de conhecimento relevante sobre determinado assunto.

2.3 Base de Conhecimento

A capacidade das organizações de gerar, registrar e compartilhar conhecimento é a base para a inovação e a competitividade nas organizações (DE CISNE, Caroline Santos, et al. 2019). O registro correto do conhecimento é

importante para apoiar decisões no ciclo de vida do produto. Ter ciência sobre o histórico dos requisitos de um produto de software seria possível por meio do registro em uma base de conhecimento, quanto a caracterização dos requisitos do produto, ou seja, os que foram implementados, os que não foram implementados ou os que estão incompletos.

As bases de conhecimento podem ser entendidas como ferramentas informacionais abordadas por usuários que buscam informações e que, como tal, devem refletir as necessidades dos usuários (Todd et. al, 1994). A interação com bases de conhecimento é um processo de comunicação em que o sucesso é determinado por um alinhamento das estruturas de conhecimento dos usuários e estruturas do sistema, ou seja, o que os usuários sabem e como estruturam o que sabem (Todd, 2006). Apesar da dificuldade de gerar a base de conhecimento de um sistema especialista, Todd (2006) justifica o uso dessa técnica para solução de problemas. A partir de uma base de conhecimento bem estruturada é possível apoiar no processo de atendimento das demandas.

Uma base de conhecimento tem a representação dos fatos usados para modelar informações relevantes para o domínio do problema com um conjunto de regras que permitem tirar conclusões a partir de fatos e estratégias de raciocínio. A capacidade de raciocínio, permite que um sistema responda de forma mais inteligente às afirmações e consultas (SYMONDS, 1988).

O conhecimento é o recurso primordial para as organizações no processo de tomada de decisão (ANGELONI, 2002). A tomada de decisão pode ser entendida como o processo de identificar um problema específico e selecionar uma linha de ação para resolvê-lo (STONER, James A. F. e FREEMAN, 1999).

3 Método

Para o desenvolvimento deste estudo, com característica qualitativa, foi realizada revisão da literatura sobre o tema dívida técnica de requisitos e o uso de base de conhecimento para gestão de dívida técnica de requisitos. Com base nos artigos encontrados, foi feita a análise de conteúdo com a aplicação baseada no protocolo PRISMA-P. Este protocolo foi desenvolvido como um roteiro para apoiar pesquisadores na realização de revisões que retornem um conjunto mínimo de itens importantes a serem considerados no protocolo de pesquisa (MOHER et al., 2015).

Na etapa de identificação, foi realizada a 1ª bibliometria nas bases *Web of Science* e *Scopus*, delimitando o período de 2015 até 2022, utilizando a sintaxe: “*technical debt*” no título, *abstract* e *keywords*. Como resultado da 1ª bibliometria foram obtidos 506 artigos, os quais foram triados na ferramenta excel para retirada de artigos duplicados. Resultou em 435 artigos, conforme listado no quadro 1.

Quadro 1 – Quantidade de artigos resultantes da 1ª bibliometria

Sintaxe: "technical debt"		
	Web Of Science	Scopus
	310	196
Total	506	
TOTAL SEM DUPLICADOS	435	

Fonte: autores (2022)

Após analisar o resultado da 1ª bibliometria, que decorreu em 435 artigos, na etapa de identificação foi constatada a necessidade em refinar a pesquisa delimitando a busca sobre dívida técnica de requisitos (*requirements debt*). Desta forma, foi realizada a 2ª bibliometria para identificação de resultados mais específicos. Além das bases *Web of Science* e *Scopus*, foi incluída a base Google Acadêmico.

O período de busca foi entre 2015 e 2022, utilizando a sintaxe: "*requirements debt*" AND "*technical debt*" no título, *abstract* e *keywords*. Os critérios de exclusão foram os tipos de registro livros, tese e dissertações. Foram obtidos como resultado da 2ª bibliometria, 6 artigos e uma tese, os quais foram triados na ferramenta excel para retirada de registros duplicados. A triagem resultou em 3 artigos, conforme ilustrado na figura 2.

