TechDebt Tracker

Um framework visual para o gerenciamento de dívida técnica em startups

Mayra Daher de Carvalho Pereira

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA
Ciência da Computação

João Pessoa, 2025

Mayra Daher de Carvalho Pereira

TechDebt Tracker: Um framework visual para o gerenciamento de dívida técnica em startups

Artigo científico apresentado ao curso Ciência da computação
do Centro de Informática, da Universidade Federal da Paraíba,
como requisito para a obtenção do grau de Bacharel em titulo

Orientador: Adriana Carla Damasceno

Março de 2025

P436t Pereira, Mayra Daher de Carvalho.
TechDebt Tracker: um framework visual para o gerenciamento de dívida técnica em startups / Mayra Daher de Carvalho Pereira. - João Pessoa, 2024.
14 f. : il.

Orientação: Adriana Carla Damasceno.
TCC (Graduação) - UFPB/CI.

1. Engenharia de software. 2. Dívida técnica. 3. Framework. 4. Desenvolvimento de software. 5. Startups. I. Damasceno, Adriana Carla. II. Título.

UFPB/CI CDU 004.41

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Ciência da computação, intitulado ***TechDebt Tracker: Um framework visual para o gerenciamento de dívida técnica em startups*** de autoria de Mayra Daher de Carvalho Pereira, aprovada pela banca examinadora constituída pelos seguintes professores:

gov.br
Documento assinado digitalmente
ADRIANA CARLA DAMASCENO
Data: 25/03/2025 21:01:33-0300
Verifique em https://validar.iti.gov.br

Profa. Dra. Adriana Carla Damasceno
Universidade Federal da Paraíba

gov.br
Documento assinado digitalmente
DANIELLE ROUSY DIAS RICARTE
Data: 27/03/2025 21:21:45-0300
Verifique em https://validar.iti.gov.br

Profa. Dra. Danielle Rousy Dias Ricarte
Universidade Federal da Paraíba

gov.br
Documento assinado digitalmente
YUSKA PAOLA COSTA AGUIAR
Data: 27/03/2025 09:59:41-0300
Verifique em https://validar.iti.gov.br

Profa. Dra. Yuska Paola Costa Aguiar
Universidade Federal da Paraíba

João Pessoa, 25 de março de 2025

Centro de Informática, Universidade Federal da Paraíba
Rua dos Escoteiros, Mangabeira VII, João Pessoa, Paraíba, Brasil CEP: 58058-600
Fone: +55 (83) 3216 7093 / Fax: +55 (83) 3216 7117

TechDebt Tracker: Um framework visual para o gerenciamento de dívida técnica em startups

Mayra Daher[1], **Adriana Damasceno**[1]

[1]Centro de Informática – Universidade Federal da Paraíba (UFPB)
João Pessoa, Paraíba, Brasil, CEP 58051-900

mayrapereira@cc.ci.ufpb.br, adriana.damasceno@academico.ufpb.br

Abstract. *Technical Debt has become a relevant topic in software engineering, being related to the consequences of introducing immature or incomplete artifacts during software development. Several studies address it in the context of more mature companies, however, few studies focus on startups. Considering the informality in the adoption of software engineering best practices by startups, the objective of this work was to develop a framework to assist them in managing Technical Debt, called TechDebt Tracker. For this purpose, the Design Science Research method was used. According to the results obtained, TechDebt Tracker proved to be effective in managing Technical Debt, as well as simple to understand and implement.*

Resumo. *A Dívida Técnica tem se mostrado relevante na engenharia de software, estando relacionada às consequências da inserção de artefatos imaturos ou incompletos durante o desenvolvimento de software. Diversos estudos abordam-na no contexto de empresas mais maduras, porém poucas pesquisas focam em startups. Considerando a informalidade na adoção de boas práticas de engenharia de software por startups, o objetivo deste trabalho foi desenvolver um framework para auxiliá-las no gerenciamento da Dívida Técnica, o TechDebt Tracker. Para isso, foi utilizado o método Design Science Research. De acordo com os resultados obtidos, o TechDebt Tracker mostrou-se relevante no gerenciamento da Dívida Técnica, além de simples de entender e executar.*

1. Introdução

São várias as designações para conceituar *startups de software* na literatura. Giardino *et al.* [Giardino et al. 2015] as definem como "Organizações focadas na criação de produtos de alta tecnologia e inovadores, com pouco ou nenhum histórico operacional, com o objetivo de expandir agressivamente seus negócios em mercados altamente escaláveis".

Em decorrência do *time to market*, dos recursos limitados, da imaturidade da equipe, da dinamicidade dos mercados, entre outros, as *startups* de desenvolvimento de *software* acabam negligenciando as boas práticas de engenharia de *software* ou as praticam de maneira informal. De acordo com Pompermaier [Pompermaier 2021], "esta informalidade aumenta os riscos de geração de dívida técnica nas diferentes etapas do desenvolvimento de *software*, seja no código criado, na arquitetura, nos testes e na documentação."

Dívida técnica refere-se ao efeito de qualquer artefato incompleto, imaturo ou inadequado no ciclo de vida de desenvolvimento de *software* [Seaman and Guo 2011]

e pode ser o resultado de decisões técnicas ou de negócios que podem comprometer a qualidade do código à longo prazo. A Dívida Técnica pode ser do tipo ingênua, decorrente da imaturidade do negócio ou do time de desenvolvedores, por exemplo. Pode ser do tipo inevitável, tal como quando o design do *software* não evolui bem. Mas também pode ser do tipo estratégica, quando a *startup* deseja aproveitar uma oportunidade de negócio. As consequências do seu acúmulo podem ser graves para a empresa. Podemos citar a atrofia do produto, frustração do cliente e do time de desenvolvedores, aumento dos custos de desenvolvimento em decorrência de retrabalho, ou em casos extremos a morte da *startup*, entre outros.

