

How Far From the Metaphor? The Diverse Understandings of Technical Debt in GitHub Issues

Iury Cavalcante*

PPGI - Programa de Pós-Graduação em Informática
Universidade Federal do Estado do Rio de Janeiro
Rio de Janeiro, Brazil
iury.cavalcante@gmail.com

Paulo Sergio Medeiros dos Santos

PPGI - Programa de Pós-Graduação em Informática
Universidade Federal do Estado do Rio de Janeiro
Rio de Janeiro, Brazil
pasemes@uniriotec.br

ABSTRACT

This study investigates how the metaphor of technical debt (TD) is used in practice to communicate complex aspects of software development, particularly in relation to quality-delivery trade-offs. The study addresses the problem of inadequate or inconsistent application of the TD concept, sometimes compounded by linguistic, cultural, and organizational factors that hinder effective communication and compromise project sustainability. A qualitative analysis of issues from GitHub repositories was conducted using the Qualitative Data Analysis method. The findings revealed four main categories of TD interpretation: Decision, Activity, Problem and Pending Task. These categories encompass eight usage contexts. Many of the observed interpretations diverge significantly from the original metaphor, reinforcing the need for clearer conceptual alignment to support technical debt management practices.

KEYWORDS

Technical Debt, Technical Communication, Software Engineering, Software Project, Qualitative Data Analysis

1 Introdução

O conceito de dívida técnica (DT) emergiu como uma abstração destinada a simplificar a comunicação sobre questões técnicas complexas [15]. Seu objetivo é criar uma linguagem acessível que permita a interação entre pessoas com diferentes papéis e níveis de conhecimento técnico, facilitando o entendimento e a colaboração em torno desses desafios.

Embora o termo "dívida técnica" tenha sido originalmente proposto como uma metáfora para comunicar de forma simples as necessidades técnicas do software visando equilibrar decisões de curto prazo com a sustentabilidade a longo prazo [8], este conceito tem sido frequentemente mal interpretado e subutilizado na prática [13], o que limita o pleno aproveitamento do arcabouço teórico construído em torno da DT.

Este estudo tem como objetivo explorar, por meio de uma análise qualitativa das *issues* em português, disponíveis no GitHub¹, como o termo DT está sendo apropriado e utilizado na prática. Busca-se identificar interpretações, aplicações e adaptações do conceito no contexto do desenvolvimento de software. Ao investigar esse uso prático, o estudo contribui ao apontar uma diversidade de entendimentos diretamente relacionados aos desafios e limitações enfrentados em uma das atividades apontadas como mais relevantes do gerenciamento da dívida técnica (GDT): a comunicação [31].

Garantir uma boa comunicação sobre a DT é imperativo, pois os debates sobre a qualidade do software extrapolam as cercanias da

engenharia. O software evoluiu de um papel coadjuvante, quando apenas apoiava processos empresariais, para uma posição de destaque ao começar a gerar receitas como produto ou serviço [30]. Atualmente, começa a despontar como o ator principal nas empresas que estruturam seus modelos de negócio em torno dele. Hoje, o software pode ser um produto, um serviço ou até mesmo o cerne da empresa, sobre a qual todas as atividades e operações são organizadas [24]. Garantir sua qualidade e a sustentabilidade, assegurando que a dívida técnica seja gerida de forma assertiva, será cada vez mais relevante.

O restante do artigo está estruturado da seguinte forma: na próxima seção vamos explorar os significados e o contexto cultural em torno da DT. Em seguida, na seção 3, apresentamos o método utilizado no estudo. A seção 4 apresenta os resultados obtidos a partir das análises realizadas. Na seção 5, discutimos as implicações desses achados e, por fim, as seções 6 e 7 trazem, respectivamente, as ameaças à validade e concluem o trabalho destacando as limitações do estudo e sugerindo direções para pesquisas futuras.

2 Fundamentação

Como este trabalho abordará aspectos relacionados à comunicação, é pertinente iniciar com uma breve contextualização sobre um ponto relevante: a natureza conceitual do termo "Dívida Técnica". Gramaticalmente, a expressão é formada pelo substantivo "dívida" e qualificado pelo adjetivo "técnica" que informa o tipo de dívida em questão. Contudo, na literatura e na prática da engenharia de software, a DT se consolidou como um termo técnico composto amplamente difundido e reconhecido como uma entidade conceitual própria.

Embora o termo esteja amplamente disseminado, as múltiplas possibilidades de interpretação, decorrentes tanto de aspectos culturais quanto da própria construção terminológica, podem influenciar diretamente a forma como é compreendido, discutido e tratado.

2.1 Dívida ou Débito?

É compreensível que o termo em inglês *debt* seja, por vezes, traduzido de forma automática para a palavra "débito", devido à semelhança gráfica. Assim como nesse caso, há diversos falsos cognatos que, embora semelhantes na forma, possuem significados distintos em português e inglês [1]. Vale destacar que existe, em inglês, um termo específico para "débito": *debit*, com grafia bastante próxima, mas cujo significado é distinto. Portanto, se a intenção original da metáfora de Cunningham [8] fosse tratar de "débito técnico", ele poderia ter empregado diretamente o termo *debit*.

À primeira vista, essa discussão pode parecer uma questão meramente linguística e secundária. No entanto, ela carrega implicações

¹<https://github.com>

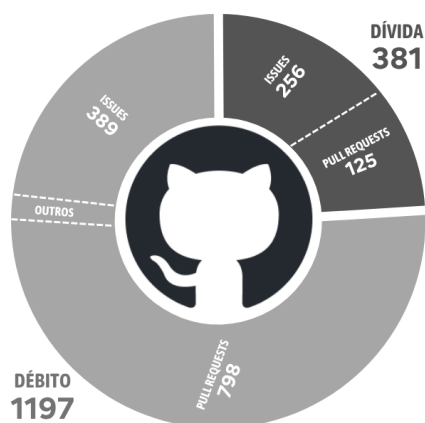


Figura 1: "Dívida Técnica" vs "Débito Técnico" no GitHub

conceituais relevantes para a compreensão do arcabouço que se formou ao redor do tema. Ao reconhecer que a metáfora faz referência a uma "dívida" (*debt*), e não a um simples "débito" (*debit*), é possível perceber nuances adicionais que enriquecem o entendimento da metáfora original, fundamentada no universo financeiro.

Nesse contexto, elementos associados a dívidas financeiras, como o conceito de juros (*interest*), ganham importância na análise da DT. O acúmulo de juros é, inclusive, defendido como um aspecto determinante para caracterizar a DT [3]. Da mesma forma, o termo *repayment*, utilizado para nomear uma das atividades de GDT [16] e frequentemente traduzido ao português como "pagamento". No entanto, na língua inglesa, *repayment* refere-se especificamente ao pagamento ou amortização de dívidas, diferentemente de *payment*, um termo mais genérico. Em algumas traduções do português para o inglês, observa-se o uso de *payment*, o que pode resultar em uma leve perda do sentido original associado à metáfora de dívida financeira.

A busca pelos termos exatos, "débito técnico" e "dívida técnica", na base GitHub apresenta o resultado três vezes maior para débito em relação à dívida, conforme observado na Figura 1. Esse é mais um indicativo de que o esclarecimento conceitual pode ser necessário.