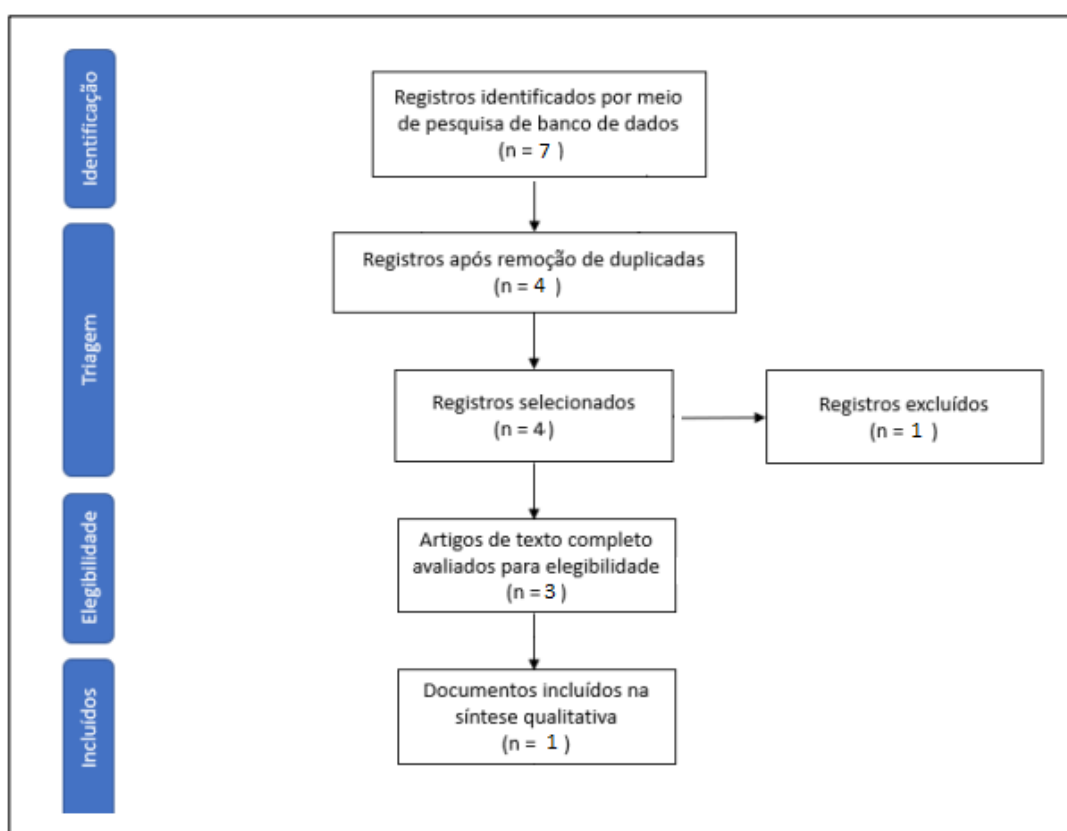


Figura 2 – Fluxograma baseado no protocolo PRISMA-P referente à 2ª bibliometria.

Fonte: autores (2022)

O critério de seleção para análise foi a leitura de todos os registros encontrados, no caso 3 artigos, devido ao resultado da triagem ter sido poucos documentos. Para complementar este estudo, foi incluído um artigo de 2012 devido a importância de ter trazido pela primeira vez o tema de dívida técnica de requisitos. Os registros selecionados estão listados no Quadro 2.

A etapa de análise é fundamental para identificar se há ou não oportunidade de aprofundamento na literatura para o uso de base de conhecimento para gestão de dívida técnica de requisitos. Foi feita a análise manual, sem uso de recurso de softwares específicos que suportam essa função.