Vários estudos abordam a dívida técnica dentro do contexto das empresas mais maduras. Seaman e Guo [Seaman and Guo 2011], por exemplo, propõem um framework para gerenciamento de dívida técnica composto por 3 etapas: identificação, mensuração e monitoramento. Outras abordagens tratam especificamente da identificação, mensuração ou dão suporte na tomada de decisão para pagamento de itens de DT. Porém, ainda são poucos os trabalhos que focam suas pesquisas de dívida técnica em *startups*.

Diante disso, esse estudo tem como objetivo dar suporte a *startups* de software quanto ao tema dívida técnica. Para tal, apresentamos um framework para auxílio do gerenciamento de dívida técnica por startups e assim mitigar suas consequências. Sabendo da informalidade da adoção de boas práticas de engenharia de *software* por *startups*, a principal característica da proposta é a simplicidade com o finalidade de facilitar a adesão.

2. Referencial Teórico

O sucesso na realização de um projeto de software para startups depende de vários fatores, dentre eles os passos adotados para sua execução. Diversas abordagens de processos foram adotadas, a exemplo do modelo V, waterfall e o iterativo e incremental [Valente 2020]. No entanto, as taxas de insucesso em projetos de software para startups chegavam a 70% devido a vários fatores, incluindo a falta de feedback do cliente, mudança de requisitos e falta de conhecimento em tecnologias usadas [Pompermaier 2021]. Estes projetos dependem principalmente da experiência de sua equipe de desenvolvimento. Este estudo também mostra que startups nas fases inicial e de validação escolhem práticas de Engenharia de Software de forma ad-hoc e baseada no conhecimento do time. Ainda segundo [Pompermaier 2021], quando a startup entrega seu produto para o mercado, percebe que poderiam ter adotado práticas de software mais consolidadas.

Dentre as práticas de engenharia de software que poderiam ser melhor usadas em startups de software, está o gerenciamento da dívida técnica. Visando compreender as dificuldades em torno deste tema, diversas pesquisas vem sendo feitas ao longo dos últimos anos. A metáfora da dívida técnica foi criada em 1992 por Cunningham [Rubin 2018] para fazer alusão a artefatos imaturos inseridos no software durante o seu desenvolvimento. Para Seaman e Guo [Seaman and Guo 2011], esta metáfora foi estendida para se referir a qualquer artefato imperfeito no ciclo de vida do software. Segundo Giardino et al. [Giardino et al. 2015], as startups enfrentam *trade-off* entre a necessidade de velocidade na entrega de resultados e qualidade em diversas etapas do desenvolvimento e por conta disso acumulam dívida técnica. Outros conceitos também se referem à dívida técnica, como o custo para resolver a dívida (principal) e juros decorrentes de se admitir a

dívida técnica, por exemplo. A dívida técnica pode trazer benefícios de curto prazo para o projeto ou consequências desastrosas. Podemos citar como benefícios: o aumento da produtividade, entrega do software em um prazo encurtado e aproveitamento de oportunidades de negócios. Já como como consequências negativas, temos a queda da qualidade do *software* a longo prazo, sub-performance e até atrofia do produto.

Haja vista que o acúmulo de dívida técnica pode trazer graves consequências, como aumento nos custos, baixa performance e insatisfação do cliente, o principal objetivo de seu gerenciamento é mitigar seus impactos negativos. Porém, segundo Rubin [Rubin 2018], nem toda dívida deve ser paga. Deve-se pagar primeiro aquelas com juros maiores e de maneira incremental enquanto realiza-se trabalho de valor para o cliente. Para facilitar seu gerenciamento, algumas abordagens foram propostas ao longo dos anos, como a abordagem de portfólio e o cálculo dos juros da dívida técnica. Para Ampatzoglou *et al.* [Ampatzoglou et al. 2015], ainda existem contradições entre os pesquisadores no que tange à gestão dos juros. Eles propõem um framework que leva em consideração as atuais teorias econômicas. Já De Almeida [De Almeida et al. 2021] entrega uma abordagem de priorização de dívida técnica orientada para negócios.

O uso de frameworks é frequentemente difundido em projetos de software. Um framework é um conjunto de funcionalidades que podem ser utilizados em vários projetos, uma abstração que une códigos comuns, conforme [Sbrocco and Macedo 2012]. Scrum [Opelt et al. 2023] é um exemplo de framework para gerenciamento de projetos de software. Ele pressupõe a existência de reuniões regulares entre os membros do time; sprints (delimitação das atividades em um tempo); papéis definidos, a exemplo de partes interessadas no projeto; e artefatos, como o backlog do produto (lista de itens do produto), sprint backlog (lista de itens da sprint), e incremento (produto entregue depois de uma sprint). Ainda de acordo com Rubin [Rubin 2018], um outro artefato pode ser adicionado ao *Scrum* para dar suporte à visualização e gerenciamento de dívida técnica: o backlog de dívida técnica. O backlog de dívida técnica pode ficar perto do sprint backlog, para que durante o sprint planning, itens de dívida técnica possam ser considerados para pagamento no sprint seguinte.

Business Model Canvas também é um framework e pode ser utilizado de maneira estratégica para dar suporte ao desenvolvimento de novos modelos de negócio, análise ou melhoria dos já existentes. Desenvolvido por por Alexander Osterwalder e Yves Pigneur, tem como ponto forte a característica de ser visual, o que facilita a comunicação e o entendimento sobre a entrega de valor da empresa, tornando-se assim, uma importante ferramenta para empreendedores na construção de seus negócios orientados a mercado, segundo Alan *et al.* [Murray and Scuotto 2015].

