



MESTRADO EM SISTEMAS E COMPUTAÇÃO

ADRIANO ROCHA BORGES

**UMA INVESTIGAÇÃO SOBRE O IMPACTO DA DÍVIDA TÉCNICA EM
ATIVIDADES DE DESENVOLVIMENTO DE SOFTWARE**

Salvador
2022

ADRIANO ROCHA BORGES

**UMA INVESTIGAÇÃO SOBRE O IMPACTO DA DÍVIDA TÉCNICA EM
ATIVIDADES DE DESENVOLVIMENTO DE SOFTWARE**

Dissertação apresentada ao Programa de Pós-Graduação, Mestrado em Sistemas e Computação da Universidade Salvador – UNIFACS, como requisito parcial para obtenção do grau de Mestre.

Orientador: Prof. Dr. Rodrigo Oliveira Spínola

Salvador
2022

Ficha Catalográfica elaborada pelo Sistema de Bibliotecas da Universidade Salvador
UNIFACS.

Borges, Adriano Rocha.

Uma investigação sobre o impacto da dívida técnica em atividades de desenvolvimento de software / A Adriano Rocha Borges. Salvador: UNIFACS, 2022.

56 f.: il.

Dissertação apresentada ao Programa de Pós-Graduação, Mestrado em Sistemas e Computação da Universidade Salvador - UNIFACS, como requisito parcial para obtenção do título de Mestre.

Orientador Prof. Dr. Rodrigo Oliveira Spínola.

1. Ciência da computação. 2. Estudo de viabilidade. 3. Engenharia de software I. Spínola, Rodrigo Oliveira, orient. II. Título.

CDD:004

ADRIANO ROCHA BORGES

UMA INVESTIGAÇÃO SOBRE O IMPACTO DA DÍVIDA TÉCNICA EM ATIVIDADES
DE DESENVOLVIMENTO DE SOFTWARE

Dissertação apresentada ao Programa de Pós-Graduação, Mestrado em Sistemas e Computação da Universidade Salvador - UNIFACS, como requisito parcial para obtenção do título de Mestre e aprovada pela seguinte banca examinadora:

Rodrigo Oliveira Spínola – Orientador _____
Doutor em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro
- UFRJ
Universidade Salvador UNIFACS

Artur Henrique Kronbauer _____
PHD – Universidade Federal da Bahia (UFBA)
Universidade Salvador UNIFACS

Thiago Souto Mendes _____
Doutor em Ciência da Computação pela Universidade Federal da Bahia - UFBA
Instituto Federal da Bahia – IFBA

Salvador, 24 de maio de 2022.

A todos que direta ou indiretamente
contribuíram para realização deste trabalho.

AGRADECIMENTOS

A todos que participaram e contribuíram de alguma forma para realização deste trabalho, sem vocês nada disso seria possível.

À minha família pela torcida, pelo apoio e pelo carinho. A meus amigos, por acreditarem no meu potencial, pelos conhecimentos compartilhados e pelo incentivo. Ao meu orientador, Rodrigo Spínola, pela oportunidade, pela competência, pelas orientações e incentivo para fazer um trabalho cada vez melhor. Obrigado.

E a Deus por me permitir chegar até aqui e por ter colocado tantas pessoas especiais em minha vida.

RESUMO

O termo Dívida Técnica (DT) é utilizado para descrever o aumento de produtividade e a redução de custos de curto prazo decorrentes de atalhos tomados no desenvolvimento e evolução do *software*. É comum que um projeto de *software* incorra em dívidas durante o seu desenvolvimento. A incidência da DT pode contribuir para o sucesso do projeto, no entanto, se não gerida adequadamente, a dívida acumulada traz riscos associados podendo causar, entre outros, atrasos inesperados na evolução do sistema e dificuldade em atingir os critérios de qualidade definidos para o projeto. A gestão da DT busca implantar ações para encontrar um equilíbrio entre os benefícios de incorrer nela e os efeitos de sua presença. Gerenciar DT envolve tomar decisões relacionadas a se um item da dívida deve ser pago e o momento mais adequado para fazê-lo. Também inclui ações preventivas, uma vez que evitar itens de dívida em estágios iniciais do desenvolvimento pode reduzir as chances desses itens impactarem outras atividades de desenvolvimento. Conhecer as causas da DT pode apoiar equipes de desenvolvimento na definição de ações preventivas para a DT. Ter informações sobre os efeitos da DT pode ajudar na priorização de itens de DT a serem eliminados, apoiando uma análise mais precisa, análise de impacto e a identificação de ações corretivas para minimizar possíveis consequências negativas para o projeto. Até o momento, grande parte dos estudos a respeito da DT tem sua atenção voltada a discussões relacionadas à DT em nível de código. Esse é um cenário preocupante porque outros tipos de dívida (documentação, requisitos, testes, fatores humanos) também podem trazer consequências impactantes, ou ainda piores, em projetos. Este trabalho aborda a necessidade crescente de expandir a pesquisa sobre DT para outras áreas de desenvolvimento de software. Para isso, são analisados seis elementos relacionados à gestão da DT: causas, efeitos, práticas de prevenção e razões para a não prevenção, práticas de pagamento e motivos do não pagamento. Foi utilizado um subconjunto dos dados coletados pelo projeto InsignTD, uma família de pesquisadores, distribuídas globalmente sobre as causas, efeitos e gestão da DT. O subconjunto consiste em dados de seis replicações, totalizando 653 respostas. No geral, os resultados indicaram que as questões de codificação relacionadas às causas, efeitos, prevenção, não prevenção, pagamento e não pagamento da DT são apenas uma pequena parte das preocupações que os profissionais enfrentam na presença da DT. Na verdade, a DT é muito comumente encontrada em outras questões do desenvolvimento de software.

Palavras-chaves: Dívida técnica, causas da dívida técnica, efeitos da dívida técnica, gestão da dívida técnica.

ABSTRACT

The term Technical Debt (TD) is used to describe the increase in productivity and the reduction of long-term costs resulting from shortcuts taken in the development and evolution of the software. It is common for a software project to incur debt during its development. The incidence of TD can contribute to the success of the project more quickly, however, if not managed efficiently, the accumulated debt yields associated risks, among others, unexpected delays in the evolution of the system and difficulty in meeting the quality criteria. defined for the project. TD management can implement actions to find a balance between the benefits of incurring it and the effects of its presence. Managing TD involves making decisions regarding whether an item of debt should be repaid and the most appropriate time to do so. It also includes preventive actions such as avoiding debt items in early stages of software development can reduce the chances of these items impacting development activities later on. Research on TD management related to understanding the causes that lead development teams to incur debt items in their projects and their effects were also done. Knowing the causes of TD can support development teams in defining and creating preventive actions for TD. Having information about the effects of TD can help in prioritizing TD items to pay, supporting more accurate analysis, impact analysis, and the identification of corrective actions to minimize potential negative consequences for the project. Until now, most studies on TD have focused on discussions related to TD with coding problems. This is a worrying scenario because other types of debt (documentation, requirements, tests, human factors) can also have impactful, or even worse, consequences on projects. This work addresses the growing need to expand TD research to other areas of software development. For this, six elements related to the management of TD were analyzed: causes, effects, prevention practices and reasons for non-prevention, reimbursement practices and reasons for non-reimbursement. A subset of the data collected by the InsignTD project, a family of globally distributed research into the causes, effects and management of TD, was used. The subset consists of data from six replications, totaling 653 practitioner software responses. Overall, the results indicated that coding issues related to the causes, effects, prevention, non-prevention, payment and non-payment of TD are only a small part of the concerns that professionals face in the presence of TD. In fact, DT is very commonly found in other software development issues.

Keywords: Technical debt, technical debt causes, technical debt effects, technical debt management.

LISTA DE FIGURAS

Figura 1 - Os 10 efeitos mais comumente citados da DT por frequência global	22
Figura 2 - Distribuição de respostas do participante	46
Figura 3 - O diagrama de corcunda para elementos de gerenciamento de TD e seus problemas de desenvolvimento de software	47

LISTA DE TABELAS

Tabela 1 - Subconjunto do questionário utilizado	32
Tabela 2 - As 10 causas mais citadas relacionadas à codificação e outros problemas de desenvolvimento	37
Tabela 3 - Categorias de causas não relacionadas à codificação	38
Tabela 4 - Os 10 efeitos mais citados relacionados à codificação e outras questões de desenvolvimento	39
Tabela 5 - Categorias de efeitos não relacionados à codificação	40
Tabela 6 - As 10 práticas de prevenção de DT mais comumente citadas relacionadas à codificação ou outras questões de desenvolvimento	41
Tabela 7 - Categorias de práticas de prevenção não relacionadas a código	42
Tabela 8 - As 10 razões mais citadas para não prevenir a DT	42
Tabela 9 - Categorias de motivos para a não prevenção do DT não relacionadas à codificação	43
Tabela 10 - As 10 práticas de pagamento da DT mais comumente citadas	44
Tabela 11 - Categorias de práticas de pagamento não relacionadas ao código	44
Tabela 12 - As 10 principais razões para não pagar a DT	45
Tabela 13 - Categorias de razões para o não pagamento da TD não relacionadas à codificação	45

LISTA DE SIGLAS

DT	Dívida Técnica
GQM	<i>Goal – Question – Metric</i>
IDE	<i>Integrated Development Environment</i>
MVC	<i>Model–view–controller</i>
PLSQL	<i>Procedural Language/Structured Query Language</i>
SPA	<i>Single Page Application</i>
SQALE	<i>Software Quality Assessment based on Lifecycle Expectations</i>
TAM	<i>Technology Acceptance Model</i>
TDMF	<i>Technical Debt Management Framework</i>
URL	<i>Uniform Resource Locator</i>
XML	<i>eXtensible Markup Language</i>
XSL	<i>eXtensible Stylesheet Language</i>

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	14
1.1 INTRODUÇÃO	14
1.2 CONTEXTO E MOTIVAÇÃO	15
1.3 OBJETIVOS E QUESTÕES DE PESQUISA	16
1.4 METODOLOGIA	17
1.5 ORGANIZAÇÃO DA DISSERTAÇÃO	18
CAPÍTULO 2 – REFERENCIAL TEÓRICO	19
2.1 DÍVIDA TÉCNICA	19
2.2 CAUSAS DA DÍVIDA TÉCNICA	20
2.3 EFEITOS DA DÍVIDA TÉCNICA	21
2.4 GERENCIAMENTO DA DÍVIDA TÉCNICA	22
2.4.1 Prevenção da Dívida Técnica	23
2.4.2 Pagamento da Dívida Técnica	25
2.5 CONSIDERAÇÕES FINAIS	27
CAPÍTULO 3 – O PROJETO INSIGHTD	28
3.1 INTRODUÇÃO	28
3.2 O PROJETO INSIGHTD	28
3.2.1 Estado Atual	30
3.3 METODOLOGIA	31
3.3.1 Coleta de Dados	32
3.3.2 Procedimentos de Análise de Dados	33
3.3.2.1 Etapa 1 – Dados demográficos	33
3.3.2.2 Etapa 2 – Preparação de dados para análise	33
3.3.2.3 Etapa 3 Erro! Indicador não definido. Classificação e análise de dados	34
3.4 CONSIDERAÇÕES FINAIS	34
CAPÍTULO 4 – RESULTADOS	36
4.1 INTRODUÇÃO	36
4.2 RESULTADOS	37
4.2.1 Dados demográficos	37
4.2.2 QP1: As causas da DT estão mais relacionadas aos problemas de codificação ou a outros problemas de desenvolvimento de software?	38
4.2.3 QP2: Os efeitos da DT são mais percebidos na codificação ou em outras questões do processo de desenvolvimento de software?	40
4.2.4 QP3: A prevenção da DT está mais relacionada aos problemas de codificação ou a outras questões no processo de desenvolvimento de software?	41

4.2.5	QP4: As razões para não prevenir a DT estão mais relacionadas aos problemas de codificação ou a outros problemas de desenvolvimento?	43
4.2.6	QP5: O pagamento da DT está mais associado a problemas de codificação ou a outras questões no processo de desenvolvimento de software	44
4.2.7	QP6:As razões para não pagar a DT estão mais relacionadas às questões de codificação ou a outras questões de desenvolvimento?	46
4.3	DISCUSSÃO	47
4.4	AMEAÇAS À VALIDADE	49
4.5	CONSIDERAÇÕES FINAIS	50
CAPÍTULO 5 – CONSIDERAÇÕES FINAIS		51
5.1	CONSIDERAÇÕES FINAIS	51
5.2	RESULTADOS	52
5.3	CONTRIBUIÇÕES	54
5.4	TRABALHOS FUTUROS	54
REFERÊNCIAS		55

CAPÍTULO 1 – INTRODUÇÃO

Neste capítulo será apresentado o contexto do trabalho, o que motivou essa pesquisa e os objetivos. Também serão apresentados a metodologia de pesquisa utilizada, o histórico do trabalho realizado e como este texto está estruturado.

1.1 INTRODUÇÃO

À medida que projetos de software passam por atividades de evolução, algumas atividades inerentes ao seu desenvolvimento são penalizadas como, por exemplo, definição de requisitos, documentação, arquitetura, uso de boas práticas e manutenção de *software*. Logo, ações corretivas são requeridas para que não haja diminuição da qualidade do *software* durante o seu ciclo de vida (LIENTZ *et al.*, 1978) (LEHMAN; BELADY, 1985) (PARNAS, 1994).

Questões como essas podem ser analisadas utilizando o conceito da Dívida Técnica (DT). DT descreve os problemas de qualidade interna do produto enfrentados por profissionais da indústria de software ao tomar atalhos durante o processo de desenvolvimento com o objetivo de aumentar a produtividade e reduzir o tempo da entrega ao cliente (Izurieta *et al.* 2012). Por outro lado, se não gerenciados, os itens de dívida geram riscos associados, entre outros, a atrasos inesperados na evolução do sistema e dificuldade em atingir os critérios de qualidade definidos para o projeto (KRUCHTEN *et al.*, 2012) (ZAZWORKA *et al.*, 2013) (GUO *et al.*, 2014). O uso do conceito de DT facilita o entendimento a respeito do dinamismo que existe dentro de projetos de desenvolvimento de software (FOWLER, 2003), facilitando o entendimento dos seus conceitos (SPÍNOLA *et al.*, 2013).

