

Modelagem Conceitual de Dívida Técnica na Engenharia de Requisitos

José F. Guilhermino da Silva ^[0000-0001-6439-2435] Maria Lencastre ^{2[0000-0002-8032-8801]}
and Jaelson Castro ^{3[0000-0002-4635-7297]}

¹ Programa de Engenharia de Computação, Universidade de Pernambuco, Brasil
{jfgs,mlpm}@ecomp.poli.br
² CIN, Universidade Federal de Pernambuco, Brasil
jbc@cin.ufpe.br

Resumo. Contexto: Dívida técnica (DT) surge de vários fatores, como processos, decisões, ações, ou sua ausência, e pode ocorrer de forma intencional ou não intencional. Incurrir em DT pode proporcionar benefícios de curto prazo, como acelerar a entrega de produtos de *software*; no entanto, não gerenciar a DT pode levar a problemas financeiros e técnicos; também pode criar situações críticas que comprometam o futuro do *software*. Dívida Técnica em Requisitos (DTR) é um tipo de DT. Problema: Apesar da vasta literatura disponível, o conceito de DTR ainda precisa ser melhor compreendido. Objetivo: Nosso objetivo é melhorar a compreensão, disseminação e aplicação de DTR na academia e na indústria. Método: Desenvolvimento de um Modelo Conceitual para DTR baseado em pesquisa bibliográfica e modelos existentes. Resultados: O modelo proposto facilita a discussão e análise de DTR e ajuda a disseminar e aplicar seus conceitos e gestão do seu processo.

Palavras-chave: Dívida técnica. Requisitos. Modelo conceitual.

1 Introdução

A metáfora da dívida técnica (DT) descreve o impacto que artefatos imaturos, produzidos durante o processo de desenvolvimento de *software*, podem trazer um benefício de curto prazo ao projeto em termos de aumento de produtividade e redução de custos. No entanto, essa vantagem inicial pode exigir pagamentos com juros no futuro [1]. O termo DT foi introduzido por Cunningham da seguinte forma [6]: “*O envio inicial de um código é como contrair uma dívida. Um pouco de dívida acelera o desenvolvimento, desde que ela seja paga prontamente com uma reescrita. Os objetos tornam o custo desta transação tolerável. O perigo ocorre quando a dívida não é paga. Cada minuto gasto em um código errado conta como juros sobre essa dívida.*”

O escopo da DT era originalmente limitado a questões de código e artefatos, sendo posteriormente expandido para diferentes estágios de um projeto de desenvolvimento de *software* [22]. Os diversos tipos de DT apresentam natureza distintas, variando conforme o momento em que são adquiridos ou as atividades com as quais estão relacionados. A literatura já delimitou vários desses tipos (de *design*, arquitetura,

documentação, teste, código, defeitos, requisitos, infraestrutura, pessoas, automação de testes, processo, construção, serviço, usabilidade, e versionamento), oferecendo uma compreensão mais profunda sobre suas características específicas [12].

Conforme descrito por Barbosa em [5, 16], é importante discutir o gerenciamento da DT no âmbito das práticas de Engenharia de Requisitos (ER), uma vez que essas atividades são inerentemente complexas, desempenhando um papel importante na definição do propósito do sistema, combinando diferentes perspectivas das partes interessadas e exercendo influência em várias etapas do ciclo de desenvolvimento de *software*, como codificação e planejamento de testes.

O estudo da metáfora de DT no contexto da ER (DTR) potencializa benefícios ainda maiores em termos, por exemplo, de prevenção de custos [8]. Como os defeitos introduzidos durante a fase da ER tendem a aumentar aproximadamente por um fator de 10, no processo de desenvolvimento de *software*, a prevenção de custos por meio da tomada de decisão apoiada no gerenciamento de DTR tem um grande potencial.

Ernst foi o primeiro a mencionar a DTR como "*a distância entre a implementação atual e o estado real do mundo*", representando-a por soluções otimizadas para um problema de requisitos [7]. Essa diferença surge de decisões que priorizam benefícios imediatos em detrimento de custos futuros. A DTR vai além da perspectiva documental dos requisitos, abrangendo questões gerais relacionadas aos requisitos, enquanto a DT aplicada à documentação se restringe aos artefatos de requisitos, estendendo o escopo da DTR por todo o processo da ER [8].