Também desenvolvido por Alexander Osterwalder e Yves Pigneur, o Canva de Proposta de Valor é uma ferramenta que aprofunda o Business Model Canvas e auxilia na construção da proposta de valor para o cliente e tem como objetivo alinhar os desejos e necessidades do cliente com as entregas feitas pelo produto ou serviço. O Canva de Proposta de Valor é composto por 2 blocos: Cliente e Mapa de proposta de valor. O componente Cliente trata de investigar as tarefas, dores e ganhos do cliente, enquanto o Mapa de Proposta de Valor foca no que a empresa pretende oferecer. De acordo com Silva [Silva 2012] o Canva de Proposta de Valor presume a aplicação de conceitos do *Leans Startups e Customer Development* para garantir o alinhamento entre as necessidades do

usuário e o framework.

Segundo Prestes [Prestes et al. 2020], é possível a adoção de práticas de Design Thinking em diversas áreas de projetos de software. Para Moreira [Ruschel 2019], o Design Thinking refere-se a um modelo de pensamento que os designers utilizam durante o ato de projetar e tem como resultado o desenvolvimento de artefatos sejam eles digitais ou não. O Design Thinking envolve a habilidade de sintetizar conhecimento de fontes diversas, por isso é considerada de caráter multidisciplinar. O conjunto de etapas que compõem o Design Thinking pode variar de acordo com o autor, mas ainda segundo Moreira[Ruschel 2019], são 6: a imersão preliminar, imersão profunda, síntese e análise de dados, ideação, prototipação e testes.

3. Métodos

Esta é uma pesquisa qualitativa, isto é, tem o objetivo de entender aspectos subjetivos sobre o uso da dívida técnica em startups de software. Ela foi desenvolvida utilizando o método de pesquisa Design Science Research [Wieringa 2014]. O Design Science Research é o método de pesquisa utilizado para a construção do conhecimento no contexto do Design Science. Segundo Dresch *et al.* [Dresch et al. 2020], tem como principais conceitos a relevância e o rigor, o que garante a confiabilidade dos resultados. Além disso, o Design Science Research tem como objetivos desenvolver artefatos, prescrever soluções e estudar o artifical. Seu principal produto é o artefato feito pelo homem. Foram adotadas as etapas propostas por Peffers *et al.* [Peffers et al. 2007] para este método, que são: (1) Identificação e motivação do problema, (2) Definição dos objetivos para uma solução, (3) Projeto e desenvolvimento, (4) Demonstração, (5) Avaliação e (5) Comunicação.

Na etapa 1 (identificação e motivação do problema), foi identificada a falta de adoção de boas práticas na construção de software por startups através da observação de como os projetos de software eram conduzidos para este tipo de empresa. Para entender o escopo do problema, foi feita uma pesquisa do estado da arte em bases de dados como SBC OpenLib[1], IEEE Xplorer [2], TD Researcher Team [3] e ACM Digital Library [4] com as palavras-chave de pesquisa: “Débito técnico”, “dívida técnica”, “technical debt”, “engenharia de software” e “software engineer”. A palavra chave startup não foi incluída dentre estes termos porque buscou-se obter resultados mais gerais de como a dívida técnica era usada em diferentes contextos da engenharia de software além das startups. Com isso, foi identificado que trabalhos sobre o gerenciamento de dívida técnica em startups não foram propostos no contexto de startups de software.

Após a investigação sobre dívida técnica e engenharia de software, foi identificado que diversos artigos tratam da relação entre dívida técnica e métodos ágeis [Pompermaier 2021], o que evidenciou a necessidade de explorar esse tema e suas conexões com o surgimento da dívida técnica em projetos de desenvolvimento de software. Também foi constatado que os métodos ágeis mais utilizados em pequenos projetos de software são o Scrum, Kanban e Lean, conforme apontado por Mendes. [Mendes 2020].

Na etapa 2 (definição dos objetivos para uma solução), foi utilizado o template

[1]https://sol.sbc.org.br/index.php/indice

[2]https://ieeexplore.ieee.org/Xplore/home.jsp

[3]https://www.tdresearchteam.com/

[4]https://dl.acm.org/

para problemas de design proposto por Wieringa [Wieringa 2014]. Com isso, chegamos ao seguinte objetivo geral: ”Como minimizar os gastos com o retrabalho de tarefas propondo um framework que gerencia dívida técnica em startups de software para ajudar gerentes de projetos usando métodos ágeis a rastrear e priorizar dívida técnica?”. Os principais desafios incluem a falta de capacitação dos membros das startups na área de engenharia de software, recursos e ambientes computacionais disponíveis, assim como decisões de negócio inadequadas e que colaboram para o aumento da dívida técnica do projeto.

No projeto e desenvolvimento (etapa 3), foi proposta a construção de um *framework* chamado TechDebit Tracker. Para isso, foi utilizado o Business Model Canvas e o Canvas de Proposta de valor com o objetivo de identificar e validar diretrizes estratégicas que agregariam valor para o desenvolvimento do artefato. Em seguida, aplicou-se o Design Thinking [Ruschel 2019] no processo criativo de geração de ideias. O propósito foi entender as necessidades e desafios dos gerentes de projetos e focar na experiência prévia que eles possam trazer para contribuir na construção do software. Com isso, seguiram-se as etapas: imersão preliminar, onde foram analisados itens similares; imersão profunda, com definição de personas e mapa de valores dos stakeholders; síntese e análise das informações (definição de insights e prioridades); ideação e prototipação (utilizando o software Trello)[5].

A demonstração (etapa 4) consistiu de três reuniões remotas usando o Google Meet [6] para apresentar o framework a uma startup do ramo de tecnologia e colher feedbacks. A startup Wisecare desenvolve soluções corporativas baseadas em tecnologias síncronas que viabilizam a comunicação das pessoas. Nasceu de um projeto de pesquisa em 2019 e atualmente tem 29 colaboradores. Alguns de seus produtos são: plataforma de teleconsulta, plataforma de gerenciamento de agenda e plataforma de educação online. Foi usado o aplicativo de mensagens WhatsApp[7] para o esclarecimento de dúvidas pontuais na aplicação do framework. Nesta apresentação, as etapas do framework e seus respectivos artefatos foram detalhados com a ajuda do manual visual desenvolvido pela autora [8]. Ainda nesta etapa, a startup participou de uma avaliação de pequena escala [Wohlin and Rainer 2022], isto é, uma versão simplificada de um estudo de caso. A avaliação em pequena escala pode ser utilizada quando um novo artefato é desenvolvido e é necessário avaliar seu valor, eficácia e capacidade inovadora.