O termo “*going into debt*” foi citado em (CUNNINGHAM, 1992) para esclarecer que a DT, independentemente do seu tamanho, pode acelerar o desenvolvimento de um software, porém, a cada minuto adicional investido na codificação de baixa qualidade, juros são incididos nessa dívida. Estudos realizados sobre DT têm sido realizados em maior quantidade e abrangência (ALVES *et al.*, 2016). Com os estudos mais recentes da área, a atenção começa a ser dada à DT presente em outros artefatos gerados durante o projeto de desenvolvimento do software, como documentação, requisitos e testes.

1.2 CONTEXTO E MOTIVAÇÃO

A DT representa decisões técnicas que podem trazer benefícios de curto prazo (por exemplo, maior produtividade e custos mais baixos), mas que podem impactar negativamente os projetos de software a longo prazo (IZURIETA *et al.* 2012). É comum que surjam itens de dívida no decorrer de um projeto de desenvolvimento de software (BROWN *et al.* 2010). Por isso, é importante realizar seu monitoramento e gerenciamento. Quando não gerenciada, a DT pode acumular à medida que o projeto vai sendo executado, gerando riscos, com impactos ligados ao tempo necessário para evolução do sistema e dificuldade em atingir os critérios de qualidade estabelecidos no início do projeto (RIOS *et al.* 2020).

O gerenciamento da DT, quando realizado de forma eficaz, pode levar a um equilíbrio entre os benefícios proporcionados pela ocorrência da dívida e os impactos negativos de sua presença. Gerenciar DT é decidir se um item de dívida será pago e qual o momento ideal para que o pagamento seja realizado (LI *et al.*). O monitoramento, a identificação, a prevenção e o pagamento da DT apoiam os gestores a tomar decisões baseadas em fatos, com uma boa resposta na qualidade do software e maior eficácia no andamento das atividades de desenvolvimento (GUO *et al.*, 2014).

Antes de adotar ações de gestão da DT, é necessário entender quais tipos de dívida ocorrem, quais técnicas de gerenciamento podem ser aplicadas, quais são as causas da DT e seus efeitos, quais as possíveis formas de pagamento e quais os motivos que levam os profissionais a não realizar o pagamento da dívida. A disposição dessas informações pode facilitar a realização das atividades de gestão, pois os profissionais poderão analisar como outras equipes lidam com itens da DT presentes em seus projetos.

Apesar do número de estudos sobre DT estar em crescimento, ainda existe uma grande concentração de pesquisas observando a DT do ponto de vista do código fonte (ALVES *et al.* 2016). Embora inicialmente a DT estivesse associada apenas a problemas em nível de código, ela também pode impactar outros artefatos do processo de desenvolvimento de software, como documentação de requisitos e casos e procedimentos de teste (ALVES *et al.* 2016). Considerar apenas a perspectiva de código é uma limitação que pode trazer riscos para projetos de software, uma vez que itens de dívida podem afetar outros artefatos produzidos durante todo o processo de desenvolvimento. Este cenário foi a motivação para o desenvolvimento deste trabalho: investigar se os elementos de gerenciamento da DT (causas, efeitos, prevenção e pagamento)

estão mais relacionados a questões de codificação ou a outras questões do desenvolvimento de software.

1.3. OBJETIVO E QUESTÕES DE PESQUISA

A maioria dos trabalhos sobre DT foca na análise da dívida em nível de código, porém a DT pode incorrer em outros artefatos, gerando outros tipos de dívida que, se não tratados, podem trazer consequências para os projetos. É necessário entender como os profissionais da indústria de software se comportam em seus projetos ao encontrar uma dívida que não seja de código, através da perspectiva de causas, efeitos, prevenção e pagamento da DT.

Este trabalho tem como objetivo investigar se os elementos de gerenciamento da DT (causas, efeitos, prevenção e pagamento) estão mais relacionados a questões de codificação ou a outras questões do desenvolvimento de *software*. Dessa forma, busca-se esclarecer a importância de outras questões de desenvolvimento de software ao lidar com a DT, de acordo com o estado da prática. Para alcançar esse objetivo, foram definidas as seguintes questões de pesquisa (QPs):

QP1: As causas da DT estão mais relacionadas aos problemas de codificação ou a outros problemas de desenvolvimento de software? Esta questão investiga se os problemas de codificação são mais prováveis de ser uma fonte de DT do que outras questões do desenvolvimento de software, como processo e teste.

QP2: Os efeitos da DT são mais percebidos na codificação ou em outras questões do processo de desenvolvimento de software? Por meio desta questão, pretende-se verificar se os efeitos do DT são mais percebidos em questões de codificação ou em outras questões de desenvolvimento.

QP3: A prevenção DT está mais relacionada aos problemas de codificação ou a outras questões no processo de desenvolvimento de software? O objetivo desta questão é investigar se as práticas utilizadas pelas equipes de desenvolvimento para evitar itens da DT estão mais relacionadas a questões de codificação ou a outras questões do desenvolvimento.

QP4: As razões para não prevenir a DT estão mais relacionadas aos problemas de codificação ou a outros problemas de desenvolvimento? O objetivo desta questão é investigar se as razões que impedem as equipes de desenvolvimento de não aplicar práticas preventivas estão mais relacionadas aos problemas de codificação ou a outros problemas de desenvolvimento de software.

QP5: O pagamento DT está mais associado a problemas de codificação ou a outras questões no processo de desenvolvimento de software? Por meio dessa pergunta, investiga-se se as práticas usadas para eliminar itens de dívida estão mais associadas às questões de codificação ou a outras questões de desenvolvimento de software.

QP6: As razões para não pagar a DT estão mais relacionadas às questões de codificação ou a outras questões de desenvolvimento? Aqui, investiga-se se as razões para não aplicar as práticas de pagamento DT estão mais associadas aos problemas de codificação ou a outros problemas de desenvolvimento de software.

1.4 METODOLOGIA

Este trabalho está contextualizado no projeto InsignTD (RIOS *et al.*, 2020b). InsignTD é uma família de *surveys* globalmente distribuída sobre as causas, os efeitos e o gerenciamento da DT (RIOS *et al.*, 2018b; RIOS *et al.*, 2020b). O projeto foi organizado com o objetivo de replicar um questionário em diferentes países, buscando resultados generalizáveis a respeito das práticas e as tendências da indústria na área da DT, incluindo as causas que levam a ocorrência da DT, os efeitos da sua existência e como as equipes reagem quando se deparam com a presença de itens da dívida (RIOS *et al.*, 2020b). O projeto organiza dados sobre os problemas práticos da DT considerando diferentes contextos do desenvolvimento de software, diferentes culturas de desenvolvimento, tamanhos de organizações e metodologias de desenvolvimento. Atualmente, pesquisadores de 12 países participam do projeto: Arábia Saudita, Argentina, Brasil, Chile, Colômbia, Costa Rica, Estados Unidos, Finlândia, Índia, Itália, Noruega e Sérvia. Nesta dissertação, foi utilizado o conjunto de dados proveniente da replicação de InsignTD no Brasil, Chile, Colômbia, Costa Rica, Estados Unidos e Sérvia.

As atividades listadas a seguir compõem a metodologia seguida para o desenvolvimento desta dissertação:

1 Revisão da Literatura: nessa atividade, foi realizada uma revisão da literatura para entendimento dos conceitos necessários para o desenvolvimento desta pesquisa;

2 Consolidação de dados do InsignTD: nessa atividade, as variáveis base para a realização do trabalho foram definidas (prevenção, causas, efeitos e pagamento). As respostas do *survey* relacionadas às variáveis definidas foram consolidadas e codificadas, classificando itens relacionados a código e itens relacionados a outras atividades do desenvolvimento de software;

3 Análise dos Dados: nessa atividade, análises foram realizadas com o objetivo de responder às questões de pesquisa consideradas nesta dissertação;

4 Disseminação dos Resultados: nessa atividade, os resultados obtidos foram disseminados em conferências e periódicos da área.

1.5 ORGANIZAÇÃO DA DISSERTAÇÃO

Este capítulo apresentou a motivação para o desenvolvimento desta dissertação, seu objetivo e a metodologia de pesquisa utilizada. A estrutura desta dissertação segue a cronologia em que as atividades de investigação foram realizadas, sendo organizada da seguinte forma:

Capítulo 2. Referencial Teórico: apresenta uma revisão bibliográfica sobre DT, suas causas, efeitos, prevenção e pagamento considerando definições importantes para o entendimento desta dissertação.

Capítulo 3. InsignTD: apresentação do projeto de pesquisa InsignTD, uma família de *surveys* globalmente distribuída cujo objetivo é identificar os motivos que levam equipes de desenvolvimento a incorrerem em DT nos projetos e os efeitos que os desenvolvedores sofrem em decorrência de sua presença.

Capítulo 4. Resultados: Apresenta os resultados obtidos a respeito da necessidade de expandir a pesquisa sobre DT para outras áreas do desenvolvimento de software, analisando os elementos relacionados à gestão da DT: causas, efeitos, práticas, razões para não prevenção, práticas de pagamento e motivos para não pagar a DT.

Capítulo 5. Considerações Finais: apresenta as considerações finais do trabalho considerando suas principais contribuições, limitações e perspectivas de trabalhos futuros.

CAPÍTULO 2 – REFERENCIAL TEÓRICO

Este capítulo apresenta os conceitos fundamentais sobre dívida técnica. Também são discutidas as causas, efeitos e gestão (prevenção e pagamento) da DT.

2.1 DÍVIDA TÉCNICA

A DT ocorre em diversos artefatos ao longo do processo de desenvolvimento de software, podendo variar de acordo com o tempo em que foi incorrida e as atividades a que está associada. No entanto, mesmo com o crescimento de pesquisas na área, muitos esforços ainda têm sido concentrados na investigação de DT em nível de código-fonte (LI et al, 2015; ALVES et al., 2016).

Li et al. (2015) classificou a DT em dez tipos. A dívida de código foi o tipo mais estudado entre os artigos primários analisados. Em outro estudo de mapeamento da área, Alves et al. (2016) relataram artefatos que têm sido frequentemente usados para identificar itens de dívida em projetos de software, constatando que o maior foco é nas estratégias de identificação de itens de dívida a partir do código-fonte. Por outro lado, outros artefatos como especificação de requisitos, documentação e relatórios de teste, entre outros, foram mencionados apenas ocasionalmente. Os autores sugeriram que o foco concentrado no código-fonte pode estar relacionado à existência de várias ferramentas que realizam análises estáticas do código, apoiando a detecção de itens de dívida.

Mais recentemente, Rios et al. (2018) atualizou a taxonomia dos tipos de dívida. Novamente, o foco das investigações também tem sido nos tipos que estão relacionados ao código-fonte. Os autores relatam que uma das possíveis explicações para isso é a influência do conceito da DT cunhado por (CUNNINGHAM, 1992), que foca especificamente nas atividades de implementação. Outra razão para essa concentração é que os tipos relacionados ao código (arquitetura, *design*, código, defeito e teste) tendem a causar efeitos que podem ser sentidos mais rapidamente pelas equipes de desenvolvimento.

No geral, ainda há uma visão concentrada da DT em nível de código. Este é um cenário preocupante, uma vez que outros tipos de dívida também podem ter consequências impactantes, ou mesmo piores, para os projetos. Assim, é necessário ir além do código-fonte e investigar outras facetas da DT. Ao investigar como os profissionais de software enfrentam a DT em seus projetos, pretende-se entender se a concentração atual de estudos é benéfica (próxima ao que

foi observado na prática) ou uma armadilha (distante do que foi observado na prática). Para tanto, a presente pesquisa será realizada sob a perspectiva de causas, efeitos, prevenção e pagamento da DT, que serão apresentados nas próximas subseções.

2.2 CAUSAS DA DÍVIDA TÉCNICA

A partir do estudo realizado pelo projeto *InsighTD*, com a aplicação do questionário na indústria de software, foi possível identificar as principais causas da ocorrência da DT (Rios et al., 2020). O conhecimento dessas causas pode auxiliar gestores e equipes a identificarem os mecanismos de prevenção da DT. O estudo permitiu reconhecer 61 causas que levam as equipes a inserir DT em projetos. As que tiveram mais citações foram prazo com 13 citações, falta de adaptação do processo ao cenário real com 7 citações, falta de maturidade para acompanhar o processo com 6 citações, esforço com 6 citações, documentação inexistente com 4 citações e alta rotatividade na equipe com 4 citações.

Como existia uma relação entre as causas, o estudo agrupou as causas nas seguintes categorias relacionadas ao desenvolvimento de software:

- Documentação: aponta as causas associadas à documentação gerada sobre o software desenvolvido. São exemplos de causas nesta categoria: falta de documentação e documentação desatualizada.
- Falta de conhecimento: está relacionada ao conhecimento técnico pertencente aos profissionais envolvidos no projeto de software. São exemplos: falta de conhecimento sobre a tecnologia.
- Fatores externos: engloba as questões relacionadas a problemas externos ao desenvolvimento e a organização. São exemplos: pressões externas e problemas com cliente.
- Metodologia: reúne questões associadas às metodologias aplicadas no projeto de desenvolvimento. Exemplos: falta de processos e alta rigidez do processo.
- Organizacional: São pontos ligados à organização, como problemas em alocação de recursos, investimentos e processos da empresa.
- Pessoas: reúne pontos ligados ao pessoal envolvido no projeto, como, por exemplo, falta de maturidade da equipe e falta de interesse do desenvolvedor.