2.2 O Nome é Parte do Problema?

Existe uma forte conotação negativa associada ao conceito de "dívida", frequentemente entendido como algo a ser evitado e que representa descontrole financeiro, inadimplência ou dificuldades econômicas. Somos socializados com a ideia de que contrair dívidas é indesejável e sinaliza problemas de gestão financeira pessoal [18] ou organizacional [10].

Além desse fator cultural, os tomadores de decisão não estão inclinados a assumir a responsabilidade por um problema denominado "técnico". Eles preferem se sentir como vítimas de um problema de tecnologia [3]. O próprio uso do adjetivo "técnico" pode atuar como um gatilho organizacional que desloca a responsabilidade para as equipes técnicas, criando barreiras na divisão de responsabilidades com as áreas de negócios ou de gestão. Compreender adequadamente o conceito de DT permite reduzir esses efeitos, favorecendo uma atuação mais integrada entre todas as áreas.

A combinação das palavras "dívida" e "técnica" parece explosiva, o que torna a atividade de comunicação no GDT ainda mais desafiadora, gerando resistências tanto por parte dos tomadores de decisão quanto dentro das próprias equipes técnicas [26]. Além disso, a natureza frequentemente invisível da DT para os stakeholders não técnicos torna a comunicação sobre o valor de negócio associado ao seu tratamento um desafio central [13]. A comunicação no GDT e um melhor entendimento do conceito se destacam como um dos aspectos de preocupação frequentemente mencionados pelos profissionais envolvidos em estudos [28, 31]. Desmistificar esse vocabulário e alinhá-lo ao contexto de negócios é uma forma de apoiar as práticas de GDT.

2.3 Muitas Definições e uma Direção

O tema DT foi abordado e amplamente difundido por diversos estudos. Uma busca rápida, realizada em maio de 2025, na base da ACM Digital Library² pelo termo "technical debt" retorna mais de 400 artigos com esse termo em seus títulos. Ao restringirmos o parâmetro de busca ao *abstract*, obtivemos quase 700 resultados. E quando a busca é feita por qualquer parte do artigo, encontramos mais de 2.500 resultados.

Esse grande volume de artigos favoreceu a elaboração de mapeamentos e revisões sistemáticas. De maneira geral boa parte destes estudos secundários demonstram de alguma forma, mesmo não sendo o objetivo principal deles, um certo grau de preocupação com classificação, taxonomia ou padronização da DT. Esses estudos fornecem uma base sólida para entender as dificuldades com várias interpretações relacionadas à ela ao longo do tempo.

Tom, Aurum e Vidgen [29], 2013, discutiram as várias interpretações e nomenclaturas encontradas na literatura, destacando a necessidade de uma taxonomia consolidada para melhorar a comunicação e o entendimento do conceito. O artigo reforça a importância de uma abordagem padronizada para a classificação da DT.

Li, Avgeriou e Liang [16], 2015, realizaram um mapeamento sistemático abrangente sobre a DT e GDT. Este artigo menciona a falta de consenso em nomenclaturas e definições, apontando as diferentes classificações utilizadas na literatura. A diversidade terminológica contribui para a confusão e a dificuldade no gerenciamento eficaz da DT.

Alves et al. [2], 2016, identificaram e discutiram a variedade de termos e classificações usados para descrever a DT. Este estudo sugere que a ausência de uma taxonomia clara pode levar a mal-entendidos e dificuldades no GDT. Ele também enfatiza a necessidade de uma terminologia padronizada para melhorar a clareza e a eficácia na comunicação sobre DT.

Rios et al. [25], 2018, conduziram um estudo terciário que destacou nos resultados que a definição é um dos tópicos mais considerados por outros pesquisadores. Destacando que o refinamento do conceito de DT é importante para transformar a forma como as atividades são realizadas na prática.

Junior e Travassos [14], 2021, também destacam a existência de uma sobrecarga interpretativa em torno da definição de DT, observando que, na prática, o termo tem sido usado para descrever praticamente qualquer tipo de problema ocorrido durante o

²<https://dl.acm.org>

desenvolvimento de software. Essa imprecisão pode levar a inconsistências conceituais e comprometer tanto a comunicação entre os envolvidos quanto a efetividade das ações de gerenciamento.

Averigou et al. [3], 2023, foram enfáticos, ao realizar uma grande revisão histórica, se tudo o que é prejudicial ao desenvolvimento de software for rotulado como DT perderemos seu uso prático, já que não seria mais possível identificá-la, medi-la ou corrigi-la concretamente.

Esses estudos têm em comum uma preocupação com a organização e classificação da informação, uma vez que interpretações equivocadas podem confundir a comunicação, dificultar o gerenciamento e a implementação de estratégias relacionadas à DT. Em 2025, desponta uma iniciativa que pretende direcionar os debates a respeito da DT, o "Manifesto para Reenquadramento da Dívida Técnica" [4], nele estão descritas crenças, princípios e cinco valores mais importantes. E um desses valores está diretamente associados ao entendimento e comunicação clara da DT:

"V2 - Valorizamos uma comunicação simples, prática e orientada por valor sobre a Dívida Técnica para todas as partes interessadas, em vez de apresentar detalhes excessivos, minuciosos e opressivos"(tradução livre)

3 Método de Pesquisa

A pesquisa foi conduzida por meio de análise documental de natureza qualitativa [21], com abordagem indutiva, aplicada a *issues* de projetos hospedados em repositórios do GitHub. Esse *corpus* textual constitui uma fonte documental espontânea, produzida no contexto real do desenvolvimento de software. A análise documental mostrou-se especialmente adequada, pois permite compreender o fenômeno investigado a partir da própria expressão dos indivíduos envolvidos. É justamente essa manifestação espontânea sobre DT que o estudo busca examinar, uma vez que outros métodos de coleta podem inibir a naturalidade das declarações [22].

A suposição geral deste estudo é que a aplicação inadequada ou a interpretação limitada do conceito original de DT impacta negativamente a sustentabilidade dos softwares. Quando os envolvidos em projetos de desenvolvimento não compreendem plenamente o conceito, há o risco de aplicações de técnicas equivocadas, o que compromete a eficácia da GDT e os benefícios associados à sua adoção [16]. Diversos estudos apontam a existência de mal-entendidos e interpretações divergentes sobre o que constitui a DT [3, 17, 28], mas as questões de pesquisa desse estudo concentraram-se em compreender: (QP1) como o conceito tem sido apropriado nas discussões? E (QP2) como essas distorções surgem ao longo do ciclo de desenvolvimento do software? Entender essa dinâmica é essencial para que se possam propor ações específicas de conscientização, formação ou ajuste de práticas.

Um resumo visual do processo de busca, seleção e análise dos dados extraídos do GitHub pode ser observado na Figura 2.

3.1 Busca dos Dados

O GitHub foi selecionado como fonte de dados devido à sua relevância como plataforma de gestão de projetos de software e pelo acesso aberto às informações dos repositórios. As *issues* foram utilizadas como base para compreender como as equipes de desenvolvimento se comunicam e interpretam a DT ao longo do ciclo de vida do



Figura 2: Processo de seleção e análise dos dados do GitHub

software. Optou-se por trabalhar exclusivamente com *issues*, em vez de *commits* ou *pull requests*, devido ao seu nível de detalhamento, interatividade e organização. As *issues* geralmente contêm descrições detalhadas e discussões entre desenvolvedores, contextualizando problemas e soluções propostas. Além disso, as interações nas *issues* fornecem uma visão rica sobre as percepções e comunicações sobre a DT. *Labels* associadas auxiliam na organização e identificação de tópicos relevantes e, diferentemente de *commits* ou *pull requests*, as *issues* abordam aspectos mais amplos relacionados ao desenvolvimento e tomada de decisões.