Considerando a importância do gerenciamento da DTR durante o ciclo de desenvolvimento de *software*, este trabalho busca aprofundar e ajudar na padronização da compreensão deste fenômeno por meio da criação de um modelo conceitual para a DTR, dando suporte à discussão e difusão dos conceitos envolvidos e permitindo ampliar a sua investigação e ensino.

Para atingir este objetivo, foram examinadas as contribuições na literatura, incluindo: estudos terciários da literatura [12, 22], mapeamentos sistemáticos da literatura (MSL) [1, 15, 20], revisão sistemática da literatura [17], ontologia [13], surveys [2, 5, 9, 21], modelos conceituais [3, 12, 20] e modelos Causa-Efeito [23]. Em seguida, foi realizado um ciclo de discussões entre os autores. Com base nessas análises, nosso Modelo Conceitual DTR foi desenvolvido para ampliar a compreensão da área. Este modelo proporciona uma representação visual dos conceitos, causas, impactos e relações das estratégias de gestão da DTR, ajudando os profissionais a compreenderem, aplicarem e gerenciarem melhor esses conceitos em seus projetos. Assim, o modelo proposto visa organizar, compreender e difundir os conceitos de DTR para a comunidade da ER, permitindo ampliar sua investigação, facilitar sua aplicação no ensino e a criação de novos experimentos e ferramentas. Porém, não se tem a pretensão de exaurir a representação de todos os termos existentes nesta proposta.

No restante deste artigo, a Seção 2 descreve a Fundamentação Teórica sobre DT e DTR. A Seção 3 apresenta a Proposta do Modelo Conceitual de DTR e sua validação. A Seção 4 detalha Trabalhos Relacionados. Por fim, a Seção 5 apresenta Conclusões e Trabalhos Futuros.

2 Referencial Teórico

A DT pode surgir como atalhos durante o desenvolvimento de *software*, resultando em custos adicionais para manutenção e evolução do sistema [21]. Ela pode ser adotada intencionalmente para acelerar o desenvolvimento de novos recursos e obter vantagens comerciais, ou surgir involuntariamente quando a equipe desconhece sua existência, localização e impacto [15]. O conceito descreve os efeitos de criar artefatos incompletos de ES durante os projetos, trazendo benefícios imediatos, mas aumentando os custos ao longo do ciclo de vida [9]. Incluir esses artefatos imaturos pode resultar em atrasos na implementação de mudanças necessárias e desafios para atender aos critérios de qualidade do projeto [1].

A gestão da DT é importante para o sucesso dos projetos de *software*, visando minimizar seu impacto negativo [22] através de atividades que previnem sua ocorrência e garante seu controle. Dentre as atividades para gerenciar a DT estão [15].

- Identificação: Identificar DT resultante de escolhas técnicas intencionais ou não intencionais em um sistema de *software*, utilizando métodos específicos.
- Medição: Quantifica o benefício e o custo de DT identificada, utilizando métodos de estimativa, ou avalia o nível geral de DT em um sistema.
- Priorização: Classifica as DT identificadas com base em critérios predefinidos, auxiliando na determinação de quais itens devem receber prioridade de resolução e quais podem ser adiados para versões futuras.
- Prevenção: Evita a manifestação de possíveis DT.
- Monitoramento: Observa as mudanças de custo e benefício de DT não resolvido ao longo do tempo.
- Reembolso/Quitação: Resolve ou atenua a DT em um sistema de *software* por meio de técnicas como reengenharia e refatoração.
- Documentação: Representa e codifica a DT de maneira uniforme, abordando as preocupações de partes interessadas.
- Comunicação: Torna as dívidas identificadas visíveis para as partes interessadas, possibilitando discussões e gestão adequadas.

Fernández-Sánchez, por sua vez, descreve conceitos associados ao custo da DT: capital da DT, juros da DT e probabilidade associada aos juros da DT [10]. O capital é o valor que tem de ser pago para retificar um item de DT; juros são a estimativa de custo adicional que tem de ser pago ao longo do tempo se o Item DT não for eliminado (retificado); e a probabilidade associada aos juros, que representa a incerteza dos juros, isto é, a probabilidade de pagamento de juros do Item DT.