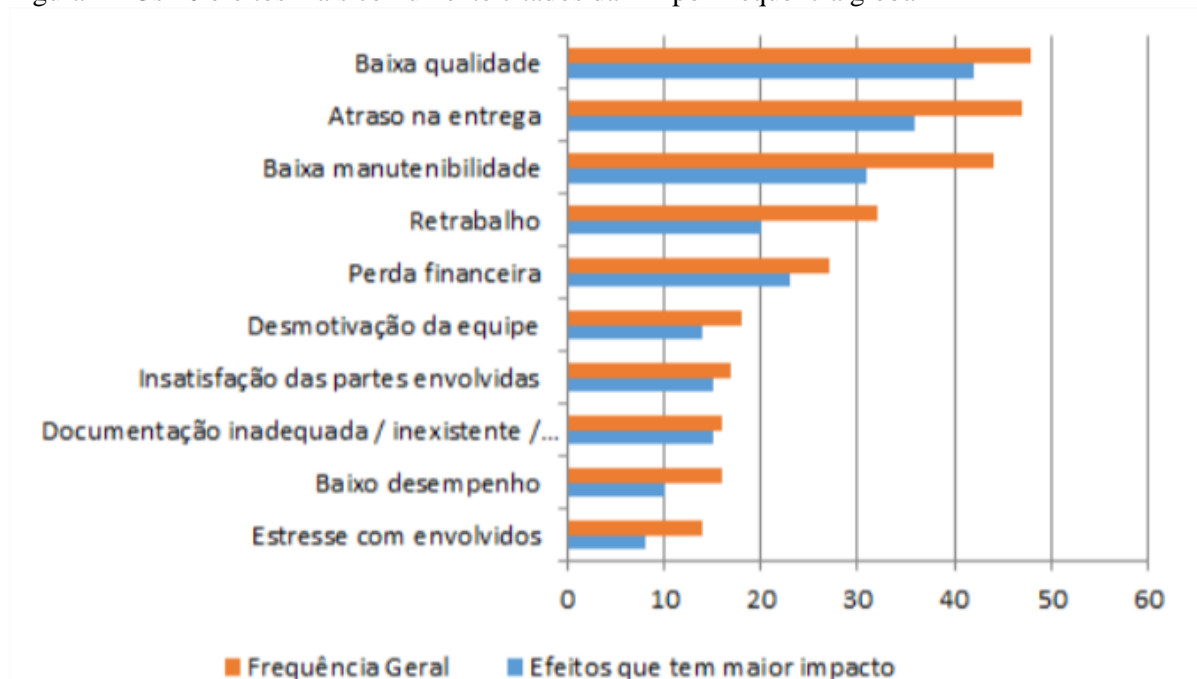
- Planejamento e gerência: são causas ligadas ao planejamento do projeto. Exemplos: prazo e escopo.
- Projeto: são questões que podem ocorrer durante o desenvolvimento do projeto, como mudanças no projeto ou design mal elaborado.

2.3 EFEITOS DA DÍVIDA TÉCNICA

O projeto InsignTD identificou um total de 66 efeitos da DT (RIOS et al., 2020). A Figura 1 apresenta a lista dos 10 efeitos mais citados juntamente com o seu impacto na visão dos participantes. Os mais comumente citados foram: baixa qualidade, atraso na entrega, baixa manutenibilidade, retrabalho, perda financeira, desmotivação da equipe, insatisfação das partes envolvidas, documentação inadequada ou inexistente, baixo desempenho e estresse com envolvidos.

Dessa lista, os efeitos mais citados e com maior impacto foram, baixa qualidade externa, atraso na entrega e baixa manutenibilidade. A baixa qualidade externa está associada à qualidade de um artefato do produto, com destaque para os erros e defeitos recorrentes que não são ajustados. São exemplos citados pelos participantes: “baixa qualidade do que foi oferecido” e “defeitos conhecidos e não corrigidos”. O atraso na entrega está ligado aos prazos não cumpridos para entrega do produto final ao cliente. A baixa manutenibilidade traz as questões encontradas durante as ações de manutenção do software, como o aumento no empenho para correção de defeitos e os desafios encontrados na evolução do software. Como é possível comprovar em “manutenção e evolução extremamente difíceis”.

Figura 1 - Os 10 efeitos mais comumente citados da DT por frequência global



Ao examinar os efeitos listados na pesquisa, os autores constataram que muitos estavam relacionados entre si, agrupando os efeitos nas seguintes categorias:

- **Questões internas de qualidade:** essa categoria contempla os efeitos ligados ao processo de manutenção do software. Foram identificados dez efeitos para essa categoria. São eles: aumento na quantidade de manutenções, baixa manutenibilidade, código mal feito e interromper atividades de desenvolvimento para pagamento de dívida;
- **Organizacional:** reúne os efeitos associados à organização. Dentre elas, destacam-se, falta de treinamento e imagem da empresa prejudicada;
- **Pessoas:** a categoria de pessoas trata dos assuntos atribuídos aos integrantes da equipe do projeto. Foram encontrados 16 itens nessa categoria, com destaque para alta rotatividade e desmotivação da equipe;
- **Planejamento e Gerência:** aponta os efeitos relacionados à gestão do projeto. Dos 11 itens identificados, se destacam estimativa de tempo imprecisa e alterações no planejamento;
- **Questões externas de qualidade:** são os efeitos percebidos sobre a qualidade durante o desenvolvimento do software. Encontramos nesta categoria efeitos como, baixa qualidade externa e falta de credibilidade do produto. Além desses, mais 8 efeitos foram incluídos nessa categoria;

- Questões de desenvolvimento: nessa categoria foram classificados os efeitos ligados às atividades de desenvolvimento do projeto. Destaque para a sobrecarga da infraestrutura e necessidade constante de reteste.

2.4 GERENCIAMENTO DA DÍVIDA TÉCNICA

O gerenciamento da DT consiste em definir mecanismos de monitoramento, prevenção e pagamento da dívida. Também contribui para a tomada de decisão a respeito da necessidade de eliminar um item da dívida e o momento ideal para tal ação (SEAMAN et al., 2011). O gerenciamento adequado da dívida permite que o projeto alcance suas metas mais cedo e de forma menos custosa, buscando a redução dos impactos negativos como um fator decisivo para o sucesso dos projetos.

Li et al. (2015) realizaram um estudo com trabalhos já publicados sobre DT para entender o estado da arte sobre o gerenciamento da DT. Os autores enumeraram oito atividades relacionadas à gestão de DT: identificação, quantificação, priorização, prevenção, monitoramento, pagamento, documentação e comunicação. Com a contribuição dos estudos realizados por (RIOS et al, 2018), as seguintes atividades foram adicionadas: visualização de TD, análise de tempo de mercado e análise de cenário.

Nas próximas subseções, as atividades de prevenção e pagamento da DT serão abordadas.

2.4.1 Prevenção da DT

Quando aplicadas à DT, as ações preventivas podem apoiar a equipe de desenvolvimento na aplicação de boas práticas que reduzem a ocorrência da dívida. Em um estudo realizado por (RIOS et al., 2018), foi investigado se a DT poderia ser prevenida. Através de um estudo de caso, fundamentado em entrevistas com dez profissionais, os autores reportaram que a dívida pode ser evitada e o ideal é investir em práticas de prevenção do que pagá-la posteriormente.

Em Freire et al. (2020), foi realizado um estudo para investigação da prevenção da dívida em projetos de software. O estudo utilizou dados oriundos do projeto *InsighTD* com o objetivo de obter informações sobre o estado da prática sobre a prevenção da DT na indústria de software. A pesquisa foi aplicada no Brasil entre dezembro de 2017 e janeiro de 2018, e sua replicação nos Estados Unidos ocorreu entre fevereiro e abril de 2019. Participou da pesquisa um grupo de 207 profissionais da indústria de software.

A princípio, foi questionado se seria possível prevenir o item específico da DT descrito pelo participante. Um total de 184 (88%) responderam que sim. O resultado é considerado um percentual relevante, pois a maioria dos participantes (82%) indicou que o item reportado pelo participante era relevante, e é um tipo de situação que acontecia com muita frequência ou ocorria de vez em quando no projeto.

Nessa mesma pesquisa, os autores tiveram a oportunidade de listar quais práticas poderiam ser adotadas para prevenir a DT. De um total de 136 práticas, as dez mais citadas foram: planejamento do projeto, adoção de boas práticas, boa definição de requisitos, criação de testes, treinamento, avaliação de código, seguir uma boa definição de processos, analisar riscos e impactos, definir boa arquitetura e realizar refatoração.

Todas as práticas foram organizadas nas seguintes categorias:

- Planejamento de projeto bem definido: agrupa as práticas associadas às atividades de planejamento do projeto. Entre elas, destacam-se o acompanhamento do planejamento do projeto, análise de riscos e impactos, prazos bem planejados, monitoramento eficaz e alocação adequada de tarefas.
- Adoção de boas práticas para desenvolvimento de software: inclui as práticas relacionadas às atividades de desenvolvimento de software, como adoção de boas práticas, documentação bem definida, arquitetura bem definida, desenho do projeto, uso adequado de padrões de projeto e uso da versão mais adequada da tecnologia
- Ter uma equipe eficaz: engloba as práticas que melhoram o conhecimento técnico e a motivação da equipe. Dentre elas, destacam-se o treinamento, boa comunicação na equipe, boa alocação de recursos na equipe, prontidão da equipe e foco;
- Controle e medição da qualidade no projeto: agrupa práticas associadas à garantia da qualidade, como: criação de testes, avaliação de código, refatoração, criação de testes automatizados e padronização de código;
- Controlar e negociar os requisitos de software: inclui práticas relacionadas às atividades de engenharia de requisitos. Por exemplo, requisitos bem definidos, boa comunicação entre stakeholders, declaração de escopo bem definida, rastreamento de mudanças de requisitos e comprometimento com o cliente;
- Seguir e melhorar um processo bem definido: organiza as práticas associadas ao processo e sua gestão, tais como: seguir o processo de projeto bem definido,

processo iterativo, flexibilização no processo definido, melhoria do processo de desenvolvimento de software e compreensão do processo de desenvolvimento seguido pela equipe;

- Identificando, gerenciando e estimando DT: contém as práticas aplicadas ao gerenciamento de DT. Dentre elas, destacam-se: a implementação de uma estratégia de pagamento de DT, monitoramento de DT, implementação de estratégia de identificação de DT e priorização de pagamento de DT.

Além disso, 24 (11%) profissionais informaram que não seria possível prevenir o item específico da DT reportados por eles, citando dezoito razões que impedem a prevenção da DT. As razões mais citadas foram: prazos curtos (com 7 citações), necessidade de redução do *time to market* (2), falta de preocupação com manutenibilidade (2), falta de conhecimento técnico (2) e falta de profissionais qualificados (2).

Ainda que a quantidade de respostas sobre a não-prevenção da DT seja relativamente pequena, os resultados revelam que a incapacidade de gerenciar prazos e questões que envolvem o nível de conhecimento da equipe dificultam a implementação de ações preventivas.

2.4.2 Pagamento da Dívida Técnica

O pagamento da dívida é uma das atividades que compõem a gestão da DT. Essa atividade fornece apoio na tomada de decisões sobre a necessidade e o momento adequado para eliminar esse item e utiliza abordagens como reengenharia e refatoração para conseguir eliminar as dívidas do projeto (GRIFFITH et al., 2014; RIOS et al., 2018).

O projeto *InsighTD* investigou o estado da prática do pagamento da DT, por meio da análise de respostas de 432 profissionais da indústria de software dos países Brasil, Chile, Colômbia e Estados Unidos (FREIRE et al., 2020). Os resultados indicaram que 258 (60%) participantes, informaram que o item da DT mencionado não foi eliminado. Essa é uma situação para ter atenção, pois 87% dos participantes, informaram que o exemplo de DT informado ocorria com muita frequência ou ocorria ocasionalmente no projeto.

O estudo identificou 34 práticas utilizadas para eliminar ou dar suporte à eliminação de itens da DT. Dentre as práticas mais citadas pelos participantes está a refatoração de código com 64 citações (26,4%), Investir esforço em atividades de pagamento DT com 21 citações (8,7%), refatoração de projeto com 16 citações (6,6%), monitoramento e controle das atividades do projeto com 12 citações (5%), priorização dos itens da DT com 11 citações (4,5%),

atualização da documentação do sistema com 9 citações (3,7%), aumento do orçamento do projeto com 8 citações (3,3%) e a realização de feedbacks em curtos períodos com 8 citações (3,3%).

Com base nas práticas identificadas, foi possível estabelecer uma relação entre elas resultando em sua organização em oito categorias:

- Questões relacionadas ao desenvolvimento: são práticas relacionadas ao processo de implementação do software, como a atualização da documentação do sistema, adoção de boas práticas e soluções de problemas técnicos;
- Infraestrutura: reúne as práticas associadas a ferramentas, tecnologias e ambientes de desenvolvimento. São elas: uso de ferramentas externas e organização do repositório de projetos;
- Questões internas de qualidade: estão as práticas aplicadas para lidar com questões de limitações que envolvem a qualidade interna do software. Para essa categoria temos a refatoração de código e a refatoração de *design*;
- Metodologia: ponto relacionado às práticas relativas aos processos executados pela equipe. Destaca-se o esforço de investimento em atividades de pagamento da DT, o esforço de investimento em atividades de teste e o uso de iterações de *feedback*;
- Organizacional: reúne as práticas ligadas às definições organizacionais, como a contratação de pessoal com especialização e a mudança na gestão do projeto;
- Pessoas: estão as práticas associadas aos membros da equipe de desenvolvimento do software. São elas: melhora na comunicação entre os membros da equipe, melhora na colaboração da equipe e comunicar-se com o cliente sobre itens de DT.
- Planejamento e gestão: aloca as práticas ligadas a rotinas da gestão. Exemplo: monitoramento e controle das atividades do projeto, priorização de itens de DT e a negociação e extensão dos prazos.

A pesquisa também demonstrou que existem empresas que não se preocupam com o pagamento da DT e apresentam 28 razões pelas quais não realizarem o pagamento. Segundo a ordem de razões mais citadas estão: baixa prioridade (19,7%), falta de interesse organizacional (16,2%), focando em objetivos de curto prazo (15,3%), custo (12,2%), falta de tempo (7,9%), decisão do cliente (4,4%), falta de recursos (3,5%), complexidade do item TD (2,2%), visão de gerenciamento insuficiente sobre o reembolso de TD (2,2%), o projeto foi descontinuado (2,2%).

Isso indica que para investir em ações de pagamento da DT, deve-se lidar diretamente com os gestores da empresa.