A coleta de dados começou com uma inspeção preliminar dos repositórios e das informações disponíveis, com o objetivo de compreender a estrutura das *issues* e identificar padrões nas descrições e informações associadas, como *labels* e comentários. Esse processo foi fundamental para a construção de uma estratégia de busca mais direcionada e eficiente. Inicialmente, foram testadas diferentes combinações de termos relacionados, como "dívida", "dívida", "técnica" e "tecnic". Essas consultas, realizadas tanto com o operador "+" quanto de forma independente, resultaram em uma quantidade significativa de dados irrelevantes e em idiomas distintos do português. A string de busca "dívida técnica" foi consolidada como a mais representativa para a pesquisa após essa exploração inicial dos dados. As buscas foram realizadas diretamente nas *issues* do GitHub, que realiza uma varredura nos campos de título, descrição, rótulos (*tags*) e comentários vinculados. Essa foi a abordagem de busca adotada neste estudo.

Os dados foram extraídos de forma manual e incluíram informações como título, *link*, descrição, *labels*, data de criação, situação, nome do projeto e os comentários associados. Esse conjunto de informações forneceu uma base rica para a análise qualitativa, composta por 256 *issues* criadas entre março de 2015 e maio de 2025, provenientes de 88 repositórios diferentes. Cabe ressaltar que não foram aplicados filtros de data, repositório ou qualquer outro tipo de restrição aos resultados obtidos.

3.2 Seleção e Análise dos Dados

O processo de análise foi fundamentado nas três etapas propostas por Glenn Bowen [6]: leitura superficial, leitura completa e interpretação. Adaptado em nosso estudo para leitura exploratória, leitura analítica e um aprofundamento da leitura com intuito de formar uma categorização. Entre cada uma dessas etapas, adotou-se um procedimento de verificação cruzada entre pesquisadores, no qual um realizava a ação, seja a aplicação dos critérios de exclusão ou a codificação, e outro revisava os resultados. Em casos de divergência,

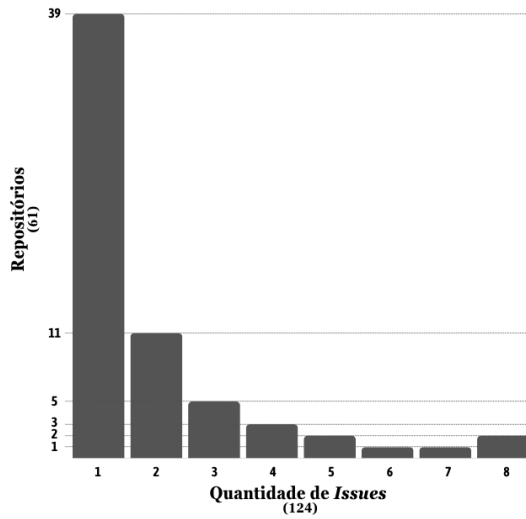


Figura 3: Distribuição de *issues* por repositório

era promovido um debate para busca de consenso. Quando não havia concordância, o código atribuído era descartado ou a *issue* era marcada como insuficiente, preservando-se o dado original sem intervenção.

Esse estudo concentra-se em analisar como o termo tem sido efetivamente utilizado por grupos que utilizam a língua portuguesa. Apesar da string de busca conseguir fazer um boa seleção de projetos brasileiros, foram adotados três critérios de exclusão: (i) *issues* que não foram escritas em português, a fim de minimizar possíveis ruídos linguísticos na interpretação dos dados; (ii) foram consideradas as *issues* duplicadas ou com conteúdo semanticamente redundante, desde que estivessem no projeto; e (iii) *issues* com descrições insuficientes para interpretar adequadamente as percepções relacionadas à DT foram descartadas. Durante a leitura exploratória, foram identificados repositórios que não estavam relacionados diretamente ao desenvolvimento de software, como guias de boas práticas e descrições de vagas. Ainda assim, optou-se por mantê-los no *corpus*, uma vez que contribuem para o objetivo do estudo de compreender como os praticantes têm utilizado o termo DT. Após essa etapa inicial, permaneceram no *corpus* 124 *issues*, distribuídas em 61 repositórios, conforme ilustrado na Figura 3. Essa quantidade de fontes afasta a preocupação com uma possível concentração de *issues* de um mesmo grupo, que poderia compartilhar percepções semelhantes sobre a DT, e reflete a diversidade resultante da seleção de *issues* provenientes de múltiplos repositórios, contextos e perfis de contribuidores.

Os dados coletados foram analisados por meio de uma abordagem qualitativa indutiva, composta por múltiplas leituras sequenciais e aplicação das técnicas de *jotting* e *memoing* reconhecidas na literatura como estratégias eficazes para registro de *insights* emergentes, auxiliando na construção gradual das categorias analíticas [21]. No contexto desse estudo, a saturação foi determinada quando as discussões analisadas começaram a apresentar padrões repetidos e os

temas emergentes se tornaram consistentes entre os dados coletados. Neste momento, analisar mais *issues* não adicionou novas interpretações ou perspectivas relevantes. O critério de saturação conceitual garantiu que o volume de dados analisados fosse suficiente para embasar as conclusões [7]. Segue o detalhamento das etapas:

- (1) **Preparação dos dados:** As *issues* coletadas foram exportadas do GitHub para uma planilha, onde os dados foram organizados e ordenados do mais recente para o mais antigo. Optou-se por iniciar a leitura dos dados mais recentes, de modo que, ao atingir o ponto de saturação, a análise já tivesse contemplado prioritariamente as *issues* mais atuais. Todas as *issues* foram identificadas por códigos numéricos sequenciais (Issue ID) com o objetivo de garantir a rastreabilidade das evidências empíricas sem expor diretamente informações sensíveis dos projetos.
- (2) **Leitura exploratória:** Realizou-se o ciclo de leituras rápida de todas as 256 *issues* com aplicação da técnica de *jotting*, na qual foram registradas anotações preliminares de observações iniciais, potenciais significados atribuídos ao termo DT e possíveis pontos de interesse.
- (3) **Aplicação dos critérios de exclusão e duplicação:** Foram aplicados os critérios previamente definidos para exclusão de duplicadas ou semanticamente redundantes que eliminaram da leitura analítica 70 *issues*, bem como a eliminação de 61 *issues* com ausência de conteúdo suficiente para análise qualitativa. Encontramos na amostra apenas uma *issue* que utilizava língua estrangeira.
- (4) **Leitura analítica:** Realizamos o segundo ciclo de leituras, desta vez de forma mais aprofundada, em cima das 124 *issues* restantes. Aplicou-se a técnica de *memoing*, na qual foram registrados memos analíticos com observações conceituais emergentes. Durante essa etapa, foram identificadas categorias preliminares de significados.
- (5) **Definição do ponto de saturação:** A análise foi conduzida até o ponto em que não surgiram novas categorias ou interpretações relevantes, caracterizando a saturação conceitual do *corpus*. Essa decisão foi tomada após a leitura aprofundada de 73 *issues* e, para confirmação, outras 20 foram analisadas sem que a situação se alterasse.
- (6) **Agrupamento e refinamento das categorias:** Após alcançar a saturação, retornou-se ao início da tabela de dados para revisar, consolidar e reagrupar os significados identificados, estruturando a categorização final.