Através da aplicação da metáfora DT dos Requisitos de Documentação [23] [5] e a ER em geral [7] [14] a comunidade de ER estabeleceu uma base conceitual de DTR, assim como uma perspectiva empírica sobre a DTR na prática. Lenarduzzi & Fucci em [14] classificam a DTR em três tipos: Necessidades incompletas dos *stakeholders*, *Requirement smells* e Implementação incompatível (ver Tabela 1).

Por sua vez, Wattanakriengkrai define que a DTR auto-admitida pode ser definida como comentários no código-fonte criados deliberadamente pelos desenvolvedores

para demonstrar que algumas partes do código estão faltando, incompletas ou não podem satisfazer os requisitos dos clientes [24].

Tabela 1. Tipo de DT de Requisitos (Tipo DTR)

Tipo de DT	Descrição
Tipo-0: Necessidades Incompletas dos Stakeholders	Necessidades Incompletas dos Stakeholders – referem-se à dívida contraída ao negligenciar as necessidades de alguns <i>stakeholders</i> ou de um grupo específico de <i>stakeholders</i> , o que acontece durante a fase de análise de requisitos. Por exemplo, a falta de requisitos de outras <i>stakeholders</i> , por exemplo, negligenciar os requisitos legais. A dívida incorrida quando um engenheiro de requisitos ou analista de negócios formaliza requisitos de usuários em um documento SRS (<i>System Requirements Specification</i>). Usando linguagem natural ou modelos gráficos, podem ser identificados os <i>requirements smells</i> (ex: linguagem subjetiva, advérbios e adjetivos ambíguos, pronomes vagos) ou seja, construções linguísticas que podem indicar uma violação para requisitos de qualidade [19]. Esses <i>smells</i> também existem para outras abordagens de documentação de requisitos - por exemplo, na UML.
Tipo-1: <i>Requirement smells</i>	A dívida incorrida quando os desenvolvedores implementam uma solução inadequada para um problema de requisitos. Este tipo captura a incompatibilidade entre o objetivo dos <i>stakeholders</i> que está no SRS e a implementação real do sistema. Assim, a dívida incorre quando os desenvolvedores implementam uma solução para um problema de requisitos.
Tipo-2: Implementação Incompatível	

Perera, em seu modelo de quantificação de DTR, inclui os conceitos de Passo de Retificação da DTR, associado ao acúmulo de Benefícios de Retificação da DTR, e incorre em um Custo de Retificação da DTR. Por sua vez, o item DTR, se não for quitado, incorre em um Juros de DTR, associado a uma probabilidade. Os juros, por sua vez, podem ser relativos: ao custo da ER relacionado com a DTR; ao custo do novo código com a DTR; ou ao custo do retrabalho associado à DTR [20].

Barbosa conduziu um *survey* com profissionais de *software* sobre DTR e seus elementos [5]. O estudo identificou 55 causas, 33 efeitos, 26 práticas de prevenção, 3 práticas de não prevenção, 18 práticas de reembolso e 16 práticas de não reembolso relacionadas à DTR. O convite para participação no *survey* foi distribuído via *LinkedIn* para grupos de profissionais da indústria, resultando em respostas de 7 países, com 78 respostas relacionadas diretamente à DT. O estudo utilizou dados de 6 replicações do questionário *InsightTD*, focando em um subconjunto específico de elementos de gerenciamento da DT. Rios também explorou a inter-relação entre causas e impactos da DT em seu trabalho [23].

Os elementos relacionados a DT podem ser agrupados em categorias que refletem as principais preocupações em um projeto de *software*: Questões de Desenvolvimento, Fatores Externos, Fatores Internos de Qualidade, Pessoas, Organizacionais e Planejamento e Gestão. Segundo Barbosa, as atividades de prevenção de DTR incluem [5]: Questões de desenvolvimento, exemplo, adoção de boas práticas; Metodologia, exemplo, acompanhamento de mudanças de requisitos e documentação bem definida; Organizacional, exemplo formação; Pessoas, exemplo

disciplina e foco; Planejamento e gerenciamento, exemplo alocação adequada de tarefas, seguir o planejamento do projeto e prazos bem planejados.