Quadro 2 – Registros selecionados para análise

Título	Tipo	País	Ano
Using real options to manage Technical Debt in Requirements Engineering	Artigo	Canadá	2015
Towards a Holistic Definition of Requirements Debt	Artigo	Alemanha e Finlândia	2019
Identification and Measurement of Technical Debt Requirements in Software Development: a Systematic Literature Review	Artigo	Brasil e Finlândia	2021
On the role of requirements in understanding and managing technical debt	Artigo	Suíça	2012

Na 2ª bibliometria também foi realizada a pesquisa utilizando as sintaxes:

- “*requirements debt*” AND “*knowledge base*” no título, *abstract* e *keywords*. Foi obtido nenhum resultado, zero publicações.
- “*requirements debt*” AND “*KDD*” no título, *abstract* e *keywords*. Foi obtido nenhum resultado, zero publicações.

4 Resultados e Discussão

Com base nas 4 publicações selecionadas, foi realizada a análise de conteúdo sobre dívida técnica de requisitos. Pode-se observar como resultado, que dentro o período delimitado entre 2015 e 2022, que houve poucas publicações sobre o tema de dívida técnica de requisitos, conforme ilustrado abaixo na Tabela 1.

Tabela 1 – Quantidade de artigos resultantes da 2ª bibliometria, distribuídos entre os anos de 2015 e 2022

Sintaxe de pesquisa: "requirements debt" AND "technical debt"								
Quantidade de artigos publicados por ano sobre dívida técnica de requisito (requirements debt)								
Base pesquisada	2015	2016	2017	2018	2019	2020	2021	2022
Google Acadêmico	1	--	--	--	--	1	1	--
Web of Science	1	--	--	--	1	--	--	--
Scopus	1	--	--	--	1	--	--	--
Total	3	--	--	--	2	1	1	--
Total SEM artigos duplicados	1	--	--	--	1	1	1	--

4.1 Análise dos Dados

Foi realizada a leitura integral das 4 publicações e a abordagem das pesquisas foi estruturada por meio de análise de conteúdo. Há diferente abordagens sobre dívida técnica de requisitos nos artigos analisados. Em *"On the role of requirements in understanding and managing technical debt"* (ERNST, Neil 2012) é explanado o conceito da dívida técnica de requisitos, trazendo a definição como a distância entre a que foi especificado nos requisitos com o que foi implementado no sistema. No artigo *"Using real options to manage Technical Debt in Requirements Engineering"*, (ABAD, Zahra Shakeri Hossein; RUHE, Guenther 2015), foi descrito método quantitativo para gerenciar decisões de requisitos sob incerteza e para gerenciar dívidas de requisitos em projetos de desenvolvimento de software.

Na publicação *"Towards a Holistic Definition of Requirements Debt"*, (LENARDUZZI, Valentina; FUCCI, Davide 2019), foi conceituado a definição holística de dívida de requisitos que inclui dívida incorrida durante a identificação, formalização e implementação de requisitos. Foi explanado sobre a categorização de tipos de dívida técnica de requisitos: Incomplete user's needs, Requirements smells e Mismatch implementation. Por fim, no artigo *"Identification and Measurement of Technical Debt Requirements in Software Development: A Systematic Literature Review"* (MELO, Ana et al. 2021), foi realizada uma revisão sistemática da literatura, trazendo como objetivo identificação e mensuração de dívida técnica de requisitos em gerenciamento de software.

É possível observar que nenhum dos artigos trouxe informações sobre o uso de base de conhecimento, tampouco sobre KDD (*knowledge discovery in database*) para gestão de dívida técnica de requisitos. Isso demonstra que há uma lacuna de pesquisa para essa proposta.

4.2 Proposta do uso de base de conhecimento para tomada de decisão na gestão de dívida técnica de requisitos

Levando em consideração que a dívida técnica de requisitos pode ser entendida como a distância entre os requisitos que foram especificados e os

requisitos que de fato foram implementados (ERNST, 2012), é possível entender a importância em ter o conhecimento sobre os requisitos de um produto, quais foram implementados e de que forma. Os requisitos são um dos pontos que podem levar ao fracasso de um projeto. Requisitos mal implementados ou implementados diferente do escopo que foi definido, podem trazer problemas ao produto, como por exemplo, ocasionar dívida técnica de requisitos.