As razões também foram organizadas nas seguintes categorias:

- Questões de desenvolvimento: reúne as razões relacionadas às atividades de desenvolvimento do software, são elas complexidade do projeto e a decisão de não alterar o framework;
- Fatores externos: agrupa as razões associadas a fatores que fogem ao controle da equipe de desenvolvimento, como a decisão do cliente, projeto descontinuado e itens de DT que não afetam o usuário;
- Questões da qualidade interna: são as razões associadas aos atributos do código e da estrutura do sistema como complexidade do item TD e número de itens TD;
- Falta de conhecimento: reúne as razões associadas à necessidade de conhecimento técnico, como a falta de conhecimento sobre TD e falta de conhecimento técnico;
- Metodologia: lista as razões vinculadas às atividades do processo. São elas: a falta de adoção de lições aprendidas, a falta de testes e a não aplicação de ações de mitigação nas causas de DT;
- Organizacional: agrupa as razões associadas a decisões organizacionais. São elas, falta de interesse organizacional, falta de recursos e alta rotatividade da equipe;
- Pessoas: nessa categoria encontram-se visão insuficiente da administração sobre o pagamento do TD, falta de comprometimento da equipe e sobrecarga da equipe;
- Planejamento e gestão: reúne as razões a respeito das ações da gestão, como baixa prioridade, foco em metas de curto prazo e custo.

2.5 CONSIDERAÇÕES FINAIS

Este capítulo apresentou resultados de pesquisas realizadas sobre causas e efeitos da DT. Também discorreu sobre a prevenção e o pagamento da DT, com atenção para as práticas empregadas nessas atividades e as razões que fundamentam a decisão de não aplicar essas práticas.

O próximo capítulo apresenta o projeto *InsighTD* no qual essa dissertação está contextualizada.

CAPÍTULO 3 – O PROJETO *INSIGHTD*

*Este capítulo apresenta o planejamento e o estado atual da execução do projeto *InsighTD*, no qual esta dissertação de mestrado está contextualizada.*

3.1 INTRODUÇÃO

Este capítulo inicialmente apresenta o projeto *InsighTD*, no qual está contextualizada esta dissertação. Em seguida, discorre sobre questões específicas relacionadas à metodologia utilizada para a realização deste trabalho. Dessa forma, nas seções restantes deste capítulo são apresentados o projeto *InsighTD*, considerando seu planejamento e população alvo. Em seguida, é apresentada a metodologia de pesquisa desta dissertação considerando as questões de pesquisa, o questionário utilizado e, por fim, os procedimentos de análise de dados.

3.2 O PROJETO *INSIGHTD*

InsighTD é uma família de *surveys* globalmente distribuída para investigar causas, efeitos, prevenção e formas de pagamento da DT em projetos de software. Ele foi planejado para ser executado de forma incremental e em larga escala, sendo fundamentado em replicações contínuas e independentes do questionário em diferentes países. No momento, pesquisadores de 12 países fazem parte do projeto: Arábia Saudita, Argentina, Brasil, Chile, Colômbia, Costa Rica, Estados Unidos, Finlândia, Índia, Itália, Noruega e Sérvia.

O elemento base para a execução do projeto *InsighTD* é o *survey*, replicado em diferentes países. O objetivo da aplicação do *survey* é obter resultados abrangentes sobre o estado da prática na área de DT. O intuito em se realizar sua replicação em diferentes países está relacionada a: (i) organizar melhor o trabalho e fazer com que a disseminação do *survey* seja mais ampla, e (ii) investigar se possíveis diferenças em práticas de desenvolvimento de cada país poderiam influenciar na forma como os participantes lidam com o conceito de DT. Além disso, a replicação em diferentes países contribui para ampliar o universo das organizações, como o tamanho e as metodologias utilizadas, e como ocorrem as percepções sobre o conceito de DT considerando as diferentes culturas de desenvolvimento.

Para auxiliar na difusão dos resultados e cooperação dos pesquisadores envolvidos, é utilizada uma estrutura que se baseia em uma única versão do questionário e uma coleção de procedimentos para orientar a análise de dados. Para compartilhar os resultados entre os participantes da pesquisa, um plano de comunicação é utilizado. Mais detalhes sobre o projeto

e informações sobre as publicações e replicações podem ser encontradas no site www.td-survey.com.

O projeto é estruturado em quatro etapas: concepção, validação, iniciação e replicação internacional. A etapa da concepção é responsável pelas atividades relacionadas ao planejamento do *survey*, destacando-se a definição das questões de pesquisa, planejamento da família de *surveys*, elaboração dos instrumentos de pesquisa, definição da população-alvo e discussão inicial sobre a análise dos dados. É nessa etapa que é realizada a definição do Time Central (TC) do *InsighTD*, que controla as atividades do projeto e tem como responsabilidade a execução da primeira rodada do *survey* no Brasil e coordenação dos Times de Replicação (TR).

A validação é subdividida em três etapas: interna, externa e estudo piloto. O objetivo da validação interna e externa é assegurar que as questões do *survey* sejam de fácil interpretação e completa a ponto de responder a questão da pesquisa, elevando o nível da validade interna do planejamento do estudo. No decorrer dessa atividade, os pesquisadores com a responsabilidade de realizar revisões, fazem a análise da estrutura do planejamento do estudo, verificando se as perguntas do questionário podem responder as questões de pesquisa. Já o estudo piloto tem a função de validar se o questionário pode ser entendido por um número pequeno de profissionais da área, avaliando também o esforço empregado para responder o questionário.

A etapa de iniciação se refere à primeira aplicação do *survey* realizada no Brasil e sua replicação nos Estados Unidos. Como a intenção é investigar o estado da prática, os participantes do *survey* pertencem à indústria de software. É considerada a participação de profissionais pertencentes a diferentes áreas (por exemplo, teste, requisitos, código, projeto) do projeto, já que existe a possibilidade de ocorrer DT em fases diferentes do projeto. As mídias sociais foram a principal ferramenta para busca dos profissionais, sendo o LinkedIn a principal ferramenta utilizada. Sua busca permite o uso de palavras-chaves, possibilitando encontrar participantes com conhecimentos específicos. O critério de busca utilizado no *InsighTD* é o país e o tipo de dívida. Por exemplo, para localizar profissionais que responderam sobre dívida de arquitetura, pode-se pesquisar pelo termo arquiteto de software. Outras formas de convidar os participantes para a pesquisa foram listas de discussão e parceiros da indústria de software.

Após os resultados da terceira etapa, um relatório de referência é gerado para contribuir na definição de como se pode executar a análise e síntese de dados. Além disso, cada parceiro dos diferentes países envolvidos recebe um pacote experimental utilizado para replicar a

pesquisa, podendo utilizar todo o conjunto de instrumentos na última etapa da pesquisa (replicação internacional). Cada replicação do *survey* é realizada de forma independente.

A disseminação contínua do *survey* permite obter uma vasta coleção de dados empíricos a respeito das causas, efeitos, prevenção e pagamento da DT, sendo usada pelos pesquisadores envolvidos no projeto para realizar análises de dados de forma isolada ou conjunta.

3.2.1 Estado atual

O projeto InsightTD é composto atualmente por 36 pesquisadores. O processo de coleta e análise de dados foi finalizado pelos times de replicação do Brasil, Chile, Colômbia, Costa Rica, Estados Unidos e Sérvia.

A forma de publicação dos trabalhos está organizada considerando áreas de concentração. As áreas de concentração são definidas da seguinte forma:

- Conceito DT: pretende investigar o nível de disseminação do conceito de DT bem como o que o caracteriza na prática;
- Causas de DT: busca mapear as principais causas e as mais prováveis de levar à presença de DT;
- Efeitos DT: busca mapear os principais efeitos e aqueles mais impactantes;
- Prevenção de DT: pretende investigar até que ponto a DT pode ser prevenida;
- Monitoramento de DT: pretende investigar se a DT vem sendo monitorado na indústria de software, as práticas de monitoramento aplicadas e os impedimentos que inibem o monitoramento de itens de DT;
- Pagamento TD: pretende investigar se a DT tem sido paga na indústria de software, as práticas de pagamento utilizadas e os impedimentos que inibem o pagamento dos itens de dívida.

Até o momento da escrita desta dissertação, 653 profissionais da indústria de software de seis países (Brasil=107, Chile=89, Colômbia=134, Costa Rica=145, Sérvia=79 e Estados Unidos=99) participaram da pesquisa. Em sua maioria, os participantes trabalham em empresas de médio porte, com equipes de cinco a nove pessoas, que adotam o processo de desenvolvimento híbrido. O projeto de software considerado possui de 10 a 100KLOC e dois a cinco anos. Em grande parte, são desenvolvedores com perfil especialista na função que desempenham no projeto.

3.3 METODOLOGIA

Este trabalho investiga as seguintes questões de pesquisa (QP):

QP1: As causas da DT estão mais relacionadas aos problemas de codificação ou a outros problemas de desenvolvimento de software? Esta questão investiga se os problemas de codificação são mais prováveis de ser uma fonte de DT do que outras questões do desenvolvimento de software, como processo e teste.

QP2: Os efeitos da DT são mais percebidos na codificação ou em outras questões do processo de desenvolvimento de software? Por meio desta questão, pretende-se verificar se os efeitos da DT são mais percebidos em questões de codificação ou em outras questões de desenvolvimento.

QP3: A prevenção DT está mais relacionada aos problemas de codificação ou a outras questões no processo de desenvolvimento de software? O objetivo desta questão é investigar se as práticas utilizadas pelas equipes de desenvolvimento para evitar itens da DT estão mais relacionadas a questões de codificação ou a outras questões do desenvolvimento.

QP4: As razões para não prevenir a DT estão mais relacionadas aos problemas de codificação ou a outros problemas de desenvolvimento? O objetivo desta questão é investigar se as razões que impedem as equipes de desenvolvimento de não aplicar práticas preventivas estão mais relacionadas aos problemas de codificação ou a outros problemas de desenvolvimento de software.

QP5: O pagamento DT está mais associado a problemas de codificação ou a outras questões no processo de desenvolvimento de software? Por meio dessa pergunta, investiga-se se as práticas usadas para eliminar itens de dívida estão mais associadas às questões de codificação ou a outras questões de desenvolvimento de software.

QP6: As razões para não pagar a DT estão mais relacionadas às questões de codificação ou a outras questões de desenvolvimento? Aqui, investiga-se se as razões para não aplicar as práticas de pagamento DT estão mais associadas aos problemas de codificação ou a outros problemas de desenvolvimento de software.

3.3.1 Coleta de Dados

Este estudo utiliza um subconjunto de dados disponíveis de 18 perguntas do questionário InshTD. A Tabela 1 apresenta essas questões, relatando seu tipo e a QP a que estão relacionadas.

Tabela 1 - Subconjunto do questionário utilizado

QP	Nº	Questão (Q)	Tipo
	Q1	Qual o tamanho da empresa onde trabalha (software e outras áreas)?	Fechada
	Q2	Atualmente você trabalha em que país?	Fechada
	Q3	Mais à frente, iremos fazer algumas perguntas sobre um projeto específico e, neste momento, gostaríamos que você escolhesse o projeto sobre o qual iria falar. Você pode escolher qualquer projeto de desenvolvimento de software (atual ou dentro dos últimos 3 anos) em que você atuou durante um período significativo. Depois de escolher o projeto sobre o qual você irá conciliar suas respostas, conte-nos, qual é o tamanho do sistema que está sendo desenvolvido nesse projeto? (LOC = linhas de código)	Fechada
	Q4	Qual é o número total de pessoas envolvidas no projeto (incluindo equipe técnica e de negócios)?	Fechada
	Q5	Qual é a idade do sistema (contando a partir do início das atividades do projeto até a data atual ou até quando você atuou no projeto)?	Fechada
	Q6	Qual função você desempenha(ou) neste projeto?	Fechada
	Q7	Como você avalia sua experiência nessa função?	Fechada
	Q8	Qual modelo de processo de desenvolvimento é utilizado neste projeto?	Fechada
	Q9	Quão familiarizado você está com o conceito de DT?	Fechada
	Q10	Em suas palavras, como você definiria DT?	Aberta
QP1	Q13	Por favor, forneça um exemplo de DT que teve um impacto significativo em seu projeto (este exemplo será usado para responder outras perguntas deste questionário):	Aberta
	Q16	Qual foi a causa que levou à ocorrência da DT que você acabou de descrever?	Aberta
	Q17	Qual outra causa ou fator contribuiu para a ocorrência da causa que você descreveu acima?	Aberta
QP2	Q18	Quais outros motivos, razões ou causas contribuíram direta ou indiretamente para a ocorrência do exemplo de DT?	Aberta
	Q19	Considerando todas as situações de DT que você vivenciou em diferentes projetos, quais causas você listaria como sendo aquelas que mais contribuem para a ocorrência da dívida (ordenadas pela probabilidade de causar a DT, com as mais prováveis listadas primeiro)? Por favor, liste até 5 causas.	Aberta
	Q20	Considerando o item de DT que você descreveu na pergunta 13, quais foram os efeitos sentidos no projeto?	Aberta
QP3	Q21	Considerando todas as situações de DT que você vivenciou em diferentes projetos, quais são os 5 efeitos que você classificaria como sendo aqueles que possuem um impacto maior no projeto (ordenados pelo seu nível de impacto, listando primeiro o de maior impacto).	Aberta
	Q22	Você acha que seria possível evitar o tipo de dívida que você descreveu na questão 13?	Fechada
QP4	Q23	Se sim, como? Se não, por quê?	Aberta
	Q26	O item da dívida do projeto foi pago (eliminado)?	Fechada
	Q27	Se sim, como? Se não, por quê?	Aberta

As perguntas Q1 a Q8 capturam a caracterização dos respondentes da pesquisa. Em Q13, eles fornecem um exemplo de um item de DT que ocorreu em seus projetos. Os participantes discutem as causas da DT em Q16 a Q18 e os efeitos em Q20. Usamos as respostas fornecidas a essas perguntas para responder a QP1 (Q16-Q18) e QP2 (Q20). Com relação à prevenção da DT, os participantes fornecem suas respostas em Q22 e Q23, e abordam o pagamento de DT em Q26 e Q27. As respostas obtidas nestas perguntas são usadas para responder QP3-4 (Q22 e Q23) e QP5-6 (Q26 e Q27).

3.3.2 Procedimentos de Análise de Dados

O procedimento de análise é dividido em três etapas: demografia, preparação dos dados para análise e classificação e análise dos dados. Sendo a Etapa 3 especificamente realizada no contexto deste trabalho.