3 Proposta do Modelo Conceitual DTR

Para a construção do modelo foi seguido um processo incremental adaptado de [11], abrangendo as etapas de conceituação, modelagem, validação e correção. Na etapa de conceituação, definimos o domínio do modelo, que diz respeito ao DT na Engenharia de Requisitos. A finalidade é dar suporte à compreensão, discussão e difusão dos conceitos envolvidos na área de DTR, permitindo ampliar sua investigação e ensino. Estabelecemos um escopo focado em prover conceitos relacionados à especificação do DTR, suas causas, impactos e ações de reembolso, deixando para trabalhos futuros a exploração de conceitos como técnicas de mensuração, prevenção e ferramentas de suporte ao DTR. Reunimos o conteúdo necessário para a construção do modelo com base em estudos terciários da literatura [12, 22], mapeamentos sistemáticos da literatura [1, 20, 15], revisão sistemática da literatura [17], modelos conceituais [12] [20] e modelos Causa-Efeito [23].

A modelagem foi realizada através de diagramas de classe UML, e para simplificar, omitimos os atributos das classes no diagrama, descrevendo alguns dos mais relevantes no texto. Definimos os usuários do modelo, que podem ser tanto profissionais da área de Engenharia de *Software* quanto pesquisadores, professores e estudantes, visando melhorar a sua compreensão sobre DTR. A validação do modelo conceitual consistiu em uma validação estática. Na etapa de correção foram feitos alguns ajustes no modelo com base nos resultados da validação.

3.1 Modelo Conceitual para Dívida Técnica em Requisitos

O modelo conceitual proposto para DTR é apresentado na Fig. 1. Ele foi fundamentado em um modelo conceitual proposto em [12] que é um modelo de DT em Engenharia de *Software* mais atual e completo que encontramos nas bases científicas, onde a DT é classificada em vários tipos. O modelo também integrou conceitos vindos de [20]. De forma complementar, o modelo inclui as categorias para causas, impactos e ações de reembolso apresentadas em [5, 23].

No Modelo de DTR proposto, o modelo conceitual de DT é recortado e estendido para seus conceitos específicos de ER com ênfase na DTR. Na Fig. 1, destacamos através de caixas cinzas os elementos trazidos de [12] e em caixas brancas os que provêm de nossa proposta, isto é, de dados que emergiram de nossas pesquisas.

Os elementos que provêm de nossa proposta, que refletem dados que emergiram de nossas pesquisas bibliográficas, em especial [20] [23], são descritos a seguir.

- **Etapa da ER:** Atividades de elicitação, análise, especificação e validação [25]; representam a etapa onde a DTR é gerada.
- **Artefatos da ER:** Os diversos artefatos que podem ser criados em um projeto específico durante o processo da ER (por exemplo Use Cases, Estórias de Usuário, Documento de Visão e Escopo, etc).

- Causas DTR:** Aspectos e situações que podem levar à ocorrência da DTR. Elas são variadas, incluindo processos, decisões, ações (ou até a ausência delas) e eventos que desencadeiam a existência dessa DTR. As causas podem ser classificadas em: Causa nas Questões de desenvolvimento, por exemplo, mudanças de requisitos e requisitos complexos; Causa nos Fatores Externos como a pressão; Causa nas Questões de metodologia, como a falta de processo bem definido; Causa nas Questões de Pessoas, como falta de comunicação e comprometimento; Causa nas Questões de Planejamento e Gestão, por exemplo, prazos e gerenciamento não eficaz do projeto.