Por meio de um olhar atento sobre o produto, tendo acesso ao conhecimento sobre os requisitos registrados em uma base, haveria a possibilidade de utilizar estas informações para apoio na tomada de decisão para a gestão da dívida técnica de requisitos. Todd (2006), justifica o uso de base de conhecimento para solução de problemas. Para o bom funcionamento de uma organização, é importante que as tomadas de decisões sejam administradas de uma forma efetiva. O conhecimento é o recurso primordial para as organizações no processo de tomada de decisão (ANGELONI, 2002)

Para gerar uma base de conhecimento, é necessário fazer a extração do conhecimento. O processo KDD (Knowledge Discovery in Database) é usado para extração de conhecimento em uma base de dados. Considerando que este processo é iterativo e interativo para identificar nos dados novos padrões que sejam válidos (Fayad, 1996), a proposta em utilizar este processo como forma de gerar base de conhecimento para gestão de dívida técnica de requisitos, parece ser potencialmente aplicável.

A cada versão do produto, registrar em uma base de dados os requisitos que foram implementados (de forma completa e incompleta) e os que não foram implementados. A partir desta base de dados, o processo KDD poderia seguir as etapas necessárias para geração do conhecimento referente a identificação das dívidas técnicas de requisitos. Adquirido este conhecimento, uma base de conhecimento seria alimentada de forma cíclica, a cada versão do produto. Isso permitiria que diferentes equipes que estejam trabalhando em projetos do mesmo produto, tenham conhecimento sobre as dívidas técnicas de requisitos que o produto possui. E, com acesso a este conhecimento, o responsável pela tomada de decisão poder ter clareza e assertividade sobre qual dívida técnica de requisitos eliminar ou até mesmo manter, dependendo da estratégia de gestão utilizada.

Na Figura 3, é possível visualizar o exemplo desta proposta de uso de base de conhecimento para o campo mencionado.

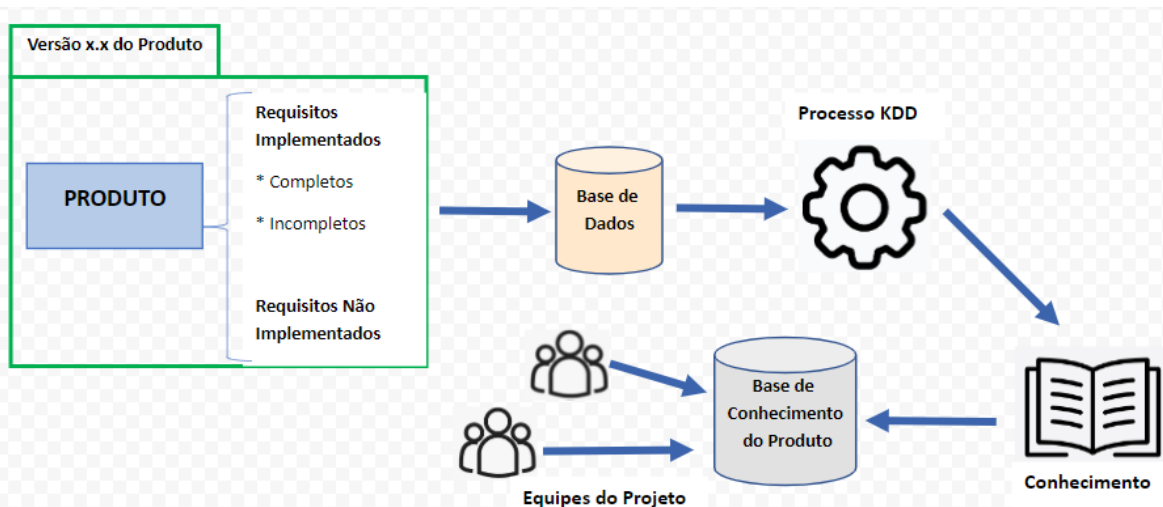


Figura 3: Proposta de uso de base de conhecimento para identificar dívida de requisitos do produto
Fonte: autores (2022)

O alcance a essas informações contribuiria na melhoria contínua do produto. Entender e apoiar o processo de evolução de requisitos é um problema importante e difícil (Ghose, 2000). Uma base de conhecimento evolui ciclicamente junto com o progresso técnico, alternando fases de urgência e maturação de novos paradigmas técnico-científicos (Saviotti, 2009).