A avaliação (etapa 5) foi feita em 2 etapas: antes e após a aplicação do framework e com 2 questionários que usam a escala Likert [Dawes 2008] com respostas variando de concordo totalmente para discordo totalmente. O objetivo foi estabelecer uma linha base antes e depois da execução do framework e contrastar os resultados para avaliar os efeitos de sua aplicação na startup. O questionário aplicado antes da execução do framework é composto por oito afirmações para medir o conhecimento do entrevistado sobre conceitos de dívida técnica. A Figura 1 apresenta as questões do questionário.

[5]https://trello.com/

[6]https://meet.google.com/

[7]https://web.whatsapp.com/

[8]Disponível em https://drive.google.com/drive/folders/16fYO9b4JdYAGANy6wfJPu66aFMxa6-1T?usp=sharing

1. Em uma escala de 1 a 5, qual o seu nível de concordância com as afirmações abaixo?
   (a) Tenho amplo conhecimento sobre Dívida Técnica e seu impacto na qualidade do software;
   (b) Aplico abordagens de Identificação de Dívida Técnica com êxito;
   (c) Aplico abordagens de mensuração do custo de Dívida Técnica, utilizando métricas, com êxito;
   (d) Aplico abordagens de monitoramento de Dívida Técnica com êxito;
   (e) Aplico abordagens de priorização do pagamento de Dívida Técnica com êxito;
   (f) Calculo com êxito o custo (horas/ano) do retrabalho necessário para correção de itens de Dívida Técnica;
   (g) Calculo com êxito o custo ($/ano) do retrabalho necessário para correção de itens de Dívida Técnica;
   (h) Incluo o custo ($/ano) do retrabalho necessário para correção de itens de Dívida Técnica na precificação do software desenvolvido.

Figura 1. Questionário pré-framework

O questionário aplicado depois da execução do framework é composto de dezoito questões que procuram capturar os benefícios adquiridos e técnicas adotadas pela startup em relação à dívida técnica, tais como priorização, monitoramento, mensuração de retrabalho e comunicação entre as partes interessadas no projeto. A Figura 2 apresenta as questões do questionário.

1. Em uma escala de 1 a 5, qual o seu nível de concordância com as afirmações abaixo em relação ao uso do framework TechDebitTracker:
   (a) Aplicar o framework incrementou o meu conhecimento sobre Dívida Técnica;
   (b) Aplicar o framework incrementou a utilização de abordagens de mensuração de Dívida Técnica;
   (c) Aplicar o framework incrementou a utilização de abordagens de monitoramento de Dívida Técnica;
   (d) Aplicar o framework incrementou a utilização de abordagens de priorização do pagamento;
   (e) Aplicar o framework incrementou o cálculo do custo (horas/ano) do retrabalho necessário para correção de Dívida Técnica;
   (f) Aplicar o framework incrementou o cálculo do custo ($/ano) do retrabalho necessário para correção de dívida técnica;
   (g) O framework é relevante para auxiliar gestores de startups no gerenciamento de Dívida Técnica;
   (h) O template de registro de itens de dívida técnica são relevantes para realizar descrições e monitoramento do itens do projeto;
   (i) As fórmulas são relevantes para a mensuração da dívida técnica;
   (j) As fórmulas são relevantes para a tomada de decisões referentes à dívida técnica;
   (k) O quadro Kanban é relevante para o gerenciamento de itens de dívida técnica;
   (l) O quadro Kanban auxilia na visualização dos itens de dívida técnica;
   (m) O fluxograma apresentado é relevante para o gerenciamento de itens de dívida técnica;
   (n) O fluxograma apresentado é compatível com o modelo de gerenciamento de dívida técnica já existente na startup;
   (o) A proposta do framework é simples de ser entendida;
   (p) O framework é simples de ser executado;
   (q) Aplicar o framework auxiliou na comunicação entre as partes interessadas no projeto em um momento de tomada de decisão sobre dívida técnica;

Figura 2. Questionário pós-framework

A etapa 6 (comunicação) consistiu de apresentação na instituição de ensino superior onde esta pesquisa foi realizada e elaboração de um relatório de atividades.

4. Resultados e Discussões

A etapa de projeto e desenvolvimento do Design Science Research resultou no framework TeckDebt Tracker. O modelo teve como principal referência o trabalho de análise custo-benefício proposto por Seaman e Guo [Seaman and Guo 2011] e é composto por um conjunto de artefatos: (1) um processo de gerenciamento da dívida técnica; (2) um quadro Kanban no *Trello*; (3) um modelo de cartão Kanban para registro do item de dívida técnica; (4) um conjunto de fórmulas para mensuração e priorização de pagamento de dívida técnica e (5) um manual visual com orientações para auxiliar gestores e equipes na adesão e implementação do framework. A Figura 3 mostra como estes artefatos compõem o framework TechDebit Tracker.

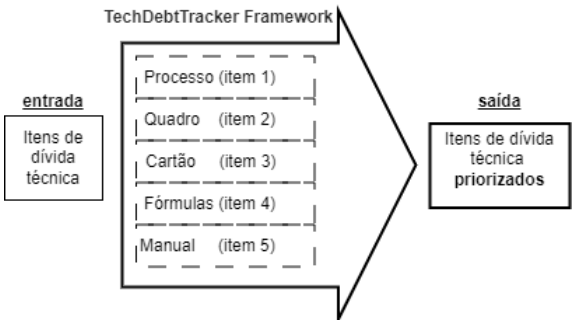

Figura 3. Componentes do framework

O processo de gerenciamento (artefato 1) representa visualmente as etapas que devem ser seguidas para cada item de dívida técnica do projeto. Ele foi desenvolvido para facilitar a compreensão das etapas e decisões, melhorar a comunicação da equipe e partes interessadas, assim como dar suporte à padronização dos processos relativos à dívida técnica. Os retângulos representam as etapas do processo e os losangos representam pontos de decisão. O fluxograma é mostrado na Figura 4.