3.3.2.1 Etapa 1 - Dados demográficos

O *survey* é composto por questões abertas e fechadas. Nas questões fechadas foi adotada a estatística descritiva para facilitar a compreensão dos dados. Foi realizado o cálculo da quantidade de participantes que escolheram uma das opções definidas em cada uma das questões fechadas. Assim, foi possível caracterizar os participantes (Q1-Q18), identificar a representatividade do item da DT descrito pelos participantes na Q13 e verificar se o item da DT foi ou não prevenido (Q22).

3.3.2.2 Etapa 2 - Preparação de dados para análise

Aplicou-se um processo de codificação para as questões abertas. Nas respostas fornecidas para Q16 a Q18 e Q20, seguiu-se o processo de codificação previamente descrito em (Nicolli *et al.*, 2020), resultando em um conjunto de causas e efeitos e seus respectivos números de ocorrências. Nas respostas fornecidas para Q23, foi realizado o processo de codificação descrito em (Freire *et al.*, 2020). A partir desse processo, identificou-se práticas de prevenção de DT quando Q22 recebe uma resposta positiva, caso contrário, reconhece os motivos para a não prevenção da DT. Por fim, seguindo o processo de codificação descrito em (Freire *et al.*, 2020), foram codificadas as respostas dadas a Q27. Da mesma forma, quando Q26 recebe uma resposta positiva, as práticas de pagamento da DT são identificadas, caso contrário, os motivos para o não pagamento são observados. Tanto para prevenção quanto para pagamento, uma lista de práticas e motivos, e seu número de ocorrências foi extraído.

O processo de codificação foi realizado por pelo menos dois pesquisadores de cada replicação. A primeira lista codificada de causas, efeitos, práticas de prevenção, motivos para não prevenir, práticas de pagamento e motivos para não pagamento (Brasil: codificado por Nicolli Rios, Sávio Freire e Rodrigo Spínola) foi enviada para as demais equipes de replicação (Chile e Colômbia: Boris Pérez, Cristian Castellanos e Dario Correal; Costa Rica: Gustavo Lopez; Sérvia: Robert Ramac, Vladimir Mandic e Nebojša Taušan; Estados Unidos: Nicolli Rios, Clemente Izurieta, Davide Falessi e Carolyn Seaman) para padronizar a nomenclatura utilizada. A consistência foi verificada por Nicolli Rios, Sávio Freire ou Rodrigo Spínola.

3.3.2.3 Etapa 3 - Classificação e análise de dados

Inicialmente, é analisado os códigos de cada elemento de gerenciamento da DT e definido se eles estão relacionados a problemas de codificação ou a outros problemas de desenvolvimento de software. Por exemplo, as práticas de pagamento *correção de bugs*, *refatoração de código* e *uso de reutilização de código* foram classificadas como práticas relacionadas a problemas de codificação. Por outro lado, as práticas de pagamento *priorização de itens da DT* e *atualização da documentação do sistema* foram associadas a outras questões de desenvolvimento de software. Este processo foi realizado por Clara Berenguer e Adriano Borges de forma independente. O consenso foi realizado por Nicolli Rios (causas e efeitos) e Sávio Freire (prevenção e pagamento). Além disso, a classificação final foi revisada por Rodrigo Spínola.

Em seguida, agrupa-se os elementos de gerenciamento de DT relacionados às outras questões de desenvolvimento de software em categorias seguindo o processo de agrupamento definido por (Rios *et al.*, 2020). As categorias revelam a relação entre as questões do processo de desenvolvimento de software (por exemplo, questões de engenharia de requisitos, questões de planejamento e gerenciamento, questões de fatores humanos) e cada elemento de gestão da DT. Os nomes das categorias surgem do processo contínuo de agrupamento dos elementos de gestão da DT em torno da preocupação central a que estão relacionados. Por exemplo, as causas *prazo* e *planejamento inadequado* fazem parte da categoria questões de planejamento e gestão, enquanto os efeitos *desmotivação da equipe* e *insatisfação das partes envolvidas* compõem a categoria fatores humanos. Este processo foi conduzido por Clara Berenguer e Adriano Borges de forma independente. O consenso foi realizado por Nicolli Rios (causas e efeitos) e Sávio Freire (práticas e motivos), e o resultado foi revisado por Rodrigo Spínola.

3.4 Considerações Finais

O projeto InsignTD tem o propósito de reunir um conjunto aberto e generalizável de dados empíricos a respeito de questões práticas (causas, efeitos, prevenção e pagamento) da DT que possam contribuir para novas pesquisas de uma forma orientada a problemas enfrentados na indústria. Neste capítulo, discutiu-se seu planejamento, estado atual de execução, e como esta dissertação está contextualizada no projeto.

O próximo capítulo apresenta os resultados obtidos a partir da análise dos dados coletados nas replicações do projeto InsignTD, considerando as questões de pesquisa definidas.

CAPÍTULO 4 – RESULTADOS

*Neste capítulo, são apresentados os resultados da análise de dados coletados no contexto do Projeto *InsighTD* com o objetivo de identificar se as causas, efeitos, práticas de prevenção, razões para não prevenção, práticas de pagamento e razões para não pagar estão mais associadas a código ou a outras atividades do desenvolvimento.*

4.1 INTRODUÇÃO

Equipes de desenvolvimento podem considerar a DT como um bom investimento, no entanto, é necessário identificar a sua presença e ter ciência do aumento dos riscos impostos ao projeto (KRUCHTEN; NORD; OZKAYA, 2012). Se a DT for bem gerenciada, os objetivos do projeto podem ser atingidos mais cedo e com menos custo. Caso contrário, a sua presença pode ocasionar problemas técnicos e financeiros, levando ao aumento dos custos de manutenção e evolução do software (MARTINI; BOSCH; CHAUDRON, 2014; NORD *et al.*, 2012).

Conhecer a DT e sua incidência é fundamental para que o bom gerenciamento ocorra. O objetivo deste capítulo é apresentar os resultados da investigação realizada sobre relação entre os elementos de gestão da DT e as atividades de desenvolvimento de um projeto. A pesquisa teve como base os seguintes elementos:

- Causas: são os motivos que levam a ocorrência da DT,
- Efeitos: são os impactos causados com a ocorrência de um item de dívida,
- Práticas de prevenção: são as ações tomadas para evitar um item de dívida,
- Razões para não prevenir: são os motivos para que as equipes não executem as práticas de prevenção da DT,
- Práticas de pagamento: são as ações tomadas para eliminar um item de dívida,
- Razões para não pagar: são os motivos que levam as equipes a não eliminar os itens da DT.

A próxima seção apresenta os resultados alcançados por meio da análise dos dados do Projeto *InsighTD*.

4.2 RESULTADOS

Foi solicitado na Q10 uma definição sobre a DT e na Q13 um exemplo de um item da DT relevante. As respostas fornecidas para essas questões foram utilizadas para validar os participantes que seriam incluídos para análise. Ao final, foram consideradas respostas de 653 profissionais de seis países (Brasil = 107, Chile = 89, Colômbia = 134, Costa Rica = 145, Estados Unidos = 99 e Sérvia = 79). A pesquisa levou em consideração para caracterizar os participantes, questões específicas do projeto de software, como o tamanho, idade, quantidade de pessoas na equipe e modelo de processo utilizado. Elementos como tamanho da organização, nível de experiência do participante e sua função também foram considerados.

4.2.1 Dados demográficos

Com relação ao nível de experiência, a maioria dos participantes é competente (~34%), seguido por proficiente (~32%), especialista (~21%), iniciante (~12%) e novato (~1%). A maioria deles trabalha como desenvolvedores, seguidos por líder ou gerente de projeto, arquiteto de software, testador, analista de processo e outras funções.

Os participantes geralmente trabalham em projetos com 2 a 5 anos (~34%), seguidos por 1 a 2 anos (~23%), menos de 1 ano (~17%), 5 a 10 anos (~15%) e mais de 10 anos (~11%). O software em desenvolvimento normalmente possui entre 10 a 100 KLOC (~30%) seguido por 100 KLOC a 10 MLOC (~30%), menos de 10 KLOC (~14%) e acima de 10 MLOC (~7%). Em relação ao tamanho da equipe dos participantes, geralmente possui de 5 a 9 pessoas (~34%), seguido por equipes com 10 a 20 pessoas (~22%), menos de 5 pessoas (~20%), mais de 30 pessoas (~16%) e 21 a 30 pessoas (~8%). Por fim, notou-se que o processo híbrido foi o mais utilizado (~45%), seguido por processos ágeis (~42%) e tradicionais (~13%).

A amostra representa um conjunto amplo e diversificado de profissionais da indústria de software, apesar de não ser possível garantir que os participantes representam os profissionais em sua totalidade.

Os resultados obtidos para cada QP serão apresentados nas subseções seguintes. Para cada uma delas, seguiu-se a mesma estrutura para responder à questão. Por exemplo, para o elemento causa da DT, inicialmente (i) discute-se sobre as causas relacionadas ao código. Em seguida (ii), apresenta-se as causas relacionadas às outras questões do desenvolvimento de software. Por fim, (iii) analisa-se quais são as categorias dessas questões (por exemplo, planejamento e gestão, fatores humanos, problemas de conhecimento).

4.2.2 QP1: As causas da DT estão mais relacionadas aos problemas de codificação ou a outros problemas de desenvolvimento de software?

De um total de 1695 citações, foram identificadas 96 causas que levam à ocorrência da DT. Do total, ~92% tinha ligação com outras questões do desenvolvimento, enquanto que apenas ~8% tinha relação com o código. Com isso, é possível constatar que existe uma tendência grande de ocorrência da DT devido a questões relacionadas a etapas do desenvolvimento de softwares não ligadas à codificação.

São 22 causas ligadas à codificação. Na Tabela 2 estão apresentadas as 10 causas mais citadas. Os destaques foram para *adoção de boas práticas*, *código desleixado*, *falta de refatoração* e *adoção de soluções de contorno como definitivas*. Todas elas direcionam para a ideia do comprometimento da qualidade interna do produto.

Tabela 2 - As 10 causas mais citadas relacionadas à codificação e outros problemas de desenvolvimento

Código		Outros problemas de desenvolvimento	
Causa	#	Causa	#
1º Não adoção de boas práticas	50	Prazo final	167
2º Código desleixado	18	Planejamento impróprio	79
3º Falta de refatoração	14	Gerenciamento de projeto não eficaz	64
4º Adoção de soluções de contorno como definitivas	11	Produzindo mais em detrimento da qualidade	62
5º Dependência de componente externo	10	Falta de experiência	52
6º Falta de práticas de reutilização	5	Pressão	50
7º Falta de testes automatizados	5	Estimativa de tempo imprecisa	46
8º Componente descontinuado	4	Falta de profissionais qualificados	42
9º Mudança no design	4	Falta de um processo bem definido	40
10º Modelo de dados inadequado	3	Teste impróprio	37

Em contrapartida, 196 causas foram identificadas e classificadas com relação à ocorrência da DT em outras etapas do desenvolvimento de software. Na terceira coluna da Tabela 2, é possível observar que as quatro causas mais citadas são todas direcionadas ao gerenciamento e planejamento do projeto. Outras questões relacionadas à falta de experiência, pressão e processos também foram comumente mencionadas.

Analisando-se as causas identificadas, foi possível observar que muitas estavam relacionadas entre si. Com isso, foi possível identificar oito categorias de causas:

- **Planejamento e gerenciamento:** agrupa causas relacionadas às atividades de planejamento e gerenciamento do projeto. Alguns exemplos são *prazo*, *planejamento inadequado* e *gerenciamento de projeto não eficaz*;

- **Fatores humanos:** reúne as causas ligadas à participação das pessoas nas atividades do projeto. Alguns exemplos são *falta de experiência*, *falta de comunicação da equipe* e *falta de comprometimento*;
- **Problemas de conhecimento:** reúne pontos que surgem das preocupações em torno do conhecimento dos membros da equipe. Dois exemplos são *falta de conhecimento técnico* e *falta de conhecimento do domínio*;
- **Engenharia de requisitos:** contém as causas relacionadas às atividades de requisitos. Dois exemplos são: *mudança de requisitos* e *questões de elicitação de requisitos*;
- **Verificação, validação e teste:** reúne as causas relacionadas à execução das atividades de garantia da qualidade. Dois exemplos são *teste impróprio / mal planejado / mal executado* e *falta de revisão de código*;
- **Questões de arquitetura:** reúne as causas associadas às decisões tomadas em relação à arquitetura de software. São exemplos: *escolha inadequada de tecnologia / ferramenta / plataforma* e *má definição da estrutura*;
- **Problemas de processo:** referem-se às causas relacionadas à definição ou execução dos processos utilizados no desenvolvimento do software. Dois exemplos são *falta de um processo bem definido* e *falta de rastreabilidade de bugs*;
- **Questões de design:** engloba as causas relacionadas ao *design* do software. Existem três causas nesta categoria: *design ruim* e *mudança no design*.

A Tabela 3 apresenta a quantidade de causas citadas com relação às categorias. A categoria *planejamento e gestão* se destacou com ~44% das citações, representando mais de três vezes as citações do segundo colocado. Também foi possível perceber que *fatores humanos* se destacaram com ~13% das citações. Preocupações associadas à engenharia de requisitos, problemas relacionados ao conhecimento e verificação, validação e testes também foram comumente mencionados em cerca de 7% das citações, cada.

Tabela 3 - Categorias de causas não relacionadas à codificação

Categorias de causas	#causa	#causas citadas	~% causas citadas
Planejamento e Gestão	68	748	44%
Fatores humanos	28	210	13%
Questões de Conhecimento	14	130	8%
Problemas de Conhecimento	16	121	7%
VV&T	14	111	7%
Problemas de arquitetura	11	67	4%
Problemas de Processo	11	53	3%
Problemas de design	3	45	3%

4.2.3 QP2: Os efeitos da DT são mais percebidos na codificação ou em outras questões do processo de desenvolvimento de software?

O resultado da pesquisa permitiu identificar um total de 73 efeitos e um total de 980 citações. Entre eles, ~64% estão relacionados a outras questões de desenvolvimento e ~36% estão relacionados à codificação.