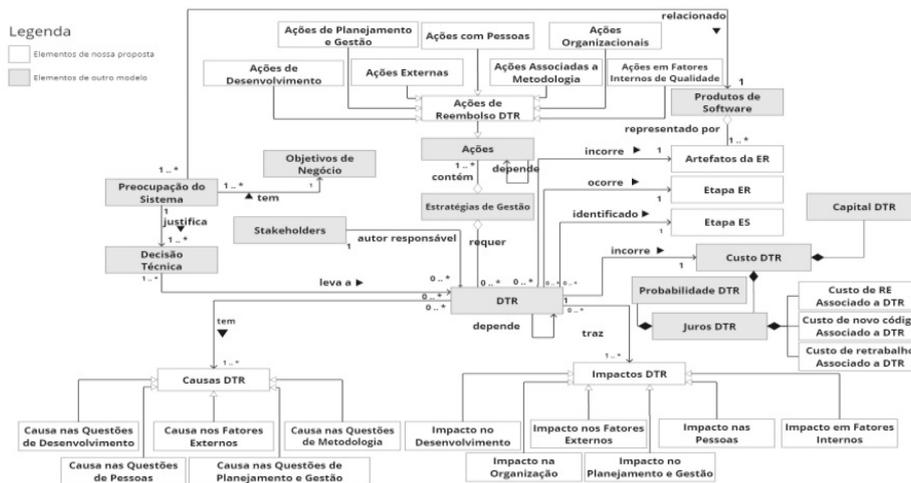


Fig. 1. Proposta de Modelo Conceitual para Dívida Técnica em Requisitos

- Impactos DTR:** podem ser observados em curto ou longo prazo. Um impacto pode ser do tipo positivo ou negativo, dependendo do contexto em que a DTR é originada e das mudanças no ambiente. Os impactos podem ser classificados em: Impacto no Desenvolvimento, por exemplo, mudança de *design*; Impacto nos Fatores Externos, por exemplo, baixa qualidade do produto, projeto não concluído; Impactos em Fatores Internos, por exemplo, código incorreto; Impacto nas Pessoas, por exemplo, estresse, insatisfação, e desmotivação das equipes; Impacto na Organização, por exemplo, perda financeira; Impacto no Planejamento e Gestão, por exemplo, atraso na entrega e retrabalho.
- Ações Reembolso DTR:** variam de acordo com os tipos de DTR e dos objetivos. Devem ser executadas para gerenciar a DTR. Elas podem ocorrer durante a ER, se a DT for identificada mais cedo (antes da implementação), ou durante a implementação se identificadas durante esse estágio. Além disso, Podem ser classificadas como: Ações de Desenvolvimento (por exemplo, alteração de escopo e atualização de documentação), Ações Externas (correção de *bugs*), Ações associadas à Metodologia (por exemplo, implementar ações preventivas de DTR), Ações em Fatores internos de qualidade (por exemplo, refatoração), Ações Organizacionais (por exemplo, mudança do gestor do projeto), Ações com

Pessoas (por exemplo, comunicar ao cliente DTR), Ações de Planejamento e Gestão (por exemplo, aumento de orçamento e prorrogação de prazos).

- **Custo DTR** (de retificação): uma DTR pode ser retificado por meio de uma ação e incorrer em um custo, o Capital. Na DTR esse valor envolve os custos para: formalizar e implementar as necessidades negligenciadas, para corrigir o *requirements smells* no documento de especificação de requisitos, e para comparar a implementação atual com as possíveis alterações [14], [20].
- **Juros DTR**: Uma DTR pode incorrer em Juros, que podem surgir: a) antes da fase de implementação, por exemplo, como um custo extra para esclarecer uma formalização ambígua de um requisito durante a Especificação de Requisitos; b) como um custo extra para conduzir entrevistas adicionais com usuários (custos de ER associados a DTR) ou durante a fase de implementação, como um custo extra para implementar uma solução alternativa para uma implementação incompatível (Custo de Retrabalho ou Custo de Novo Código associado ao DTR) [18].
- **Probabilidade DTR**: estimativa associada ao Juros de DTR, uma vez que pode não fazer diferença se alguns requisitos não forem atendidos; exemplo, itens de DTR pertencentes a um recurso não usado não incorrerão em Juros de DTR [20].

3.2 Avaliação do Modelo Conceitual para Dívida Técnica em Requisitos

A avaliação do modelo conceitual foi uma validação estática, envolvendo a apresentação e análise da solução proposta sem necessidade de utilização direta [26]. Especialistas em modelagem na área de ER analisaram os componentes do modelo, sua utilidade, dificuldade de compreensão, pontos fortes e fracos, corretude, completude, além de sua viabilidade para uso e recomendação.