O modelo de cartão para o registro do item de dívida técnica (Artefato 2) é apresentado na Figura 5, nele os itens de dívida técnica são registrados. Este modelo foi adaptado a partir do trabalho de Oliveira [Oliveira 2015] e sua finalidade é fornecer detalhes de novos itens de dívida técnica para facilitar a tomada de decisão. Também ajuda a identificar os tipos de dívida técnica mais cometidos pela equipe e os desenvolvedores que mais entregam artefatos imaturos para traçar estratégias de prevenção ou amortização de dívida técnica. Ele é dividido em três seções: identificação, medição e priorização da dívida técnica. Na seção identificação, são fornecidos os campos: id (identificador), data de identificação, responsável, tipo de dívida técnica (de projeto, de requisitos, de arquitetura, de testes, etc), rastreio (localização no código), descrição e intencionalidade (se a dívida técnica foi deliberada, consciente). A seção medição, é preenchida posteriormente e contém os campos *principal* e juros. O campo *principal* recebe respostas numéricas e representa a quantidade de horas estimadas para uma dívida técnica ser resolvida na data atual. Para preencher esse campo, deve-se utilizar o resultado obtido ao calcular o *principal*. O campo juros representa a estimativa de trabalho futuro extra que será necessário se esse item da dívida não for pago. Para preencher esse campo deve-se utilizar o resultado obtido ao calcular os juros. Na seção priorização, o campo tomada de decisão compara

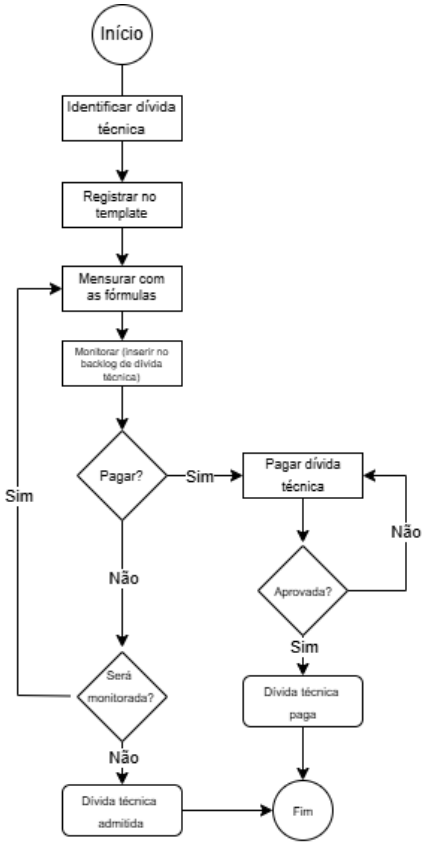

Figura 4. Processo de gerenciamento de dívida técnica

os valores absolutos dos campos juro e principal. Caso os juros sejam maiores que o principal, a dívida técnica deve ser paga. Se não, ela é considerada admitida.

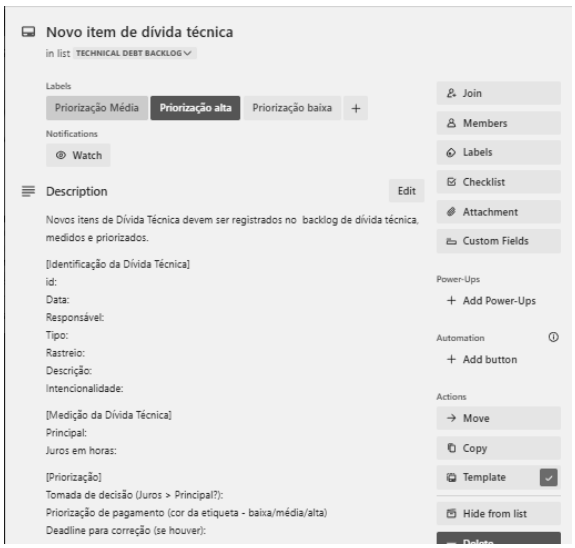

Figura 5. Modelo para registro do item de dívida técnica

O quadro Kanban (Artefato 3) pressupõe que o projeto usa o método ágil Scrum e foi adaptado do trabalho de Santos [Santos et al. 2016]. Este artefato tem como obje-