Como resultado deste estudo, foi observado que há uma lacuna de pesquisa na literatura quanto ao uso de base de conhecimento para gestão de dívida técnica de requisitos. Diante do exposto, este artigo trouxe com resultado a proposta a criação de uma base de conhecimento, através do processo KDD para gestão de dívida técnica de requisitos.

5 Considerações finais

O presente artigo teve como objetivo verificar se há na literatura a oportunidade do uso de base de conhecimento para gestão de dívida técnica de requisitos. Para isto, foi feita a revisão bibliográfica na literatura entre o período entre 2016 e 2022, buscando o que foi publicado sobre dívida técnica de requisitos. O resultado desta pesquisa confirma a proposição P2 em que há oportunidade de aprofundamento de pesquisa na literatura sobre o uso de base de conhecimento para gestão de dívida técnica de requisitos, haja vista que não foi identificado nenhum artigo sobre este tema dentre todos os documentos analisados. E refuta a proposição P1 em que não há oportunidade de aprofundamento a literatura a oportunidade do uso de base de conhecimento para gestão de dívida técnica de requisitos.

Diante do exposto, este artigo trouxe como resultado a proposta do uso de base de conhecimento para tomada de decisão na gestão de dívida técnica de requisitos, utilizando o processo de extração de conhecimento KDD (*Knowledge Discovery in Database*). A pouca quantidade de artigos sobre o tema de dívida técnica de requisitos, trouxe como resultado a viabilidade do uso de base de conhecimento para a gestão de dívida técnica de requisitos, se apresentando assim como sugestão para estudos futuros para pesquisadores que tenham interesse.

Referências

ABAD, Zahra Shakeri Hossein; RUHE, Guenther. Using real options to manage technical debt in requirements engineering. In: 2015 IEEE 23rd International Requirements Engineering Conference (RE). IEEE, 2015. p. 230-235.

ALLMAN, Eric. Managing technical debt. *Communications of the ACM*, v. 55, n. 5, p. 50-55, 2012.

AMPATZOGLU, Areti et al. The financial aspect of managing technical debt: A systematic literature review. *Information and Software Technology*, v. 64, p. 52-73, 2015.

ANGELONI, Maria Terezinha (Org). *Organizações do Conhecimento: infraestrutura, pessoas e tecnologia*. São Paulo: Saraiva, 2002.

BHATTI, Muhammad Wasim et al. An investigation of changing requirements with respect to development phases of a software project. In: 2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM). IEEE, 2010. p. 323-327.

BORDES, Antoine et al. Learning structured embeddings of knowledge bases. In: Twenty-fifth AAAI conference on artificial intelligence. 2011.

DE CISNE, Caroline Santos; KANEOYA, Paula Hidemi; DOS SANTOS, Luana Carla Moura. Compartilhamento e registro de conhecimento: um estudo de caso na empresa Knowtec Sharing and recording knowledge: a case study in Knowtec (Brazil). *Revista ACB*, v. 20, n. 1, p. 98-111, 2015.

ERNST, Neil A. On the role of requirements in understanding and managing technical debt. In: 2012 Third International Workshop on Managing Technical Debt (MTD). IEEE, 2012. p. 61-64.

ERNST, Neil A. et al. Agile requirements evolution via paraconsistent reasoning. In: International Conference on Advanced Information Systems Engineering. Springer, Berlin, Heidelberg, 2012. p. 382-397.