4 Resultados

A análise permitiu identificar 17 percepções ou intenções manifestadas pelos participantes ao utilizar o termo DT, indicando os diferentes sentidos atribuídos ao conceito nas *issues* dos projetos. Com base nessas percepções ou intenções, foi possível identificar um conjunto de oito contextos de uso, que representam situações concretas nas quais a DT foi mencionada durante o desenvolvimento de software. Estes contextos de uso estão associados à QP2.

Os contextos de uso identificados foram organizados por afinidade, a partir dos quais emergiram indutivamente quatro categorias:

Decisão, Atividade, Problema e Pendência. Essas categorias respondem a QP1, pois ajudam a compreender os diferentes significados de como o conceito da DT tem sido utilizado na prática.

A Figura 4a sintetiza as distintas formas de apropriação do conceito observadas nas *issues*. Ela apresenta da esquerda para a direita as 17 percepções ou intenções identificadas, os oito contextos de uso e as quatro categorias. Compondo toda diversidade de entendimentos encontrada na base analisada. Nas próximas seções apresentaremos os achados em detalhes por meio dos exemplos mais representativos encontrados nas *issues*. Cada seção representa uma categoria, as subseções são os contextos de uso com seus respectivos exemplos práticos de textos extraídos. Os achados foram ordenados, do menor para o maior, conforme seu volume de observações.

4.1 Percepção Menos Observada: Decisão

A categoria "Decisão" reúne situações em que a identificação da DT decorre de uma escolha deliberada para viabilizar entregas ou contornar limitações temporárias do projeto. Nessas situações, o conceito de DT é utilizado explicitamente como uma forma de registrar concessões técnicas feitas no planejamento ou execução. Como ilustra o seguinte trecho:

"No momento não temos informações sobre quais usuários são administradores ou não, porém podemos deixar isso para a próxima sprint. Pode fazer esse controle diretamente no controller da página, marcando um usuário como admin se o idUFFS dele estiver dentro de uma lista de idsUFFS hardcoded no próprio controller. Deixamos isso como dívida técnica para podermos entregar o site nessa sprint." (Issue ID: 119)

Este exemplo evidencia o uso do conceito de DT como uma justificativa consciente para adiar a implementação completa de uma funcionalidade, priorizando a entrega dentro do prazo estabelecido da sprint.

4.2 Percepção da Dívida Técnica como uma Atividade

Os dados analisados evidenciaram o uso comum da DT para referir-se a ações técnicas, planejadas ou não, durante o ciclo de desenvolvimento de software que visam manter sua qualidade, como, testes, documentação, refatoração e atualizações tecnológicas de maneira geral.

4.2.1 Atividades que representam uma ação de melhoria. Grande parte das *issues* analisadas estavam dentro do contexto de uso denominado Ação de Melhoria. Este contexto corresponde a atividades técnicas orientadas a aprimorar a qualidade estrutural do sistema, corrigir aspectos técnicos não ideais ou otimizar funcionalidades já existentes. São ações que podem ser associadas ao pagamento de DT, mas que nem sempre fazem parte de um plano formalizado como no GDT.

Neste contexto, a percepção identificada estava ligada à busca por eficiência, otimização e qualidade contínua, sendo muitas vezes fruto de inspeções técnicas rotineiras e iniciativas proativas de melhoria do código:

"Mapear as dívidas técnicas, refatorando o código conforme necessário para melhorar clareza e extensibilidade." (Issue ID: 234)

"Os critérios de aceitação foram feitos com exceção do campo de nome que será transferido para uma US de perfil. Não foram produzidos testes, mas estes serão dívidas técnicas transferidas para um TS." (Issue ID: 27)

4.2.2 Lista de atividades que precisam ser executadas. Embora com menor frequência, algumas *issues* foram classificadas sob o contexto de uso denominado Registro de Atividades Técnicas. Este contexto refere-se ao comportamento de listar um conjunto de itens técnicos necessários que representam trabalho ainda não realizado, funcionando como uma espécie de planejamento das tarefas a serem executadas. Não foi identificada uma associação clara com as características de adiamento relacionadas ao *trade-off* clássico do conceito de DT. As listas podem incluir tarefas de documentação, testes, revisões técnicas, ajustes em artefatos ou verificações. Isso é exemplificado em:

Issue 220 — Dívida técnica da tela da home

"Fazer verificação para ficar em full width no mobile, como a tela do perfil."

4.3 Percepção como Sinônimo de Problema

A apropriação do conceito da DT como sinônimo de problemas técnicos apareceu em quantidade muito maiores do que a conotação de uma atividade que precisa ser feita, detalhada na subseção anterior. Isso pode ser explicado, em parte, por estudos que classificam *bugs* ou erros como um tipo de DT [16][27], apesar de não existir um consenso sobre essa classificação, elas foram observadas e constituem a base analisada. As *issues* associadas com esse contexto de uso geralmente descrevem a identificação de erros ou restrições tecnológicas. Também estavam associados a tarefas consideradas concluídas que necessitavam de retrabalho. O uso do termo nesse contexto reflete uma compreensão prática da DT como algo que precisa ser resolvido ou evitado para não gerar impactos negativos.

A análise revelou três contextos de uso que apresentam a DT como um problema. São elas Não conformidades em avaliações, Correção de falhas e Ameaça a ser evitada todas detalhadas a seguir.

4.3.1 Não conformidades em avaliações. Este recorte aborda situações em que a DT foi associada à identificação de problemas durante revisões formais, inspeções e execução de testes. As *issues* analisadas neste contexto geralmente relatam a detecção de ajustes necessários em funcionalidades, configurações e lacunas de documentação. Essas situações emergem como resultado direto de processos estruturados de avaliação, como *checklists*, planos de teste e critérios de aceitação.

Trata-se de um uso da DT como sinônimo de problema técnico identificado em revisões, refletindo a percepção de não conformidade diante de expectativas previamente estabelecidas. Como exemplo, em uma das *issues* analisadas:

Issue 224 — Dívida técnica de refatorar a tela de editar perfil

"A seguinte tela não está responsiva e está desalinhada."

4.3.2 Correção de falhas. Pode-se observar *issues* que designavam os problemas que surgiram durante a execução do sistema em ambiente de produção como uma DT. Esses problemas estavam frequentemente relacionados a erros encontrados durante o uso do sistema e reportados por usuários:

Issue 229 — Dívida técnica: ajustar queries de find para Aluno, Disciplina, Nota, Turma

"Foi corrigido o erro no relacionamento das tabelas aluno/disciplina/nota/turma, com adição de view para filtragem de dados e performance e alteração no retorno dos endpoints."