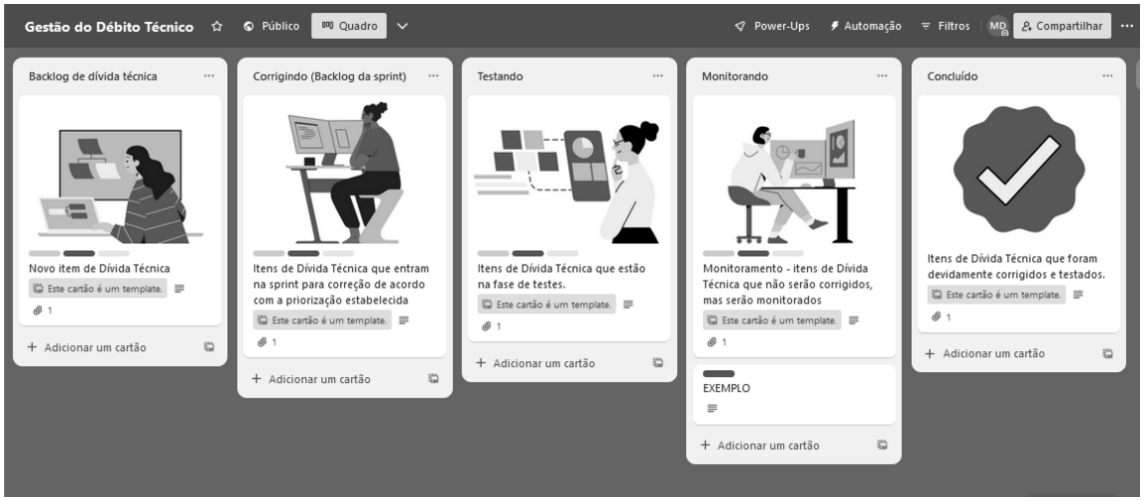

Figura 6. Quadro Kanban

tivo facilitar a adaptação e ajustes aos processos já existentes no time, visto que pode ser utilizado separadamente ou em conjunto com o backlog da sprint (os itens da coluna corrigindo podem entrar no backlog da sprint). Para avaliar seu uso, foi feito um protótipo usando a ferramenta Trello [9] composto das seguintes colunas: backlog de dívida técnica, onde os novos itens de dívida técnica são registrados; corrigindo, composto dos itens escolhidos para serem corrigidos na sprint; testando, onde os itens já corrigidos são testados; monitorando, que concentra os itens para serem monitorados; e concluído, que registra os itens que já foram concluídos no projeto. A Figura 6 apresenta o modelo de quadro Kanban.

O conjunto de fórmulas para auxiliar na tomada de decisão (Artefato 4) é composto pelo cálculo do principal e estimativa dos juros, e se baseia nos trabalhos de Curtis *et al.* [Curtis et al. 2012] e Seaman e Guo [Seaman and Guo 2011]. O principal refere-se à quantidade extra de esforço para pagar os itens de dívida técnica na data atual e é calculado por item. Dados históricos do esforço para correção de um item similar, ou até a estimativa de um especialista também podem ser utilizados para calcular o valor do principal .

Segundo Curtis *et al.* [Curtis et al. 2012], o cálculo do principal é feito da seguinte forma:

$$((DT) \times (percentual\,a\,corrigir) \times (horas\,empregadas) \times (\$\,por\,hora)) \quad (1)$$

Na fórmula acima, (DT) simboliza o item de dívida técnica, (percentual a corrigir) quantifica o quanto desse item será resolvido, e ($ por hora) quantifica o custo por hora dessa correção (valor da hora de um desenvolvedor, por exemplo).

Já os juros representam as potenciais consequências de um item de dívida técnica e é composto por probabilidade dos juros e valor estimado dos juros.

$$(J = Probabilidade\,dos\,juros * valor\,estimado\,dos\,juros) \quad (2)$$

[9] https://trello.com/

A Probabilidade dos juros refere-se às chances de que o item de divida técnica possa de fato afetar e causar danos relevantes ao projeto. Para o seu cálculo, é necessário ser mais preciso, então, de acordo com Seaman e Guo [Seaman and Guo 2011], pode-se tomar os seguintes valores: 0.8 para alta probabilidade, 0.5 para média probabilidade, 0.2 para baixa probabilidade.

O valor Estimado dos Juros é uma estimativa do trabalho ($/ horas) extra que será necessário se esse item da dívida não for resolvido.

$$(Valor\,estimado\,de\,juros\ =\ W\ *\ C) \tag{3}$$

Onde, W refere-se à severidade da penalidade e tem valor baseado em estimativa (em termos de ”alto”, ”médio”ou ”baixo”). Exemplos: 0.8 - 0.9 para alto, 0.5 para médio e 0.1 para baixo. C é o esforço médio ($/ horas) das últimas N modificações no item ou similiar (análise de dados históricos, feita pelo gestor), e funcionará como um fator de ponderação.

O manual visual (artefato 5) foi desenvolvido para auxiliar a aplicação das fórmulas, utilização dos demais artefatos que compõem o framework e apresenta um jeito prático de como gerenciar a dívida técnica . Uma página do manual é apresentada abaixo na Figura 7.

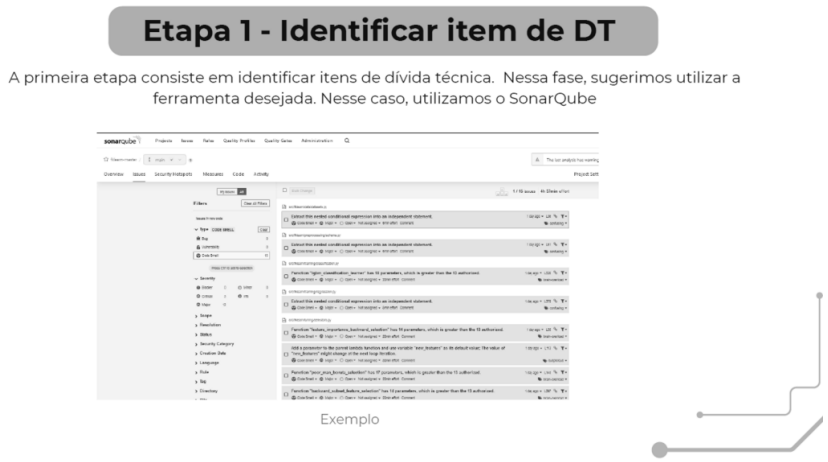

Figura 7. Manual visual

Em suma, no fluxo principal (artefato 1) do framework um novo item é identificado, registrado no cartão para o registro (artefato 2), mensurado através do cálculo do principal e do juros (artefato 4), e monitorado, ou seja, inserido no backlog de dévida técnica no quadro Kanban (artefato 3). Em seguida, acontece a tomada de decisão, se o item que representa a dívida será pago ou não, analisando se os juros são maiores que o principal. Se a resposta for positiva, há o pagamento da dívida técnica, isto é, os problemas decorrentes da atividade que não foi executada corretamente são resolvidos. Na sequência, o item é avaliado nos testes, se aprovado passa para o status de pago e vão para a coluna concluído. Caso a decisão seja não pagar a dívida técnica, ela passa para o status de admitida e será monitorada, posteriormente remensurada e reavaliada. A identificação

de itens de dívida técnica pode ser feita através de revisão de código, ferramentas automatizadas de análise estática do código, uso de *check list* de boas práticas, detecção manual através do feedback da equipe de desenvolvimento ou da equipe de qualidade, entre outros.