A validação iniciou com uma primeira apresentação do Modelo Conceitual de DTR em uma sessão presencial com o participante P0 (a única não realizada *on-line*), conforme Tabela 3. Com base nos resultados da avaliação inicial, o modelo foi ajustado, seguido pelas etapas subsequentes do protocolo descritas na Tabela 2.

Nos novos encontros *on-line*, o modelo proposto foi apresentado seguindo um roteiro, sendo precedido pela apresentação dos fundamentos. Devido a conflitos de agenda, não foi possível realizar um grupo focal com todos os especialistas simultaneamente. Foram organizados encontros individuais com o primeiro autor, segundo autor e um especialista, envolvendo os participantes P1, P4, P5, P6 e P7. Os participantes P2 e P3 puderam participar juntos de um grupo focal. Posteriormente, os especialistas responderam individualmente a um questionário *on-line* de forma assíncrona, acessado através de convite por *e-mail* com *link* privado.

Selecionamos especialistas com as seguintes características: (1) Professores e Pesquisadores ativos na comunidade de ER, expostos ao tipo de informação que buscamos, (2) com pelo menos três anos de experiência na área de ER, e (3) dispostos a compartilhar seus conhecimentos. No recrutamento dos participantes, utilizamos amostragem de conveniência [4], recrutando pessoas de nossas redes pessoais, incluindo 2 participantes de Portugal (de 2 universidades) e 6 do Brasil (de 3 estados diferentes e 4 universidades). Ver mais detalhes do perfil dos participantes na Tabela 3. Previamente às entrevistas, compartilhamos com os participantes os objetivos de nossa pesquisa.

Tabela 2. Protocolo para Avaliação do Modelo Conceitual

Etapa	Detalhamento
1.Convite	Notes Foram feitos 10 convites por <i>e-mail</i> a especialistas, e oito aceitaram.
2.Agendamento do Encontro	Os encontros foram agendados para serem realizados online (via <i>Google Meet</i>) nos meses fevereiro e março de 2024, sendo informado o objetivo aos participantes.
3.Envio de Documetação	Foi, então, enviada uma apresentação para os participantes que aceitaram o convite, via e-mail, contendo uma breve explanação de DT, DTR, passos do gerenciamento de DT, principais causas, impactos, e práticas de reembolso de DT, além do modelo conceitual e detalhamento explicativo do seu elementos
4.Realização do Encontro	O encontro foi por videoconferência (<i>Google Meet</i>). Foram apresentados os objetivos e foi feita a apresentação, usando o mesmo material antes enviado. Após a apresentação foi aberto para perguntas e, quando pertinente, foram realizadas discussões com o objetivo de deixar mais claro o conteúdo e dar exemplos. Em média os encontros duraram de 1:30hr; eles foram gravados e depois transcritos.
5.Construção de Questionário	O questionário foi construído no <i>Google Forms</i> . Contendo 12 perguntas.
6.Envio de Questionário	Os questionários foram criados através do <i>Google Forms</i> e um <i>link</i> privado foi disponibilizado para os especialistas.
7.Análise das Respostas	Como a amostra não permite uma análise estatística significativa devido ao número de envolvidos, focamos na análise qualitativa dos dados coletados pelo <i>Google forms</i>

Tabela 3. Perfil dos Participantes

ID	Quantidade de anos no Ensino da ER	Quantidade de anos no Mercado na ER	Quantidade de anos na Pesquisa na ER	Conhecimento Prévio de DT e de DTR(1 a 5)l=min
P0	27	3	31	DT (4) e DTR (4)
P1	22	0	22	DT (4) e DTR (4)
P2	13	0	20	DT (3) e DTR (3)
P3	13	0	15	DT (4) e DTR (4)
P4	13	17	10	DT (4) e DTR (4)
P5	30	30	30	DT (1) e DTR (1)
P6	0	1	14	DT (4) e DTR (3)
P7	3	0	3	DT (4) e DTR (3)

Usamos uma abordagem direcionada na avaliação de aspectos do modelo em questão, considerando a corretude e completude de seus elementos, a adequação das definições, a facilidade de entendimento, a utilidade como artefato no desenvolvimento de sistemas e sua aplicabilidade futura em projetos ou no ensino. Nos concentramos nos *feedbacks* específicos relacionados à DTR, excluindo os

comentários gerais sobre DT Genéricos [12], os resultados detalhados estão disponíveis em [27].