4.3.3 Ameaça a ser evitada. Em menor frequência, a DT foi utilizada como uma abordagem preventiva para alertar sobre práticas que poderiam gerar problemas futuros. Esse foi o único contexto de uso encontrado na base analisada que rotula a DT de forma negativa antes do fato gerador. Ele foi observado, principalmente, em *issues* relacionadas a recomendações de boas práticas e descrições de vagas de emprego, onde era solicitado o desenvolvimento de código sem a geração de DT. Exemplos dessas situações podem ser observadas nas descrições a seguir:

"[...] A padronização e boas práticas tornam o código mais organizado, facilitando a colaboração, a manutenção e a escalabilidade, além de prevenir erros e evitar dívidas técnicas."(Issue ID: 248)

Issue 247 — [Remoto] Product Engineer (Next.js) na [...]

"[...] Responsabilidades Participar ativamente na criação, desenvolvimento e manutenção da nossa plataforma multi-produto, sempre tomando os cuidados necessários para não gerar dívidas técnicas; [...]"

4.4 Percepção mais Observada: Algo Que Não Foi Concluído ou Está Pendente

A análise documental revelou que a DT foi frequentemente utilizada para informar algo que não foi concluído. Esse uso reflete uma percepção prática da DT como resultado de falhas no cumprimento de prazos, planejamento inadequado ou replanejamento devido a limitações de tempo, recursos ou critérios de aceitação mal definidos. As *issues* associadas a essa categoria destacaram situações em que tarefa programadas não foram concluídas e quase sempre eram postergadas para a próxima entrega. Foi possível identificar padrões consistentes relacionados ao uso da DT como pendência, agrupados em dois contextos de uso: Tarefa não concluída; e Demanda Emergente.

4.4.1 Não entrega do escopo planejado. De todas as observações realizadas durante a análise, essa foi a percepção mais frequentemente associada à DT. As *issues* com tarefas que estavam originalmente planejadas e agendadas, mas que não foram concluídas devido à falta de tempo, recursos ou outros fatores, eram replanejadas e marcadas como se fossem uma dívida que seria concluída no futuro.

"[...] Essa US foi alocada a princípio na Sprint 3, e tornou-se dívida técnica para a Sprint 4."(Issue ID: 38)

Issue 167 — Adicionar Footer Front-end e desenhar design da Busca Avançada

Autor: "Objetivo

Adicionar Footer na página de timeline

Desenhar design da Busca Avançada".

Colaborador 1: "Pessoal, essa issue é da release 3 mesmo ou é dessa de agora (4)?"

Colaborador 2: "É da release 3, entrou como dívida técnica, fecharemos hoje."

4.4.2 Demandas que emergiram durante a execução. A análise revelou que a DT também foi utilizada para descrever como pendência, algo novo que surgiu a partir de necessidades detectadas no decorrer da execução de uma tarefa de codificação ou teste. Essas novas necessidades, sem histórico prévio ou planejamento inicial, foram adicionadas ao escopo do projeto. Esse contexto de uso reflete uma prática de incorporar novas demandas ao *backlog* do projeto a título de DT.

Issue 156 — Linha temporal do ato do processo licitatório

Autor: "Descrição

Elaborar uma linha temporal do ato do processo licitatório, desde a fase inicial e final de uma licitação[...]

Colaborador 1: "Dívida Técnica: Procurar quais são os tipos de atos

- Abertura

- Aviso de Vencedor

- Outro [...]"

5 Discussão

Os achados revelam que o uso prático do conceito de DT frequentemente se distancia de sua definição original, assumindo significados diversos de acordo com o contexto. Essa variedade de interpretações podem dificultar a comunicação, pois a percepção da DT e o seu gerenciamento, ainda são incipientes [28]. Nesta seção, discutimos esses achados à luz da literatura existente, buscando compreender suas implicações teóricas e práticas.

5.1 Achados versus Teoria

A seguir, os achados são examinados considerando os diferentes conceitos que orientam a identificação da dívida técnica quanto à sua intencionalidade [19], visibilidade [15] e origem [12]. A seleção desses classificadores não busca exaurir todas as possíveis categorizações da DT, mas representam um recorte teórico capaz de discutir os achados sob diferentes aspectos do fenômeno. A Figura 4b também sintetiza esse alinhamento dos achados com a teoria.

A distinção entre dívida técnica intencional e não intencional, proposta por McConnell [19], oferece uma possível perspectiva para analisar os resultados deste estudo. As ocorrências em que a DT é descrita como uma decisão consciente para viabilizar entregas refletem diretamente a noção de dívida intencional, assumida com plena consciência e expectativa de resolução futura. Já as atividades de melhoria, embora frequentemente rotuladas como DT, estão mais associadas às atividades do GDT, como por exemplo o pagamento da dívida. Estando, portanto, apenas indiretamente

Formas de apropriação da DT (4a)				Alinhamento com a literatura (4b)			
Percepções		Contextos de uso	Categorias	Intencionalidade [19]	Visibilidade [15]	Origem [12]	Metáfora original [8]
Alinhado	Adiantar entregas	TOMADA DE DECISÃO	DECISÃO	✓	✓	✓	✓
	Postergar solução						
Próximo	Erro	CORREÇÃO DE FALHAS	PROBLEMA	✓	✓		
	Correção						
	Defeito						
Distante	Diferente do combinado	NÃO CONFORMIDADE		✓	✓		
	Receio						
	Medo	AMEAÇA A SER EVITADA					
	Prevenção						
	Cautela	AÇÃO DE MELHORIA	ATIVIDADE				
	Qualidade Contínua						
	Otimização	TAREFA TÉCNICA					
	Planejamento						
	Todoist	TAREFA EMERGENTE					
	Alteração de escopo						
	Nova necessidade	TAREFA NÃO CONCLUÍDA	PENDÊNCIA				
	Protelação						

Figura 4: Diversidade de entendimentos e alinhamento com a literatura

relacionadas ao conceito. As *issues* que tratam DT como *bugs*, falhas ou erros indicam uma dívida não intencional, surgida sem planejamento ou consciência técnica. Em contrapartida, casos em que a DT é mencionada como algo a ser evitado demonstram uma postura consciente, porém negativa, sugerindo um uso normativo e preventivo do termo. Por fim, as interpretações que associam DT a pendências ou tarefas não concluídas que não se enquadram diretamente nessa tipologia, revelando uma apropriação sem base conceitual clara.

A proposta de organização do “panorama da dívida técnica” (*Technical Debt Landscape*) propõe um avanço conceitual importante na tentativa de associar o uso do termo DT aos elementos invisíveis do sistema e que geralmente só são perceptíveis aos desenvolvedores [15]. Ao confrontar essa abordagem com os achados deste estudo, observa-se que a categoria “Decisão” é a que mais se aproxima do entendimento proposto ao tratar a dívida como uma concessão técnica invisível. Em contrapartida, as categorias “Problema” e “Pendência” revelam uma apropriação que extrapola os limites do que deveria ser considerado dívida técnica sob a ótica do *Technical Debt Landscape*: erros visíveis, falhas operacionais e atrasos de entrega são exemplos de elementos fora do retângulo conceitual proposto pelos autores, pois representam problemas ou lacunas explícitas, e não dívidas invisíveis e estruturais. A categoria “Atividade”, por sua vez, reforça a ambiguidade do uso prático do termo, ao incluir ações de melhoria contínua ou documentação como se fossem DT, quando na verdade se tratam de ações que até podem estar atuando sobre a dívida real, mas não são ela em si.