ERNST, Neil A. et al. Measure it? manage it? ignore it? software practitioners and technical debt. In: Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering. 2015. p. 50-60.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. From data mining to knowledge discovery in databases. *AI magazine*, v. 17, n. 3, p. 37-37, 1996.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, v. 39, n. 11, p. 27-34, 1996.

GHOSE, Aditya. Formal methods for requirements engineering. In: Proceedings International Symposium on Multimedia Software Engineering. IEEE, 2000. p. 13-13.

HUSSAIN, Azham; MKPOJIOGU, Emmanuel OC; KAMAL, Fazillah Mohmad. The role of requirements in the success or failure of software projects. *International Review of Management and Marketing*, v. 6, n. 7, p. 306-311, 2016.

HUSSAIN, Azham; MKPOJIOGU, Emmanuel OC. Requirements: Towards an understanding on why software projects fail. In: AIP Conference Proceedings. AIP Publishing LLC, 2016. p. 020046.

KRUCHTEN, Philippe; NORD, Robert L.; OZKAYA, Ipek. Technical debt: From metaphor to theory and practice. *Ieee software*, v. 29, n. 6, p. 18-21, 2012.

LENARDUZZI, Valentina; FUCCI, Davide. Towards a holistic definition of requirements debt. In: 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). IEEE, 2019. p. 1-5.

LI, Zengyang; AVGERIOU, Paris; LIANG, Peng. A systematic mapping study on technical debt and its management. *Journal of Systems and Software*, v. 101, p. 193-220, 2015.

M. Fowler. Technical debt quadrant, 2009;
<http://martinfowler.com/bliki/TechnicalDebtQuadrant.html>

MAGALHÃES, Thiago Gonçalves; DALMAU, Marcos Baptista Lopez; DE SOUZA, Irineu Manoel. Gestão do conhecimento para tomada de decisão: um estudo de caso na empresa júnior. *Revista Gestão Universitária na América Latina-GUAL*, v. 7, n. 2, p. 108-129, 2014

MCMANUS12, John; WOOD-HARPER, Trevor. Understanding the sources of information systems project failure. 2007.

MELO, Ana et al. Identification and Measurement of Technical Debt Requirements in Software Development: A Systematic Literature Review. *arXiv preprint arXiv:2105.14232*, 2021.

MOHER, D. et al. Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) *BMJ* 2015;
<https://www.bmj.com/content/bmj/349/bmj.g7647.full.pdf>

S. McConnell, Managing Technical debt, 2008;
<http://www.construx.com/uploadedfiles/resources/whitepapers/Managing%20Technical%20Debt.pdf>

SAVIOTTI, Pier Paolo. Knowledge networks: structure and dynamics. In: *Innovation Networks*. Springer, Berlin, Heidelberg, 2009. p. 19-41.

SEAMAN, Carolyn; GUO, Yuepu. Measuring and monitoring technical debt. In: *Advances in Computers*. Elsevier, 2011. p. 25-46.

SYMONDS, Andrew J. Creating a software-engineering knowledge base. *IEEE Software*, v. 5, n. 2, p. 50-56, 1988.

STONER, James A. F. e FREEMAN, R. Edward. Administração. Tradução de Alves Calado 5ª ed., Rio de Janeiro: Livros Técnicos e Científicos, 1999.

TODD, Ross J.; PARKER, Joan; YERBURY, Hilary. Knowledge representation and multimedia knowledge base design: A methodology for alignment. In: Proceedings of the 2nd International Interactive Multimedia Symposium 1994. 1994. p. 543-548.

TODD, Ross J. From Information to Knowledge: Charting and Measuring Changes in Students' Knowledge of a Curriculum Topic. Information Research: An International Electronic Journal, v. 11, n. 4, p. n4, 2006.

USAMA, M. Advances in knowledge discovery and data mining. 1996.

W. Cunningham, "The WyCash Portfolio Management System," Proc. OOPSLA, ACM, 1992; <http://c2.com/doc/oopsla92.html>.