A avaliação (Etapa 5 do Design Science Research) trouxe resultados positivos. A primeira fase foi realizada antes da aplicação do framework e consistiu na aplicação de um questionário onde foi possível inferir que o entrevistado considerava seu conhecimento prévio sobre dívida técnica razoável, assim como a aplicação de técnicas de gerenciamento nas práticas da empresa. Porém, ao ser solicitado para avaliar o cálculo dos custos do retrabalho decorrente da dívida técnica, o entrevistado considerou as práticas aplicadas na empresa como insuficientes.

De acordo com os resultados obtidos no questionário respondido depois da aplicação do framework, a utilização do TechDebt Tracker incrementou o conhecimento sobre dívida técnica na equipe, o monitoramento da dívida técnica, a priorização de pagamento e o cálculo do custo para correção. Ainda de acordo com a avaliação, os artefatos propostos foram considerados relevantes para o gerenciamento de dívida técnica. O framework também foi avaliado como simples de ser entendido, executado e relevante para o suporte à comunicação entre os stakeholders. Além disso, o entrevistado sugeriu como melhoria, incluir aspectos do *roadmap* do produto na etapa da tomada de decisão.

Foram identificadas duas ameaças à validade do trabalho. Ambas externas e dizem respeito ao participante do estudo e influenciam na possibilidade de generalização dos resultados obtidos. A primeira refere-se à seleção, visto que somente um participante foi selecionado para participação na avaliação em pequena escala, o que pode ser considerado um número baixo. Por conta disso, os resultados não podem ser generalizados, mas continuam pertinentes. A segunda ameaça está relacionada ao grau de instrução do entrevistado, doutorando em ciência da computação. Isso impede que os resultados alcançados sejam generalizados para populações com diferentes níveis de de educação. Porém, também não os invalidam. Em ambos os casos, os resultados sinalizam positivamente sobre avaliação em pequena escala, utilizada para validar o trabalho.

5. Trabalhos relacionados

Não identificamos trabalhos que realizem o gerenciamento de dívida técnica usando o método ágil Scrum e elementos visuais aplicados a *startups* de software.

A pesquisa de Oliveira [Oliveira 2015] teve como objetivo avaliar a aplicação do *Framework* de gerenciamento de dívida técnina elaborado por Seaman e Guo [Seaman and Guo 2011]. Para tal, foi realizada uma pesquisa-ação em dois projetos de software que utilizam *Scrum*. O trabalho se mostrou relevante a partir do momento que apresentou uma experiência real da implementação do *framework* e conseguiu identificar pontos de sucesso e melhorias e recomendou adaptações do mesmo.

Santos [Santos et al. 2016], em seu trabalho, elaborou uma proposta de extensão do *Scrum* para apoiar a gestão da dívida técnica, além de uma avaliação do mesmo. A proposta foi desenvolvida baseada inicialmente em evidências da literatura (métodos secundários), assim como na experiência de profissionais da área (pesquisa de campo). Esta pesquisa pode funcionar também como referencial teórico para quem busca eficiência na gestão da dívida técnica.

Detofeno *et al.* [Detofeno et al. 2022] propõe um método de priorização de dívida técnica usando o código-fonte do projeto, as necessidades do time e as prioridadas do negócio. Assim, identifica o código mais relevante a ser priorizado para pagamento de dívida técnica. No entanto, a priori, não apresenta integração com o método Scrum e não aborda recursos visuais para facilitar a comunicação com o time. Além disso, apresenta um perfil de complexidade mais adotado por empresas maiores e mais experientes que startups.

O estudo de Chicote [Chicote 2017] apresentou uma técnica de gestão de dívida técnica baseada na utilização de elementos gráficos para organizar informações e ideias com o objetivo de trazer visibilidade ao processo e dar suporte à gestão de dívida técnica por startups em estágio inicial. A técnica apresentada consiste em utilizar fita adesiva cortada com tamanho proporcional ao da dívida técnica correspondente.

A pesquisa de Pacheco *et al.* [Pacheco et al. 2018], propõe o design e a implementação de visualizações de dívida técnica que têm como objetivo prover comunicação entre os *stakeholders* e assim dar suporte à tomada de decisão. É um trabalho focado em visualização através de gráficos, árvore, diagrama de Sankey e etc. Porém, não aborda métodos para priorização de dívida técnica.

Já o trabalho de Freire *et al.* [Freire et al. 2023] avalia a utilização de IDEA diagramas como suporte para atividades de gestão de dívida técnica. Os diagramas IDEA são utilizados para analisar práticas aplicáveis e práticas a evitar no contexto da dívida técnica. Na análise dos diagramas sob o ponto de vista dos profissionais de software no contexto de projetos de desenvolvimento de software, foram convidados profissionais, em sua maioria, de empresas de médio e grande porte. Faltam avaliações sobre a adoção em empresas como *startups*, como os diagramas IDEA podem ser integrados a métodos ágeis como o *Scrum*, e definição de critérios para priorização. Também é necessário definir quais práticas serão abordadas nos diagramas, visto que dívida técnica é um tema amplo, com vários tipos e práticas associadas.

Os trabalhos supracitados e a presente pesquisa tem em comum o objetivo de simplificar o gerenciamento da dívida técnica, entregando ferramentas adaptáveis à realidade das equipes. Os trabalhos de Oliveira [Oliveira 2015] e Santos [Santos et al. 2016] serviram de base para o uso do backlog de dívida técnica no *Scrum*. Já os estudos de Chicote e de Pacheco, assim como essa pesquisa, utilizam elementos visuais para facilitar a compreensão, a comunicação entre os envolvidos, e consequentemente a gestão da dívida técnica.