Outra maneira de tentar entender esses achados é por meio da estrutura conceitual do *Technical Debt Quadrant* que considera a origem da DT em prudente ou imprudente e deliberada ou inadvertida [12]. A categoria “Decisão”, identificada neste estudo, encaixa-se com clareza no quadrante da dívida prudente e deliberada, em que

há *trade-offs* conscientes visando acelerar entregas, com intenção de resolver a limitação posteriormente. Por outro lado, a categoria “Atividade”, ao representar ações de melhoria contínua, se aproxima do quadrante prudente e inadvertida, porém os achados não revelaram um processo de amadurecimento técnico como sinalizado por Fowler. Já as categorias “Problema” e, principalmente, “Pendência”, fogem da estrutura proposta pelo autor. Tarefas não concluídas por falta de tempo, atrasos, mudanças de escopo, erros ou *bugs* não constituem DT, pois não resultam de decisões de design (deliberada) nem de ignorância técnica (inadvertida), são apenas problemas de planejamento ou execução, cujo enquadramento como DT não encontra acolhimento nessa perspectiva.

5.2 Distanciamento da Metáfora

Esse descolamento entre teoria e prática já havia sido apontado por autores como Avgeriou et al. [3], que alertam para o risco da banalização do conceito quando ele não é utilizado de forma estratégica. E, por vezes, ele é utilizado com intuito de atrair a atenção das pessoas [9].

Um dos achados deste estudo foi a apropriação do termo DT com a intenção de representar uma decisão consciente, tomada com o objetivo de viabilizar entregas ou contornar limitações temporárias do projeto. Esse uso está alinhado com o sentido original da metáfora, conforme proposto por Cunningham [8], que associava a DT ao compromisso assumido ao se optar por uma solução subótima no presente, com a expectativa de “pagamento” posterior. Apesar de representar a interpretação que encontra mais potencialidades nos estudos sobre assunto, essa abordagem foi observada em poucas *issues* analisadas.

A baixa incidência desse entendimento no material analisado reforça a necessidade de educação conceitual e alinhamento terminológico nas equipes de desenvolvimento. A relevância de se

identificar como ela vem sendo utilizada está na possibilidade de formular ações específicas para cada contexto de uso ou intenção. O emprego mais comum da DT nas *issues* está relacionado à descrição de tarefas não concluídas, representando uma interpretação pragmática que se distancia da intencionalidade original do conceito. Embora essa perspectiva seja útil na organização do *backlog* e na priorização de atividades, mais relacionadas ao gerenciamento de projetos, ela pode comprometer a eficiência das estratégias de GDT.

Uma possível explicação para esse fenômeno é a associação linguística direta entre as palavras “dívida” e “técnica”. O termo “dívida” frequentemente evoca a ideia de algo que precisa ser pago ou resolvido, enquanto “técnica” remete a tarefas operacionais como codificação e testes. Assim, quando uma tarefa não é entregue no prazo, ela é inconscientemente percebida como uma pendência a ser resolvida, ou seja, como uma “dívida” que é da equipe “técnica”. Essa interpretação reflete o uso prático e informal do conceito, mas compromete um dos primeiros passos do GDT que é a identificação [20]. A identificação é importante, pois estudos apontam que 21% do tempo dos desenvolvedores são gastos com atividades de DT [5], mas se falta a compreensão do que é entendido como DT, provavelmente, iremos tratar esse fenômeno de forma equivocada.

6 Ameaças à Validade

Esta seção apresenta as limitações do estudo com base em critérios de validade compatíveis com abordagens qualitativas interpretativas, conforme os critérios propostos no contexto da engenharia de software por Petersen e Gencel [23]. A validade, nesse paradigma, está relacionada à coerência interna da interpretação, à aplicabilidade em outros contextos e à transparência do processo analítico.

Credibilidade. Os achados foram construídos por meio de análise qualitativa de *issues* públicas extraídas do GitHub. O processo interpretativo teve caráter indutivo, com as categorias sendo construídas diretamente a partir da análise dos dados, por meio de leitura, codificação aberta e refinamento progressivo. Estratégias como múltiplas passagens pelo *corpus*, anotações reflexivas (*memos*) e revisão cruzada das codificações foram utilizadas para aumentar a confiança nas interpretações propostas. Uma amostra contendo a análise dos exemplos utilizados no resultados foi disponibilizada como recurso adicional na seção A.2.

Confirmabilidade. Embora a subjetividade do pesquisador seja reconhecida como elemento constitutivo da análise qualitativa, buscou-se reduzir seu impacto por meio da documentação de decisões analíticas e da rastreabilidade entre os dados e as interpretações. O uso de procedimentos sistemáticos e a disponibilização do material utilizado na análise (Disponibilidade de Artefato) permite que outros pesquisadores compreendam o caminho percorrido e avaliem a consistência dos achados.

Transferibilidade. O estudo se baseia em dados públicos de repositórios hospedados no GitHub, o que oferece um retrato legítimo de interações reais entre desenvolvedores. No entanto, os achados podem não ser diretamente aplicáveis a contextos corporativos privados, equipes com culturas distintas ou domínios organizacionais específicos. Cabe ao leitor, munido da descrição contextual fornecida na seção A.1, julgar a aplicabilidade dos resultados a outros cenários.

7 Conclusão

Metáforas e analogias são ferramentas valiosas para facilitar a comunicação e o entendimento de conceitos complexos, pois elas simplificam ideias abstratas ao compará-las com situações familiares e tangíveis. A metáfora da dívida técnica é um ótimo exemplo de como essas comparações podem tornar conceitos técnicos mais acessíveis. Entretanto, uma comunicação eficaz deve acontecer sem ambiguidades. Mesmo tratando-se de uma metáfora, é importante que haja um certo grau de consistência, a fim de evitar mal-entendidos e interpretações equivocadas. Isso é fundamental para promover um entendimento comum sobre a adoção de práticas que aprimorem a qualidade do software entregue no longo prazo [11].

Estratégias e técnicas desenvolvidas pela academia sobre DT não alcançam o mesmo efeito, ou nenhum efeito, em algumas das situações observadas neste estudo. Os achados revelaram quatro formas distintas de uso do termo DT, como decisão, atividade, problema e pendência, evidenciando os contextos em que ocorrem as principais distorções conceituais. Com base nesse diagnóstico, recomenda-se o estabelecimento de um vocabulário comum sobre o tema e a promoção de momentos de alinhamento entre os membros das equipes, especialmente em contextos ágeis, nos quais o termo é frequentemente utilizado como sinônimo de pendência. Essas recomendações têm implicações diretas para a indústria, ao orientar práticas de comunicação e gestão mais precisas, e para a academia, propõe-se explorar como as dinâmicas de GDT nas organizações são impactadas pela subutilização ou diluição do conceito original da DT, com o intuito de subsidiar a elaboração de ferramentas ou instrumentos que promovam a conscientização dos envolvidos no processo de desenvolvimento de software, ajudando a diminuir o distanciamento entre teoria e prática.