Os participantes reportaram 28 efeitos relacionados à codificação. Os 10 mais citados podem ser observados na Tabela 4 (segunda coluna). Destacam-se a capacidade que a equipe pode ter em relação à evolução do código e a possibilidade de empregar práticas de refatoração com o intuito de melhorar a qualidade interna do software. Além disso, outros efeitos comuns são: *código ruim*, *baixa qualidade interna* e *baixo desempenho*.

Tabela 4 - Os 10 efeitos mais citados relacionados à codificação e outras questões de desenvolvimento

	Codificação		Outros problemas de desenvolvimento	
	Efeitos	#	Efeitos	#
1º	Baixa manutenção	91	Atraso na entrega	139
2º	Retrabalho	86	Baixa qualidade externa	52
3º	Necessidade de refatoração	32	Aumento do esforço no trabalho	41
4º	Código ruim	28	Aumento do custo	29
5º	Baixa qualidade interna	24	Baixo financiamento	25
6º	Baixa performance	24	Desmotivação do time	24
7º	Pare de desenvolver. atividades para pagamento de dívidas	14	Insatisfação das partes envolvidas	23
8º	Dificuldade em implementar o sistema	10	Estresse com as partes interessadas	23
9º	Aumento na quantidade de atividades de manutenção	8	Sobrecarga da equipe	15
10º	Reutilização de código baixo	8	Projeto não concluído	13

Sobre as outras etapas do desenvolvimento, foram reportados 80 efeitos. Os cinco mais comumente (terceira coluna da Tabela 4) citados refletem preocupações sobre o gerenciamento e planejamento do projeto (*atraso na entrega*, *aumento do esforço*, *aumento do custo*, *perda financeira*) e qualidade externa do produto (*baixa qualidade externa*). Causas relacionadas a fatores humanos também foram reportadas, com destaque para *desmotivação da equipe*, *insatisfação das partes envolvidas* e *estresse dos stakeholders*.

A Tabela 5 apresenta as categorias definidas para os efeitos. Na categoria *planejamento e gestão*, está quase metade dos efeitos citados, reforçando que questões gerenciais do desenvolvimento de software são frequentemente afetadas pela ocorrência de itens de dívida. Em seguida, tem-se a categoria *fatores humanos*, com ~18% dos efeitos citados, mostrando que DT pode trazer consequências negativas a questões relacionadas à interação entre as pessoas

durante o processo de desenvolvimento. *Questões relacionadas à qualidade* também são uma preocupação comum. Outras duas categorias foram definidas:

- **Questões organizacionais:** agrupam efeitos que impactam a organização. Exemplos de efeitos nesta categoria são *imagem da empresa prejudicada* e *perda de competitividade no mercado*;
- **Problemas de documentação:** agrupa efeitos relacionados à documentação (ex.: *documentação inadequada / inexistente / desatualizada*).

Tabela 5 - Categorias de efeitos não relacionados à codificação

Categorias de efeitos	#efeitos	#efeitos citados	~% efeitos citados
Planejamento e Gestão	27	300	49%
Fatores humanos	13	110	18%
Problemas de qualidade	10	84	14%
VV&T	9	23	4%
Projeto	4	21	4%
Problemas de conhecimento	9	19	3%
Problemas de arquitetura	4	18	3%
Questões organizacionais	3	10	2%
Documentação	1	6	1%

4.2.4 QP3: A prevenção DT está mais relacionada aos problemas de codificação ou a outras questões no processo de desenvolvimento de software?

Para essa questão, o resultado da pesquisa retornou um total de 819 citações a respeito das práticas de prevenção da DT, sendo identificadas 89 práticas. Desse total, ~84% são itens relacionados a outras questões do desenvolvimento de software, enquanto apenas ~16% estão associados a código. Esse resultado indica uma tendência de que outras questões do desenvolvimento desempenham um papel importante na prevenção do DT.

Relacionado à codificação, foi identificado um total de 13 práticas de prevenção de DT. A Tabela 6, segunda coluna, apresenta as 10 mais citadas. As práticas que se destacaram foram: *adoção de boas práticas*, *uso de boas práticas de design*, *refatoração*, *revisão de código*, *aumento do tempo de análise e design*, *uso da versão mais adequada da tecnologia* e *reutilização adequada do código*.

A *adoção de boas práticas* e o *uso de boas práticas de design* refletem as preocupações que os profissionais possuem ao realizar suas atividades de codificação e *design*. As práticas de *refatoração* e *revisão de código* estão relacionadas à melhoria contínua do código em desenvolvimento. Por fim, *aumentar o tempo de análise e design*, *usar a versão mais adequada da tecnologia* e o *reaproveitamento adequado do código* estão relacionados às preocupações

que as equipes devem ter em torno de uma análise adequada das funcionalidades, implementação da estrutura do software e reutilização do software, respectivamente.

Tabela 6 - As 10 práticas de prevenção de DT mais comumente citadas relacionadas à codificação ou outras questões de desenvolvimento

Código			Outras questões do desenvolvimento	
	Práticas de Prevenção	#	Práticas de Prevenção	#
1º	Adoção de boas práticas	49	Requisitos bem definidos	57
2º	Usando boas práticas de design	26	Melhor gerenciamento de projetos	43
3º	Reestruturação	12	Oferecendo treinamento	36
4º	Revisão de código	10	Seguindo o planejamento do projeto	34
5º	Aumento do tempo para análise e design	7	Melhorar o processo de desenvolvimento de software	33
6º	Use a versão mais adequada da tecnologia	7	Melhore a documentação	26
7º	Reutilização adequada de código	6	Prazos bem planejados	26
8º	Controle de versão	5	Planejamento do projeto mais adequado	24
9º	Considerando as restrições técnicas	4	Criação de testes	24
10º	Impro. a sustentabilidade do projeto	4	Alocação de profissionais qualificados	23

Em contrapartida, foram identificadas 76 práticas de prevenção associadas a outras questões do desenvolvimento de software. Na tabela 6, terceira coluna, é possível observar as 10 mais citadas. Elas refletem pontos diferentes referente ao processo de desenvolvimento de software, como gerenciamento (*segundo o planejamento do projeto e melhor gerenciamento do projeto*), o próprio processo (*melhorando o processo de desenvolvimento de software*), a documentação (*requisitos bem definidos*) e a qualificação da equipe (*fornecendo treinamento*).

Na Tabela 7, tem-se a distribuição das práticas de prevenção da DT em suas respectivas categorias. Destaca-se que ~34 % das práticas estão associadas a atividades de gerenciamento de projetos. Ocupando o segundo lugar, tem-se o processo que é seguido pela equipe com ~12%. Outras categorias também foram identificadas: questões relacionadas a requisitos, VV&T, gerenciamento de DT e fator humano. As demais categorias foram menos comumente citadas.

Tabela 7 - Categorias de práticas de prevenção não relacionadas a código

Categorias de práticas de prevenção	#prática	#práticas citadas	~%práticas citadas
Planejamento e Gestão	21	232	34%
Problemas de Processo	8	80	12%
Engenharia de Requisitos	5	69	11%
VV&T	11	67	10%
Gestão TD	7	64	10%
Fatores humanos	11	61	9%
Problemas de Conhecimento	4	51	8%
Problemas de documentação	2	28	4%
Problemas de arquitetura	3	27	4%
Questões Organizacionais	2	4	1%
Problemas de infraestrutura	2	3	1%

4.2.5 QP4: As razões para não prevenir a DT estão mais relacionadas aos problemas de codificação ou a outros problemas de desenvolvimento?

Identificou-se 25 razões que levam as equipes a não prevenirem a DT, resultando em 63 citações. Destas, ~87% estão relacionadas a outras questões de desenvolvimento, enquanto apenas ~13% estão relacionadas à codificação. Novamente, outras questões do desenvolvimento têm um papel importante na prevenção da DT.

A Tabela 8 apresenta as razões relacionadas à codificação e aquelas relacionadas a outras questões do desenvolvimento. Existem quatro razões relacionadas ao código: *falta de conhecimento técnico, mudança contínua dos padrões de codificação, falta de boas soluções técnicas e falta de preocupação com a manutenibilidade*.

Tabela 8 - As 10 razões mais citadas para não prevenir a DT

Código		Outras questões do desenvolvimento	
Razões	#	Razões	#
1° Falta de tecnologia. conhecimento	3	Prazo curto	14
2° Falta de boas soluções técnicas	2	Gestão ineficaz	7
3° Falta de preocupação com a manutenção	2	Falta de previsibilidade no desenvolvimento do software	5
4° Mudança contínua de padrões de codificação	1	Mudança de requisitos	5
5° -	-	Pressão por resultados	4
6° -	-	Problemas de documentação	2
7° -	-	Falta de maturidade do processo	2
8° -	-	Falta de profissionais qualificados	2
9° -	-	Sistema legado difícil de curar	2
10° -	-	Aceitando o TD	1

Na Tabela 9 são apresentadas as categorias relacionadas às razões para não pagamento da DT. A categoria *planejamento e gestão* mais uma vez destaca-se com cerca de 38% das citações. As demais categorias foram menos citadas. Embora exista uma concentração de citações em algumas categorias, ainda assim o resultado sugere que outras questões

relacionadas ao desenvolvimento de software também podem influenciar negativamente as equipes na prevenção de DT.

Tabela 9 - Categorias de motivos para a não prevenção do DT não relacionadas à codificação

Razões das Categorias	#razões	#razões citadas	~%razões citadas
Planejamento e Gestão	2	21	38%
Engenharia de Requisitos	2	6	11%
Codificação	1	5	9%
Fatores externos	2	5	9%
Fatores humanos	4	4	8%
Problemas de Processo	2	3	6%
Problemas de design	1	2	4%
Problemas de documentação	1	2	4%
Problemas de Conhecimento	1	2	4%
Gestão TD	2	2	4%
Problemas de arquitetura	1	1	2%
Problemas de infraestrutura	1	1	2%
Questões Organizacionais	1	1	2%

4.2.6 QP5: O pagamento da DT está mais associado a problemas de codificação ou a outras questões no processo de desenvolvimento de software?

Foram identificadas 32 práticas de pagamento da DT, resultando em um total de 315 citações. Desse total, ~56% estão relacionadas a outras questões de desenvolvimento. Em contrapartida, ~44% estão associadas a código. Comparando com as outras variáveis do estudo, as porcentagens encontradas indicam que as atividades de codificação ocupam uma posição de destaque nas ações de pagamento da DT.

Na Tabela 10 estão listadas as oito práticas de pagamento identificadas relacionadas à codificação. A *refatoração de código* e a *refatoração de design* são as práticas mais citadas. Ambas estão associadas à evolução da estrutura interna do sistema sem realizar alterações no comportamento externo. As práticas de *solução de problemas técnicos* e *correção de bugs* são focadas na correção de problemas relacionados a código.

Por fim, as práticas de *análise de código*, *revisão de código* e *reutilização de código* podem auxiliar as equipes de software na implementação de iniciativas de pagamento da DT. Embora essas práticas não atuem diretamente na eliminação de um item de dívida, elas contribuem para a definição de um cenário favorável para o pagamento da DT.

Tabela 10 - As 10 práticas de pagamento da DT mais comumente citadas

Código		Outras questões do desenvolvimento		
Práticas de pagamento	#	Práticas de pagamento	#	
1º	Refatoração de código	80	Investir esforço em atividades de reembolso TD	33
2º	Refatoração do design	25	Investir esforço em atividades de teste	22
3º	Adoção de boas práticas	10	Priorizando itens TD	15
4º	Resolvendo problemas técnicos	9	Negociando extensão de prazo	14
5º	Corrigindo erro	6	Atualizar a documentação do sistema	9
6º	Usando análise de código	3	Monitorar e controlar as atividades do projeto	9
7º	Revisão de código	3	Aumente o orçamento do projeto	9
8º	Usando a reutilização de código	2	Melhorar o processo de desenvolvimento	8
9º	-	-	Contratação de profissionais especializados	8
10º	-	-	Usando iterações de feedback curtas	7

Também foram identificadas 24 práticas de pagamento relacionadas a outras questões do desenvolvimento. Na terceira coluna da Tabela 10 está a lista das mais citadas. É possível observar que elas estão relacionadas a diferentes questões do desenvolvimento: documentação (*atualização da documentação do sistema*), decisões organizacionais (*contratação de profissionais especializados*), gerenciamento de projetos (*aumento do orçamento do projeto, monitoramento e controle das atividades do projeto, negociação da extensão do prazo, investimento de esforço no pagamento da DT e priorizando itens de TD*), processo (*melhorando o processo de desenvolvimento e usando iterações de feedback curtas*) e qualidade de software (*investindo esforço em atividades de teste*).

Na Tabela 11 são listadas as categorias de práticas de pagamento. O destaque foi para *gestão e planejamento* e *gestão da DT*, com ~32% e ~27% do total das citações. As categorias de *verificação, validação e teste*, e *problemas de processo* foram citadas por ~12% dos participantes, enquanto as demais categorias foram citadas com menor frequência.

Tabela 11 - Categorias de práticas de pagamento não relacionadas ao código

Categorias de práticas de pagamento	#prática	#práticas citadas	~% práticas citadas
Gestão TD	4	56	32%
Planejamento e Gestão	8	47	27%
VV&T	1	22	13%
Problemas de Processo	5	21	12%
Documentação	1	9	6%
Questões organizacionais	1	8	5%
Fatores humanos	1	6	4%
Engenharia de Requisitos	1	3	2%
Problemas de infraestrutura	1	3	2%
Problemas de design	1	2	1%

4.2.7 QP6: As razões para não pagar a DT estão mais relacionadas às questões de codificação ou a outras questões de desenvolvimento?

Foram identificadas 27 razões para o não pagamento dos itens da DT, totalizando 319 citações. Das razões identificadas 99,7% estão relacionadas a outras questões do desenvolvimento. A Tabela 12 apresenta as dez razões mais comumente relatadas para o não pagamento de itens da DT. Nota-se que grande parte das razões (*foco em metas de curto prazo, falta de tempo, custo, falta de recursos, esforço, projeto descontinuado, complexidade do item da DT e visão gerencial insuficiente sobre o pagamento da DT*) tem relação com o planejamento e gerenciamento do projeto. As demais razões estão associadas a fatores externos (*decisão do cliente*) e humanos (*sobrecarga da equipe*).