Nos encontros remotos, o primeiro autor deste artigo fez a apresentação (aproximadamente 20 minutos), explicações e anotações. O segundo autor também fez explicações. As perguntas e as respostas mais relevantes estão na Tabela 4. A análise e respostas mais completas podem ser encontradas em [27].

Tabela 4. Table Pontos mais relevantes apontados na avaliação do Modelo Conceitual DTR

ID	Questões (Q) e Principais Respostas dos Participantes (P)
Q1	Os elementos no modelo são pertinentes? É necessário incluir algo, excluir ou alterar?
P0	Poderiam se especializar os <i>stakeholders</i> (e grau de experiência), as ações (com alguns exemplos), e referenciar a maturidade da organização. Também podiam especializar os artefatos de ER
P1	Os elementos são pertinentes. Sugiro exemplos para alguns elementos para evitar ambiguidades e pensar se é necessário incluir aspectos de rastreabilidade para identificar alguns tipos de DTR.
P2	Justificar porque itens foram removidos do modelo de [12]. Explorar o conceito de custo (ex: métricas e probabilidades que podem ser usadas); Incluir ações do processo de ER visam diminuir a DT, tais como validação e verificação, modelagem, rastreamento; Definir classes abstratas e concretas; Definir de atributos. A classe <i>Status</i> não poderia ser um atributo?
P6	Sim, são pertinentes. Mas também são genéricos e se aplicam a vários tipos de problemas, para além de DTR. Senti falta de um ponto central das DTs, que é a noção de juros (<i>interest</i>), ponto chave na definição original de Cunningham, além da noção de ferramentas de ER.
Q2	As correlações apresentadas no modelo são condizentes com a realidade dos projetos?
P0	Sim. Podia incluir para o projeto (número de pessoas envolvidas, etc.)
P1	Os elementos com a caixa branca, sim. Relativamente à maturidade, validar com mais estudos de caso. Analisar a diferença entre Ações e Fatores e se as Pessoas são fatores internos/ externos.
P2	Nas heranças, para diferenciar Causas, Impactos e Ações é suficiente um atributo para o tipo. Para as relações (correlação entre causas, impactos e ações) se a intenção é distinguir quais causas levam a quais ações resolvem quais impactos, então incluir esses relacionamentos.
P3	Para ter certeza seria melhor fazer uma instanciação em um projeto. Um ponto que não ficou claro é a relação entre Tipos DTR com Fases da ER e Ocorrência.
Q3	Os elementos estão corretos?
P4	Sim, os elementos e termos utilizados estão corretos, apresentam clareza e intuição.
Q4	Os elementos apresentam definições adequadas?
P3	Para ter certeza seria melhor fazer uma instanciação em um projeto.
Q5	O modelo é de fácil entendimento?
P2	Médio. Há vários fatores pra isso: o conceito é complexo, o diagrama é grande, o contexto de uso do modelo e seu objetivo não estão totalmente claros para mim.
P4	É facilmente compreensível, abordagem intuitiva e clara em relação à sua aplicabilidade.
P6	Razoavelmente fácil, quando combinado com o texto.

Q6 O modelo é um artefato útil no processo de desenvolvimento de sistemas?

P2 O modelo oferece um *overview* sobre DT em requisitos e estimula a buscar mais informações ou refletir sobre o tema, mas não diretamente usá-lo no processo de desenvolvimento de *software*.

P3 Sim para melhor entender o que é DT no contexto de requisitos.

P4 Sim, o modelo apresentado é um artefato útil no processo de desenvolvimento de sistemas.

P6 Atualmente não, por ainda ser um tanto genérico, mas pode vir a ser em versões futuras.

P7 É uma ferramenta importante para auxiliar o time de desenvolvimento a entender, priorizar e gerenciar efetivamente a qualidade de seus requisitos durante o ciclo de vida do projeto.