DISPONIBILIDADE DE ARTEFATO

Estão disponíveis na plataforma Zenodo³ documento contendo protocolo de coleta de dados, análise dos dados e informações para replicação do estudo. Também está disponível o *dataset* com *issues* coletadas em 01/06/2025 de acordo com os critérios descritos na seção 3.1 - Busca dos dados. Esse conteúdo pode ser obtido pelo link <https://doi.org/10.5281/zenodo.17330164>.

REFERENCES

- [1] Fábio Alexndrini and Edensil Brandl. 2023. Inglês e português: diferentes, mas próximos. *Revista de Extensão e Iniciação Científica da UNISOCIESC* 11, 1 (dez. 2023). <https://dalfovo.com/ojs/index.php/reis/article/view/446>
- [2] N. S. R. Alves, T. S. Mendes, M. G. de Mendonça, R. O. Spínola, F. Shull, and C. Seaman. 2016. Identification and management of technical debt: A systematic mapping study. *Information and Software Technology* 70 (2016), 100–121.
- [3] Paris Avgeriou, Ipek Ozkaya, Alexander Chatzigeorgiou, Marcus Ciolkowski, Neil A. Ernst, Ronald J. Koontz, Eltjo Poort, and Forrest Shull. 2023. Technical Debt Management: The Road Ahead for Successful Software Delivery. In *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)*. IEEE Computer Society, Los Alamitos, CA, USA, 15–30. doi:10.1109/ICSE-FoSE59343.2023.00007
- [4] Paris Avgeriou, Ipek Ozkaya, Heiko Koziulek, Zadia Codabux, and Neil Ernst. 2025. Manifesto from Dagstuhl Perspectives Workshop 24452 – Reframing Technical Debt. arXiv:2505.13009 [cs.SE] <https://arxiv.org/abs/2505.13009>
- [5] Terese Besker, Antonio Martini, and Jan Bosch. 2019. Software developer productivity loss due to technical debt—A replication and extension study examining developers’ development work. *Journal of Systems and Software* 156 (2019), 41–61. doi:10.1016/j.jss.2019.06.004

³<https://zenodo.org>

- [6] Glenn Bowen. 2009. Document Analysis as a Qualitative Research Method. *Qualitative Research Journal* 9 (08 2009), 27–40. doi:10.3316/QRJ0902027
- [7] Virginia Braun and Victoria Clarke. 2021. *Thematic Analysis: A Practical Guide*. SAGE Publications, London.
- [8] Ward Cunningham. 1992. The WyCash Portfolio Management System. <http://c2.com/doc/oopsla92.html> Presented at OOPSLA '92 Experience Report.
- [9] George Fairbanks. 2020. Ur-Technical Debt. *IEEE Software* 37, 4 (2020), 95–98. doi:10.1109/MS.2020.2986613
- [10] Tiago César Farinelli and Marcelo Augusto Ambrozini. 2021. Alavancagem ótima e gerenciamento da estrutura de capital nas empresas brasileiras do agronegócio. *Revista de Administração FACES Journal* 20, 2 (Apr./Jun. 2021), 49–68. <https://revista.fumec.br/index.php/facesp/article/view/8694> Publicado online em 09/03/2022.
- [11] Markus Finke, Thomas Neff, and Tobias Reichl. 2023. How to introduce TD Management into a Software Development Process – A Practical Approach. In *2023 ACM/IEEE International Conference on Technical Debt (TechDebt)*. 52–61. doi:10.1109/TechDebt59074.2023.00013
- [12] Martin Fowler. 2009. Technical Debt Quadrant. <https://martinfowler.com/bliki/TechnicalDebtQuadrant.html> Accessed: 2025-06-20.
- [13] J. Holvitie, S. A. Licorish, R. O. Spinola, S. Hyrynsalmi, S. G. MacDonell, T. S. Mendes, J. Buchan, and V. Leppänen. 2018. Technical debt and agile software development practices and processes: An industry practitioner survey. *Information and Software Technology* 96 (2018), 141–160. doi:10.1016/j.infsof.2017.11.015
- [14] Helvio Jeronimo Junior and Guilherme Horta Travassos. 2022. Consolidating a common perspective on Technical Debt and its Management through a Tertiary Study. *Information and Software Technology* 149 (2022), 106964. doi:10.1016/j.infsof.2022.106964
- [15] Philippe Kruchten, Robert L. Nord, and Ipek Ozkaya. 2012. Technical Debt: From Metaphor to Theory and Practice. *IEEE Software* 29, 6 (2012), 18–21. doi:10.1109/MS.2012.167
- [16] Zhilei Li, Paris Avgeriou, and Peng Liang. 2015. A Systematic Mapping Study on Technical Debt and Its Management. *Journal of Systems and Software* 101 (2015), 193–220. doi:10.1016/j.jss.2014.12.027
- [17] Erin Lim, Nitin Taksande, and Carolyn Seaman. 2012. A Balancing Act: What Software Practitioners Have to Say about Technical Debt. *IEEE Software* 29, 6 (2012), 22–27. doi:10.1109/MS.2012.130
- [18] Gilberto Luiz, Neuza Silva, and Marcelo Pinto. 2022. Os significados do crédito para as famílias de baixa renda. *RACE - Revista de Administração, Contabilidade e Economia* 20 (04 2022), 453–476. doi:10.18593/race.22099
- [19] Steve McConnell. 2008. Managing technical debt. *Construx* (2008). <https://www.construx.com/resources/whitepaper-managing-technical-debt/> Accessed: 2025-05-17.
- [20] Ana Carolina Candido De Melo, Nathália Accioly, Roberta Fagundes, and Wylliams Santos. 2023. Identifying and Measuring Technical Debt in Software Requirements: a supporting guide. In *Proceedings of the XIX Brazilian Symposium on Information Systems (SBSI '23)*. ACM, 356–363. doi:10.1145/3592813.3592925
- [21] Matthew B. Miles, A. Michael Huberman, and Johnny Saldaña. 2014. *Qualitative Data Analysis: A Methods Sourcebook* (3rd ed.). SAGE Publications, Thousand Oaks, CA.
- [22] Hani Morgan. 2022. Conducting a Qualitative Document Analysis. *The Qualitative Report* (01 2022). doi:10.46743/2160-3715/2022.5044
- [23] Kai Petersen and Cigdem Gencel. 2013. Worldviews, Research Methods, and their Relationship to Validity in Empirical Software Engineering Research. In *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement*. 81–89. doi:10.1109/IWSM-Mensura.2013.22
- [24] Eric Ries. 2011. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business, New York.
- [25] N. Rios, R. O. Spinola, M. Mendonça, and C. Seaman. 2018. A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. *Information and Software Technology* 70 (2018), 100–121.
- [26] Ciro Goulart dos Santos. 2016. *Um estudo empírico sobre a gerência de dívida técnica em projetos de desenvolvimento de software que utilizam Scrum*. Dissertação de Mestrado. Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Faculdade de Informática, Porto Alegre, Brasil. Orientador: Prof. Dr. Rafael Prikladnicki.
- [27] Carolyn Seaman and Y. Guo. 2011. Measuring and Monitoring Technical Debt. In *Proceedings of the International Conference on Software Engineering*. 485–486. doi:10.1145/1985793.1985904
- [28] Victor Machado Silva, Helvio Jeronimo Junior, and Guilherme Horta Travassos. 2019. A Taste of the Software Industry Perception of Technical Debt and its Management in Brazil. *Journal of Software Engineering Research and Development* 7 (Jul. 2019), 1:1 – 1:16. doi:10.5753/jsred.2019.19
- [29] E. Tom, A. Aurum, and R. Vidgen. 2013. An exploration of technical debt. *Journal of Systems and Software* 86, 6 (2013), 1498–1516.
- [30] Nirnaya Tripathi, Markku Oivo, Kari Liukkunen, and Jouni Markkula. 2019. Startup ecosystem effect on minimum viable product development in software

startups. *Information and Software Technology* 114 (2019), 77–91. doi:10.1016/j.infsof.2019.06.008

- [31] Jesse Yli-Huuma, Andrey Maglyas, and Kari Smolander. 2016. How do software development teams manage technical debt? – An empirical study. *Journal of Systems and Software* 120 (2016), 195–218. doi:10.1016/j.jss.2016.05.018

A Recursos Adicionais

Tabelas adicionais e exemplos da análise realizada na seção de Resultados.