Tabela 12 - As 10 principais razões para não pagar a DT

Outros problemas do desenvolvimento			
Razões	#	Razões	#
1º Focando em objetivos de curto prazo	69	7º Complexidade do item de DT	12
2º Falta de interesse da organização	48	8º Esforço	11
3º Falta de tempo	41	9º Visão gerencial insuficiente sobre o pagamento da DT	10
4º Custo	34	10º Complexidade do projeto	10
5º Falta de recursos	19	1º O projeto foi descontinuado	6
6º Decisão do cliente	13	12º Sobrecarga da equipe	6

Na Tabela 13 agrupamos as razões por categorias. A categoria *planejamento e gestão* destaca-se com ~58% das citações, indicando que as razões desta categoria são decisivas para o não pagamento dos itens da dívida. As categorias questões organizacionais e gerenciamento de DT também foram comumente citadas por ~16% e ~11% dos participantes.

Tabela 13 - Categorias de razões para o não pagamento da TD não relacionadas à codificação

Categorias de motivos	#razões	#razões citadas	~% razões citadas
Planejamento e Gestão	7	185	58%
Questões organizacionais	2	50	16%
Gestão da DT	7	34	11%
Fatores externos	1	13	5%
Problemas de conhecimento	3	12	4%
Fatores humanos	3	11	4%
Problemas de arquitetura	2	11	4%
VV&T	1	2	1%

4.3 DISCUSSÃO

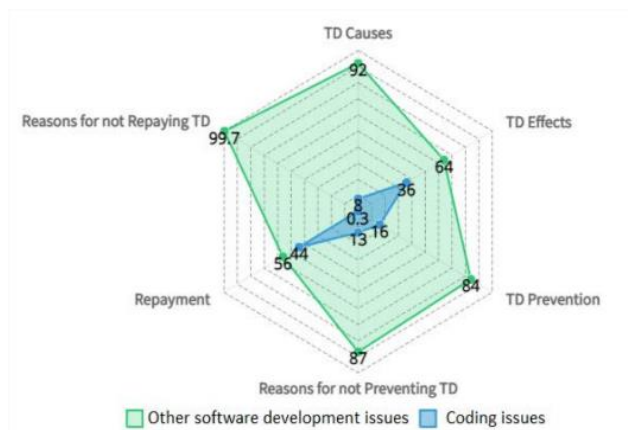
No geral, os resultados indicaram que as questões de codificação relacionadas às causas, efeitos, prevenção, não prevenção, pagamento e não pagamento da DT são apenas uma pequena

parte das preocupações que os profissionais enfrentam na presença da DT. Na verdade, a DT é muito comumente encontrada em outras questões do desenvolvimento de software.

O gráfico radar apresentado na Figura 2 mostra, em percentual, a distribuição das respostas dos participantes sobre cada uma das variáveis investigadas quanto à classificação codificação / outras questões do desenvolvimento.

Para todos os elementos do gerenciamento da DT investigados, a maioria das respostas está relacionada a outras questões de desenvolvimento de software. A diferença é significativa para os elementos: causas, prevenção, motivos para não prevenir e motivos para não retribuir. Já os valores relacionados ao pagamento da DT são muito próximos entre os dois grupos (56% vs 44%). Isso é uma indicação de que, embora os profissionais percebam que a DT seja onipresente em projetos de desenvolvimento de software, eles também veem que o pagamento de itens de dívida está muito comumente relacionado a atividades de codificação.

Figura 2 - Distribuição de respostas do participante

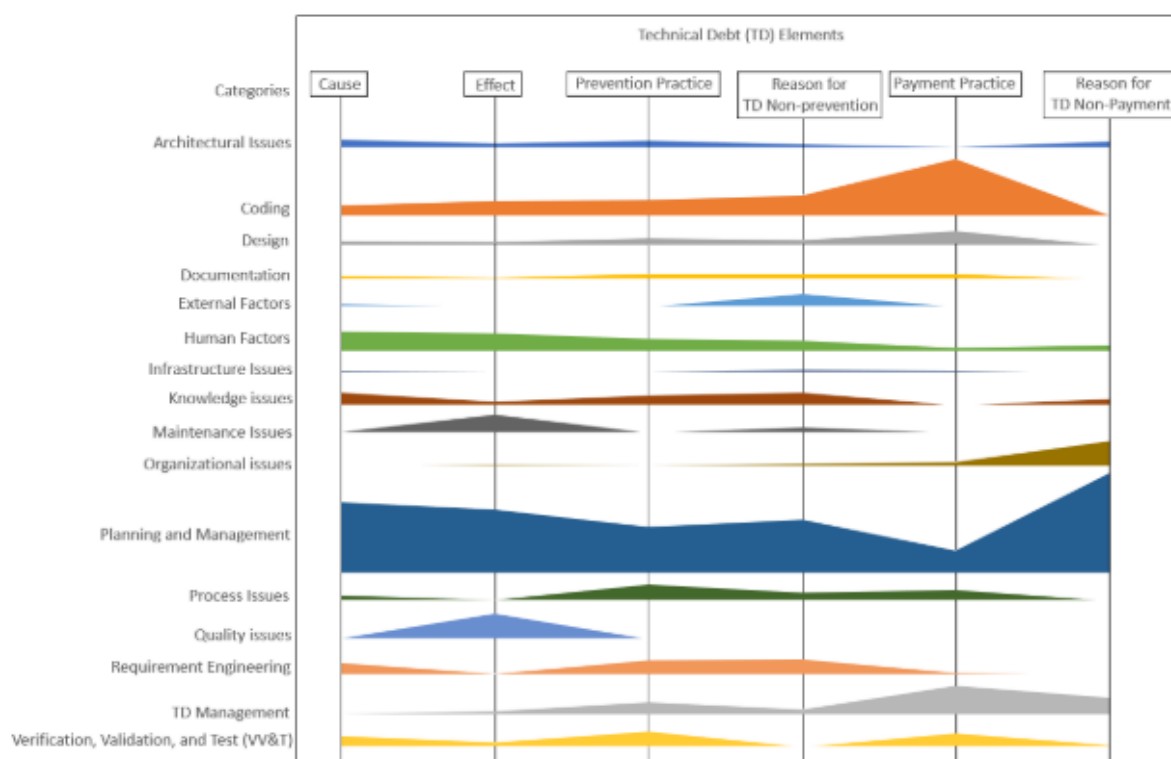


Para melhor representar a distribuição dos elementos da gestão da DT (causas, efeitos, práticas de prevenção, razões para não prevenção, práticas de reembolso e razões para não reembolso) em relação às diferentes preocupações relacionadas a eles durante o desenvolvimento de projetos de software, foi utilizado o diagrama de *hump* (Figura 3).

O cálculo dos valores da combinação de cada elemento do gerenciamento da DT com as questões do desenvolvimento de software (incluindo a codificação) foi realizado através da divisão do número de citações de um elemento de uma questão do desenvolvimento de software específica pelo número total de citações desse elemento. Por exemplo, considerando que as práticas de prevenção têm 819 citações e a categoria planejamento e gerenciamento tem 232 citações. Logo seu valor projetado no diagrama *hump* é de 28% ($232/819 * 100$). Esse cálculo

é um pouco diferente do realizado nas Tabelas 3, 5, 7, 9, 11 e 13 porque agora a codificação foi considerada como outra questão do desenvolvimento de software, de forma que se possa projetar todos os resultados (código e não código) no mesmo diagrama.

Figura 3 - O diagrama de corcunda para elementos de gerenciamento de TD e seus problemas de desenvolvimento de software



A leitura do diagrama pode ser realizada de forma horizontal ou vertical. Na horizontal, temos uma visão abrangente da repercussão dos elementos de gerenciamento da DT em cada questão do desenvolvimento de software. Percebe-se que a codificação tem uma função importante para todos os elementos analisados, com foco principal para o pagamento da DT. Existe um grande agrupamento de práticas relacionadas ao pagamento da DT, enquanto poucas razões para o não pagamento dos itens de dívida estão relacionados a questões da codificação. Deve-se ter atenção a outras questões que precisam ser tratadas em projetos de software, por exemplo, planejamento e gestão. Muito sobre o não pagamento da DT pode ser entendido examinando essas questões. Os fatores humanos também são importantes, fazendo com que a DT seja avaliada sobre outro ponto de vista: moral, satisfação, motivação, comunicação e comprometimento da equipe. Outros problemas comumente encontrados em vários elementos do gerenciamento da DT são problemas de arquitetura, problemas de design, documentação, problemas de conhecimento, problemas de processo, engenharia de requisitos e VV&T.

Já a análise do diagrama de forma vertical evidencia os impactos de todas as questões do desenvolvimento de acordo com o elemento do gerenciamento da DT. Por exemplo, questões de planejamento e gestão, organizacionais e de gestão de DT são decisivas para o não pagamento de itens de dívida. Outro ponto de destaque é que a existência de itens de dívida pode impactar principalmente no planejamento e na gestão, nas questões de qualidade e questões de manutenção, fatores humanos e codificação.

O diagrama de *hump* pode ajudar profissionais na análise e compreensão sobre como os itens da DT se relacionam com as diversas questões dos projetos, desde questões organizacionais até questões de nível de codificação. Observando a Figura 3, um profissional pode notar que os efeitos da DT são mais relacionados a questões de codificação, fatores humanos, manutenção, qualidade e planejamento e gerenciamento. Se o profissional estiver mais interessado em descobrir mais sobre os fatores humanos, então ele pode observar nos resultados detalhados ao longo do capítulo e notar que a desmotivação da equipe, a insatisfação das partes envolvidas e o estresse com os stakeholders são as principais preocupações a serem mitigadas nesse contexto.

Para pesquisadores, os resultados indicam a necessidade de empregar esforços de pesquisa em outras questões do desenvolvimento de software. Para apoiar a compreensão da DT em nível de código, também é necessário, por exemplo, investigar estratégias para mitigar os motivos gerenciais que levam equipes de software a não pagarem itens de dívida. Outro tópico interessante para investigação seria a relação entre fatores humanos do desenvolvimento de software e a DT.

4.4 AMEAÇAS À VALIDADE

Como em qualquer estudo experimental, existem ameaças à validade neste trabalho (WOHLIN *et al.*, 2012). As ameaças foram removidas quando possível ou seus efeitos foram mitigados quando a remoção não foi possível.

Em relação à validade externa, o estudo focou em profissionais do setor e buscou obter uma variedade de participantes quanto ao nível de experiência e ambiente de trabalho. Mesmo assim, não é possível generalizar os resultados deste estudo. Para fortalecer a validade externa, as etapas futuras desta pesquisa incluem a expansão dos dados coletados por meio de outras replicações do projeto *InsighTD*.

Em relação à validade interna, o questionário poderia trazer ameaças ao estudo. Conforme discutido em (RIOS *et al.*, 2020), o questionário passou por uma série de validações (três internas e uma externa) e um estudo piloto para identificar eventuais problemas antes de sua execução. Também vale ressaltar que os participantes poderiam agir de forma diferente, o que costuma ocorrer porque fazem parte de um estudo. Para evitar isso, foi explicado claramente o objetivo do estudo e foi pedido aos participantes que respondessem às perguntas com base em sua própria experiência. Também foi deixado explícito que o questionário era anônimo, e que os dados coletados seriam analisados sem considerar a identidade dos participantes. Além disso, os participantes poderiam ter interpretado mal o uso dos termos prevenção e pagamento da DT. Para investigar se esta ameaça se manifestou, todas as respostas sobre como os participantes evitaram e pagaram o item da dívida foram analisadas (Q23 e Q27) para verificar se houve respostas inválidas. Uma alta proporção de respostas inválidas significaria que as perguntas poderiam ter sido mal interpretadas. Ao final, não foi identificada nenhuma resposta inválida, indicando que esta ameaça não afetou o estudo.

A principal ameaça à validade da conclusão está relacionada à análise qualitativa realizada. Para mitigá-la, as análises foram realizadas separadamente por dois pesquisadores, e o consenso foi realizado por um terceiro, mais experiente. Também, procedimentos adicionais foram considerados para buscar consistência na nomenclatura usada por cada equipe de replicação durante sua codificação de atividades. Por último, a classificação dos elementos de gerenciamento da DT em código/não-código, bem como a definição de suas categorias, foram tarefas essencialmente subjetivas. Para mitigá-las, foi seguido um rigoroso procedimento de análise. O processo de classificação sempre foi realizado individualmente em duplas, sendo revisado por, pelo menos, um pesquisador experiente.

4.5 CONSIDERAÇÕES FINAIS

Este capítulo apresentou o ponto de vista dos profissionais da indústria de software sobre a forma de incidência da DT em todo o processo de desenvolvimento de um software, indicando que as causas, efeitos, práticas de prevenção, razões para não prevenir, práticas de pagamento e razões para o não pagamento da dívida estão mais relacionadas a outras etapas do desenvolvimento do que a codificação.

No capítulo seguinte, são apresentadas as considerações finais do trabalho realizado.

CAPÍTULO 5 – CONSIDERAÇÕES FINAIS

Neste capítulo são apresentadas as considerações finais sobre o trabalho realizado, incluindo suas principais contribuições, os resultados obtidos e os próximos passos a serem realizados com a continuidade desta pesquisa.

5.1 CONSIDERAÇÕES FINAIS

O termo dívida técnica contextualiza os efeitos que podem existir ao se utilizar um caminho mais rápido e fácil durante o processo de desenvolvimento de um software com o objetivo de aumentar a produtividade e reduzir o tempo de entrega do produto ao cliente, em detrimento de segurança, estabilidade e escalabilidade futuros. É um campo de estudo que vem crescendo consideravelmente, pois o conceito de DT é considerado de fácil entendimento tanto por pesquisadores quanto por profissionais da indústria de software.

Vários estudos têm sido conduzidos na área para identificar como os profissionais de software identificam, gerenciam, monitoram e quantificam os itens da DT existentes nos projetos. Entretanto, o foco desses estudos tem sido na perspectiva da incidência da DT ocasionada por questões relacionadas a codificação. Considerar apenas essa perspectiva de análise é uma limitação que traz riscos para projetos de software. Dessa forma, os estudos sobre a DT precisam abranger diferentes atividades de um processo de desenvolvimento, como documentação, requisitos e testes.