6. Conclusões

Este trabalho apresentou uma proposta de framework simplificado para gerenciamento de dívida técnica por startups. Para o desenvolvimento da mesma, foi utilizado o método *Design Science Research*, e o resultado foi avaliado através de uma avaliação em pequena escala com uma empresa. Segundo os resultados da avaliação, o TechDebt Tracker mostrou-se eficaz quanto ao que foi proposto e deu suporte quanto ao gerenciamento de dívida técnica. Para além disso, a proposta também auxiliou na comunicação entre os stakeholders, e na difusão de conhecimento sobre dívida técnica no time.

Esses resultados tem implicações relevantes para a construção e propagação de

conhecimento sobre dívida técnica, principalmente dentro do escopo das startups, visto que em suas fases iniciais, a maioria é composta por times ainda imaturos.

As limitações dessa pesquisa incluem a escala reduzida da avaliação, e a amostragem enxuta de participantes. Como trabalhos futuros, sugerimos a aplicação do framework em mais startups, assim como em times como diferentes graus de maturidade. Da mesma forma, sugerimos também testar outras fórmulas para auxiliar no cálculo da dívida técnica e na tomada de decisão; e assim avaliar a eficácia das diferentes práticas.

7. Anexos

O manual visual completo desenvolvido pode ser acessado no seguinte link: `https://drive.google.com/file/d/1s1TCNvQXTMjmZcZQwhrgUpge1uz7B8dh/view?usp=sharing`.

8. Referências

Referências

Ampatzoglou, A., Ampatzoglou, A., Avgeriou, P., and Chatzigeorgiou, A. (2015). Establishing a framework for managing interest in technical debt. In *5th International Symposium on Business Modeling and Software Design, BMSD*.

Chicote, M. (2017). Startups and technical debt: managing technical debt with visual thinking. In *2017 IEEE/ACM 1st International Workshop on Software Engineering for Startups (SoftStart)*, pages 10–11. IEEE.

Curtis, B., Sappidi, J., and Szynkarski, A. (2012). Estimating the principal of an application's technical debt. *IEEE software*, 29(6):34–42.

Dawes, J. (2008). Do data characteristics change according to the number of scale points used? an experiment using 5-point, 7-point and 10-point scales. *International journal of market research*, 50(1):61–104.

De Almeida, R. R., do Nascimento Ribeiro, R., Treude, C., and Kulesza, U. (2021). Business-driven technical debt prioritization: An industrial case study. In *2021 IEEE/ACM International Conference on Technical Debt (TechDebt)*, pages 74–83. IEEE.

Detofeno, T., Malucelli, A., and Reinehr, S. (2022). Priortd: A method for prioritization technical debt. In *Proceedings of the XXXVI Brazilian Symposium on Software Engineering*, pages 230–240.

Dresch, A., Lacerda, D. P., and Junior, J. A. V. A. (2020). *Design science research: método de pesquisa para avanço da ciência e tecnologia*. Bookman Editora.

Freire, S., Rocha, V., Mendonça, M., Izurieta, C., Seaman, C., and Spínola, R. (2023). Assessing idea diagrams for supporting analysis of capabilities and issues in technical debt management. In *International Conference on Product-Focused Software Process Improvement*, pages 243–258. Springer.

Giardino, C., Paternoster, N., Unterkalmsteiner, M., Gorschek, T., and Abrahamsson, P. (2015). Software development in startup companies: the greenfield startup model. *IEEE Transactions on Software Engineering*, 42(6):585–604.

Mendes, J. G. P. (2020). Uma análise sobre o uso de metodologias ágeis de desenvolvimento de software em startups.

Murray, A. and Scuotto, V. (2015). The business model canvas. *Symphonya. Emerging Issues in Management*, pages 94–109.

Oliveira, F. (2015). *Gerenciamento de dívida técnica em projetos de software utilizando Scrum: Uma pesquisa-ação*. Novas Edições Acadêmicas.

Opelt, A., Gloger, B., Pfarl, W., and Mittermayr, R. (2023). *Agile contracts creating and managing successful projects with Scrum*. Wiley Online Library.

Pacheco, A., Marín-Raventós, G., and López, G. (2018). Designing a technical debt visualization tool to improve stakeholder communication in the decision-making process: A case study. In *Research and Practical Issues of Enterprise Information Systems: 12th IFIP WG 8.9 Working Conference, CONFENIS 2018, Held at the 24th IFIP World Computer Congress, WCC 2018, Poznan, Poland, September 18–19, 2018, Proceedings 12*, pages 15–26. Springer.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77.

Pompermaier, L. B. (2021). Modelo press: evoluindo a adoção de práticas de engenharia de software em startups digitais.

Prestes, M. P. et al. (2020). Estudo exploratório sobre design thinking no desenvolvimento de software.

Rubin, K. S. (2018). *Scrum Essencial: Um guia prático para o mais popular processo ágil*. Alta Books Editora.

Ruschel, B. (2019). *Guia Prático do Design Thinking: Aprenda 50 ferramentas para criar produtos e serviços inovadores.* Ebook Kindle.

Santos, C. G. d. et al. (2016). Um estudo empírico sobre a gerência de dívida técnica em projetos de desenvolvimento de software que utilizam scrum.

Sbrocco, J. and Macedo, P. C. d. (2012). Metodologias ágeis: engenharia de software sob medida. *São Paulo: Érica*, 8:9.

Seaman, C. and Guo, Y. (2011). Measuring and monitoring technical debt. In *Advances in Computers*, volume 82, pages 25–46. Elsevier.

Silva, P. d. J. (2012). *Modelos de negócio para startups: o complemento value proposition canvas na metodologia business model canvas*. PhD thesis, Instituto Superior de Economia e Gestão.

Valente, M. T. (2020). *Engenharia de software moderna*. independente.

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.

Wohlin, C. and Rainer, A. (2022). Is it a case study?—a critical analysis and guidance. *Journal of Systems and Software*, 192:111395.