A.1 Tabela de categorias e códigos

Todas as categorias e códigos foram definidas de forma resumida nas Tabela 1 e 2.

A.2 Exemplo da análise

Todos os exemplos de *issues* apresentados na seção 4 - Resultados foram incluídos na Tabela 3, que apresenta os passos da leitura exploratória (*jotting*) e da leitura analítica (*memoing*) realizados durante a análise.

Tabela 1: Categorias que emergiram das percepções da DT

Categoria: descrição	Exemplo de percepções [Contexto de Uso]
Decisão: DT como justificativa para escolhas técnicas feitas para atender prazos ou contornar limitações.	Entregar um escopo menor de forma consentida, registrando os impactos futuros, para não atrasar uma entrega [DEC]; Implementar uma solução não ideal com a intenção de entregar dentro prazo acordado [DEC]
Atividade: DT associada a tarefas técnicas como testes, refatoração ou documentação, com foco em manter a qualidade.	Criar testes [ATT]; Elaborar documentação [ATT]; Configurar ferramentas [ATT]; Refatorar código [AAM]; Atualização de bibliotecas utilizadas no sistema [AAM]
Problema: DT tratada como erro, falha ou limitação que precisa ser resolvida.	Correção de erro ou bug [PCF]; Resolver ou solucionar problemas em produção [PCF]; Tarefas concluídas com resultado diferente do esperado [PNC]; Estado de alerta para não criar um problema [PAE];
Pendência: DT usada para indicar tarefas não finalizadas, adiadas ou bloqueadas.	Alteração do escopo pela descoberta de novas necessidades [PTE]; Replanejamento de entregas não realizadas [PTN];

Tabela 2: Resumo dos contextos de uso em que o termo DT foi utilizado

Código	Contexto de Uso	Descrição
DEC	Tomada de Decisão	Refere-se a escolhas conscientes para viabilizar entregas ou contornar restrições. A DT é registrada como uma concessão técnica planejada e justificada, próxima da definição original.
ATT	Tarefa Técnica	Representa o registro de tarefas técnicas a serem executadas, como testes, ajustes ou configurações. A DT funciona como uma lista de pendências técnicas, sem necessariamente implicar adiamento.
AAM	Ação de Melhoria	Ações que buscam aprimorar a estrutura do código, desempenho ou arquitetura. A DT está ligada a melhorias contínuas e pode ser percebida como atividades de GDT, pagamento ou prevenção.
PCF	Correção de Falha	Abrange erros e defeitos identificados em produção ou durante o uso do sistema. A DT é empregada como rótulo para problemas que impactam o funcionamento e exigem correção.
PNC	Não Conformidade	Situações em que falhas são encontradas em revisões, inspeções ou testes. A DT expressa desvios em relação a critérios estabelecidos e remete à necessidade de ajustes.
PAE	Ameaça a Ser Evitada	Uso preventivo de boas práticas para alertar sobre padrões que podem gerar problemas futuros. A DT é tratada como algo negativo e que não deve ser utilizado.
PTE	Tarefa Emergente	Tarefas não previstas inicialmente que surgem durante a execução. A DT é usada para destacar novas demandas não planejadas, incorporadas ao backlog.
PTN	Tarefa Não Concluída	Atividades que estavam previstas, mas não foram concluídas por falta de tempo ou recursos. A DT comunica o replanejamento e a postergação da entrega.

Tabela 3: Exemplo da análise realizada nas *issues* utilizadas na seção 4 - Resultados

ID	Título	...	<i>Jotting</i>	<i>Memoing</i>	Contexto de Uso	Categoria
119	Criar página que lista serviços	...	Descreve uma decisão clara e inequívoca do conceito de DT	O detalhamento da issue deixa claro qual é a motivação de implementar uma solução não ideal. E, principalmente, a intenção de entregar dentro prazo acordado.	Tomada de decisão	Decisão
234	Refatoração	...	Denota uma intenção de melhoria do código. O título reforça essa percepção.	A intenção é busca por "algo" que poderia ter sido feito melhor e melhorar. Ou seja, não tem carácter de corrigir o que não esteja funcionando. O contexto está mais relacionado ao processo de melhoria contínua.	Ação de melhoria	Atividade
220	Dívida técnica da tela da home	...	Sugere uma atividade de verificação	É cadastrada uma atividade que recomenda uma verificação da responsividade de uma tela da aplicação.	Registro de atividade	Atividade
224	Dívida técnica de refatorar a tela de editar perfil	...	Uso do termo para indicar um problema encontrado e que deve ser resolvido	A intenção da issue é informar que o comportamento da tela não estava funcionando com esperado. Testes ou verificações foram feitas para atestar o problema. Ele não pode ser considerado um erro ou falha. Essa não conformidade foi classificada como DT.	Não conformidade	Problema
229	Dívida técnica: ajustar queries de find para Aluno, Disciplina, Nota, Turma	...	Uso do termo para representar um erro	Apesar da intenção e apontar um erro estar clara no título. A resposta de outro colaborador ainda confirma que foi feita uma correção: "Foi corrigido o erro no relacionamento das tabelas aluno/disciplina/nota/turma, com adição de view para filtragem de dados e performance..."	Correção de falhas	Problema
248	mackdown nuel	...	Recomendação sobre como prevenir a DT	Faz recomendações sobre padronização e boas práticas de projeto, que o grupo deve utilizar para evitar DT	Ameaça a ser evitada	Problema
167	Adicionar Footer Front-end e desenhar design da Busca Avançada	...	Atividade não concluída quando planejada e ficou como "dívida de execução"	A conversa do grupo deixa claro que a tarefa estava planejada para release 3. Mas como não tinha sido concluído, um membro do grupo informa que foi acrescentada na release atual como "dívida técnica".	Não entrega do escopo planejado	Pendência
156	Linha temporal do ato do processo licitatório	...	Aparentemente encontraram novas necessidades que não estavam no escopo inicial (olhar melhor contexto)	A anotação que lista "Dívidas Técnicas"(tarefas) foi incluída após alguns dias de execução e teve o intuito de completar a necessidade da issue, visto que não foi cadastrada uma issue a parte e a ela foi toda concluída. O termo está muito mais relacionado ao registro de alguma pendência do que na decisão de registrar algo para resolução futura para que algo fosse concluído no prazo.	Demanda emergente	Pendência