Este trabalho apresentou a análise da relação entre diferentes preocupações do desenvolvimento de um software e seis elementos relacionados à gestão da DT: causas, efeitos, práticas preventivas, razões para não prevenir, práticas de pagamento e razões para não pagar. Para alcançar os resultados, um subconjunto dos dados coletados pelo projeto *InsighTD*, uma família de pesquisas distribuídas globalmente em causas, efeitos e gestão da DT foi utilizada como base.

A pesquisa seguiu a seguinte linha de trabalho: (i) consolidação dos dados de *InsighTD*, o que possibilitou a identificação das variáveis base do estudo e (2) análise dos dados, com base nas análises das respostas do *survey* foi possível responder às questões da pesquisa.

5.2 RESULTADOS

Ao analisar as respostas dos profissionais da indústria de *software*, foi possível observar que os resultados indicam que as questões de codificação relacionadas às causas, efeitos, prevenção, não prevenção, pagamento e não pagamento da DT, estão de fato mais comumente relacionadas a outras fases do desenvolvimento de software.

O gráfico de radar apresentado na Figura 2 apresenta as porcentagens da distribuição das respostas dos participantes a cada um dos elementos investigados em relação às categorias codificação e outros problemas de desenvolvimento de software. Sobre as causas da DT, foram identificadas 96 que levam à ocorrência da DT, com o total de 1695 citações. Desse total, aproximadamente 92% estavam relacionados a outros problemas de desenvolvimento, enquanto apenas 8% estavam relacionados ao código. Sobre os efeitos, os profissionais relataram um total de 73 efeitos da DT em projetos, com um total de 980 citações. Entre eles, aproximadamente 64% estão relacionados a outras questões de desenvolvimento e 36% estão relacionados à codificação.

Em relação à prevenção, os resultados mostram um total de 89 práticas de apoio à prevenção de itens da DT, resultando em 819 citações. A partir disso, 84% são itens relacionados a outros problemas de desenvolvimento, enquanto apenas 16% estão associados ao código. Os participantes também relataram as razões existentes para não prevenir a DT. Foram relatadas 25 razões que levam à não prevenção de itens da DT, resultando em 63 citações. Destes, 87% estão relacionados a outras questões de desenvolvimento, enquanto apenas oito 13% estão relacionados à codificação.

Sobre a variável pagamento, foram identificadas 32 práticas de pagamento da DT, resultando em 315 citações. Delas, 56% estão relacionadas a outros problemas de desenvolvimento, enquanto 44% estão associadas ao código. Diferente das outras variáveis, essas porcentagens têm pequena diferença, indicando que os problemas de codificação desempenham um papel fundamental nas iniciativas de reembolso da DT. O estudo também demonstrou as razões relatadas pelos participantes para não realizar o pagamento da DT. Foram identificadas 27 razões para não pagar um item da DT, totalizando 319 citações. Destes, 99,7% estão relacionados a outras questões de desenvolvimento e apenas a falta de acesso ao código do componente (0,3%) está associada ao código.

Algumas categorias como gerenciamento da DT, planejamento e gerenciamento, fatores humanos, engenharia de requisitos, foram definidas para facilitar a análise conjunta dos resultados. Por exemplo, após classificar cada causa informada, foi possível observar que a

categoria planejamento e gestão destacou-se com aproximadamente 47% das citações, representando mais de três vezes as citações da segunda categoria classificada. Isso é uma indicação de que as causas da ocorrência de TD estão fortemente relacionadas a questões de gerenciamento de projetos.

Após a análise de todas as classificações foi possível perceber que existem muitas outras questões que precisamos estar atentos ao lidar com DT em projetos de software, principalmente, planejamento e gestão. Muito sobre o não pagamento da DT pode ser entendido olhando para essas questões. Os fatores humanos também chamam a atenção, indicando claramente que a DT, mais do que aspectos técnicos do desenvolvimento de software, também diz respeito à moral da equipe, satisfação, motivação, comunicação e comprometimento.

Outros problemas comumente encontrados em vários elementos do gerenciamento da DT são problemas de arquitetura, problemas de *design*, documentação, problemas de conhecimento, problemas de processo, engenharia de requisitos e VV&T.

Os resultados apresentados nesta dissertação foram reportados em dois artigos já publicados:

- C. BERENGUER; A. BORGES; S. FREIRE; N. RIOS; and R. SPINOLA. Investigating the Relationship between Technical Debt Prevention and Software Development Activities: A Survey with Professionals. In The Americas Conference on Information Systems (AMCIS), 2021, Toronto.
- C. BERENGUER, A. BORGES, S. FREIRE, N. RIOS, R. RAMAC, V. MANDIC, N. TAUSAN, N. RIOS, S. FREIRE, B. PEREZ, C. CASTELLANOS, D. CORREAL, A. PACHECO, G. LOPEZ, D. FALESSI, C. SEAMAN, C. IZURIETA, AND R. SPÍNOLA. Technical Debt is not Only about Code and We Need to be Aware about It. In the 20th Brazilian Symposium on Software Quality. 2021.

O artigo aceito no SBQS foi premiado com o Distinguished Paper Award, sendo convidado a ser estendido para o Journal of Software Engineering Research and Development (JSERD), cuja submissão foi efetuada em março deste ano.

5.3 CONTRIBUIÇÕES

A partir desta dissertação, foi evidenciado que a DT não é apenas sobre o código fonte. É necessário estar atento a outras etapas importantes do processo de desenvolvimento de

software que podem ser impactadas por sua presença. O uso combinado dos diagramas e os resultados detalhados contribuem para uma orientação abrangente para equipes de desenvolvimento de software sobre o que esperar da presença da DT e como reagir a eles considerando vários problemas de desenvolvimento de software.

Além disso, foi indicada a importância da definição de estratégias e ferramentas para apoiar o gerenciamento da DT em todas as etapas do projeto de software (gerenciamento do projeto, requisitos e testes). Com isso, pode-se apoiar gestores de projetos na tomada de decisões sobre estratégias que podem ser implementadas para prevenir, monitorar e pagar a DT.

Finalmente, o trabalho também atenta sobre a necessidade de investigar estratégias para mitigar as razões que levam as equipes de software a não prevenir e pagar os itens da dívida.

5.4 TRABALHOS FUTUROS

Como trabalhos futuros, pode-se citar:

- Investigar se o tipo de dívida impacta como os profissionais gerenciam os itens da DT;
- Verificar a relação entre fatores humanos do desenvolvimento de software e a DT;
- Desenvolver um instrumento de gestão da DT englobando o diagrama de *Hump* e os resultados detalhados, e investigar como posicioná-lo em uma estratégia de apoio à gestão de DT; e
- Expandir os resultados incluindo mais dados oriundos de outras replicações do projeto *InsighTD*.

REFERÊNCIAS

ALVES, Nicolli; MENDES, Tiago; MENDONÇA, Manoel; SPÍNOLA, Rodrigo; SHULL, Forrest ; SEAMAN, Carolyn. Identification and management of technical debt: A systematic mapping study. **Information and Software Technology**, v. 70, p. 100-121, 2016.

BERENGUER Clara; BORGES Adriano; FREIRE Savio; RIOS Nicolli; and SPÍNOLA Rodrigo. In-vestigating the Relationship between Technical Debt Prevention and Software Development Activities: A Survey with Professionals. *In: THE AMERICAS CONFERENCE ON INFORMATION SYSTEMS (AMCIS), 2021, Toronto. Proceedings [...]* 2021.

BERENGUER Clara; BORGES Adriano; FREIRE Savio; RIOS Nicolli; RAMAC Robert; MANDIC Vladimir; TAUSAN Nebojša; PÉREZ Boris; CASTELLANOS Camilo; CORREAL Dario; PACHECO Alexia; LOPEZ Gustavo; FALESSI Davide; SEAMAN Carolyn; IZURIETA, Clemente; SPÍNOLA, Rodrigo. Technical Debt is not Only about Code and We Need to be Aware about It. *In: BRAZILIAN SYMPOSIUM ON SOFTWARE QUALITY, 20., 2021. Proceedings [...]* 2021.

BROWN, Nanette; CAI, Yuanfang; GUO, Yuepu; KAZMAN, Rick; KIM, Miryung; KRUCHTEN, Philippe; LIM, Erin; MACCORMACK, Alan; NORD, Robert; OZKAYA, Ipek; SANGWAN, Raghvinder; SEAMAN, Carolyn; SULLIVAN, Kevin; ZAZWORKA, Nico. Managing technical debt in software-reliant systems. *In: FOSER 10: FSE/SDP WORKSHOP ON FUTURE OF SOFTWARE ENGINEERING RESEARCH, 2010. Proceedings [...]* 2010.

CUNNINGHAM, Ward. The WyCash portfolio management system. **ACM SIGPLAN OOPS Messenger**, v. 4, n. 2, p. 29-30, 1992.

FREIRE, Sávio; RIOS, Nicolli; GUTIERREZ, Boriz; TORRES, Darío; MENDONÇA, Manoel; IZURIETA, C.; SEAMAN, Carolyn ; SPÍNOLA, Rodrigo. Surveying Software Practitioners on Technical Debt Payment Practices and Reasons for not Paying off Debt Items. *In: EASE '20: ASSESSMENT IN SOFTWARE ENGINEERING, 2020, Trondheim, Noruega. Proceedings [...]* 2020 p. 210–219 <https://doi.org/10.1145/3383219.3383241>

FREIRE, Sávio; RIOS, Nicolli; MENDONÇA, Manoel; FALESSI Davide, SEAMAN, Carolyn, IZURIETA, Clemente and SPÍNOLA, Rodrigo. Actions and impediments for technical debt prevention: results from a global family of industrial surveys. *In: ACM SAC, 35., 2020. Proceedings [...]* 2020. p.1548–1555.

GUO, Yuepu; SPÍNOLA, Rodrigo; SEAMAN, Carolyn. Exploring the costs of technical debt management—a case study. **Empirical Software Engineering**, v. 21, n. 1, p. 159-182, 2016.

GRIFFITH Isaac; TAFFAHI Hanane; IZURIETA Clemente; CLAUDIO David, 2014. A simulation study of practical methods for technical debt management in agile software development. *In: WINTER SIMULATION CONFERENCE, 2014. Proceedings [...]* 2014. p. 1014-1025.

IZURIETA Clemente; VETRÒ, Antonio; ZAZWORKA Nico; CAI, Yuanfang; SEAMAN Carolyn; SHULL, Forrest. Organizing the Technical Debt Landscape. *In: WORKSHOP ON MANAGING TECHNICAL DEBT (MDT)*, 3., 2012, Zurich. **Proceedings** [...] 2012. p. 23-26, 2012.

KRUCHTEN, Philippe; NORD, Robert; OZKAYA, Ipek 2012. Technical debt: from metaphor to theory and practice. **IEEE Software**, v. 29, n. 6, p. 18 21, dec. 2012. DOI: 10.1109/MS.2012.167

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

MCCONNELL, Scot. Technical Debt. 10x Software Development [Blog]. 2007. Disponível em: <http://blogs.construx.com/blogs/stevemcc/archive/2007/11/01/technical-debt-2.aspx> Acesso em: 20 jan. 2022.

PROJECT MANAGEMENT INSTITUTE. 2017. **A Guide to the Project Management Body of Knowledge: PMBOK Guide**.6. ed. Pennsylvania, The United States, 2017.

RIOS, Nicolli; MENDONÇA, Manoel; SPÍNOLA, Rodrigo. A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. **Information and Software Technology**, v. 102, p. 117-145, 2018.

RIOS, Nicolli; SPÍNOLA, Rodrigo; MENDONÇA, Manoel; SEAMAN, Carolyn. The practitioners' point of view on the concept of technical debt and its causes and consequences: a design for a global family of industrial surveys and its first results from Brazil. **Empirical Software Engineering**, v. 25, n. 5, p. 3216-3287, 2020.

RIOS, Nicolli; MENDES, Leonardo; CARDEIRAL, Cristina; MAGALHÃES, Ana Patricia; PEREZ, Boris; CORREAL, Darío ; SPÍNOLA, Rodrigo. Hearing the Voice of Software Practitioners on Causes, Effects, and Practices to Deal with Documentation Debt. *In: INTL. CONFERENCE ON REQUIREMENTS ENGINEERING: FOUNDATION FOR SOFTWARE QUALITY*, 26., 2020. **Proceedings** [...] 2020.

RIOS, Nicolli.; MENDONÇA, Manoel; SEAMAN, Carolyn; SPÍNOLA, Rodrigo. The Practitioners Point of View on the Concept of Technical Debt and its Causes and Consequences: A Design for a Global Family of Industrial Surveys and its First Results from Brazil. **Empirical Software Engineering**, v. 1, p. 1, 2020.

ROMBACH, H. Dieter; BASILI, Victor R.; SELBY, Richard W. Experimental Software Engineering Issues: Critical Assessment and Future Directions. *In: INTERNATIONAL WORKSHOP*, 1992. **Proceedings** [...] 1993.

SPÍNOLA, Rodrigo; ZAZWORKA, Nico; VETRÓ Antonio; SEAMAN Carolyn; SHULL Forrest. Investigating technical debt folklore: Shedding some light on technical debt opinion. *In: INTERNATIONAL WORKSHOP ON MANAGING TECHNICAL DEBT (MTD)*, 4., 2013, San Francisco. **Proceedings** [...] 2013. p. 1-7.

STRAUSS, Anselm ; CORBIN Juliet. **Basics of qualitative research**: techniques and procedures for developing grounded theory. Califórnia, EUA: Sage Publications, 1998.

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

TAMBURRI, Damian A.; KRUCHTEN, Philippe, LAGO, Patricia; VLIET, Hans V. Social debt in software engineering: insights from industry. **Journal of Internet Services and Applications**, v. 6, n. 1, p. 1-17, 2015.

WOHLIN, Claes et al. **Experimentation in software engineering**. [*S.l.*]: Springer Science & Business Media, 2012.

ZAZWORKA, Nico; SPÍNOLA, Rodrigo; VETRO, Antonio; SHULL, Forrest; SEAMAN, Carolyn. A case study on effectively identifying technical debt. *In*: EASE 13: INTERNATIONAL CONFERENCE ON EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING, 17., 2013. **Proceedings** [...] 2013. p. 42-47.