Q7 Você utilizaria o modelo em projetos futuros ou no ensino? Justifique, por favor.

P0 Sim, dá uma visão geral. Poderia evoluir para um catálogo.

P1 Sim, é útil em projetos e no ensino, uma vez que ilustra para ambos os elementos da gestão DTR

P2 Sim, poderia tentar usar em projetos de pesquisa ou para ensinar esse assunto aos alunos.

P3 Sim para ensino, como uma forma de apresentar para os alunos o uso de DT na ER.

P4 É valioso no desenvolvimento de sistemas; estou muito satisfeito com a abordagem adotada. Certamente empregaria esse modelo em minha empresa, acreditando que resolveria diversos problemas relacionados à DTR *software*, conforme proposto nesta pesquisa.

P6 Não no estado atual do modelo, vide resposta à questão anterior

P7 Sim. O modelo está bem explicado e direcionado para Requisitos.

Com relação à análise dos resultados, vamos nos deter aqui naquelas respostas referentes às duas últimas questões, dada a sua relevância.

Com relação à questão Q6 – “O modelo é um artefato útil no processo de desenvolvimento de sistemas?”, as respostas foram positivas. O participante P2 afirma que o modelo oferece uma visão geral e estimula a reflexão sobre o tema; já o P3 diz que ajuda na compreensão de DT no contexto de requisitos. P4 afirmou que é útil no desenvolvimento de sistemas; P7 considerou que é importante para ajudar na priorização e gestão da qualidade dos requisitos ao longo do ciclo de vida do projeto. Porém, P6 expressou dificuldades devido à falta de familiaridade com o conceito de DT, mencionando: “Não consigo oferecer uma resposta adequada, pois possuo um conhecimento limitado na área de dívida técnica”.

Com relação à questão Q7- “Você utilizaria o modelo em projetos futuros ou no ensino?”, a maioria dos participantes considerou que o modelo seria útil para o ensino (P1, P2, P3), pesquisa (P2) e em projetos (P4, P7). No entanto, um dos participantes (P6) indicou que, no momento atual, não adotaria o modelo devido à sua natureza ainda um tanto generalista; contudo, ressaltou a possibilidade de se tornar um artefato útil em versões futuras. Devido à relevância dos comentários dos especialistas, o modelo conceitual de DTR, proposto neste estudo, passou por ajustes específicos em relação à versão inicial apresentada aos participantes. Os ajustes foram feitos para atender solicitações, entre elas, a apresentação do objetivo e escopo do modelo, detalhamento dos custos da DTR (Capital, Juros e Probabilidade de Juros) e inclusão de exemplos. Também houve revisão das relações entre os Tipos de DTR com as Fases da ER e os Tipos de Ocorrência. Consideramos que o modelo atendeu seu propósito para a maioria dos participantes, motrando ser útil para apoiar a compreensão, discussão e disseminação dos conceitos relacionados à DTR, ampliando seu uso, pesquisa e ensino.

4 Trabalhos Relacionados

Identificamos 4 trabalhos relacionados à nossa proposta. A Tabela 5 compara esses trabalhos, incluindo nossa proposta na última coluna.

Tabela 5. Trabalhos Relacionados e Modelo Conceitual DTR Proposto

Aspecto	[20]	[12]	[13]	[23]	Proposta
Tipo	Modelo de quantificação de DTR	Modelo Conceitual	Ontologia	Diagramas Causa e Efeito	Modelo Conceitual DTR
Foco	ER	ES	ES	ES	ER
Objetivo	Construção de modelo RTDQM	Sintetizar evidências relacionadas à conceituação, características e gestão de DT	Base para a construção de uma estrutura comum para discussão de DT na ES	Uso de diagramas probabilísticos de causa-efeito para apoio à gestão de DT	Facilitar a compreensão, disseminação, uso e ensino da DTR.
Fundamentação	MSL	Estudo terciário	Análise e revisão da literatura	<i>Survey</i>	Análise e Revisão da Literatura
Validação	Iterações durante o MSL	Baseado em evidências de estudos secundários	Revisões por pares	Uso do modelo TAM	Especialistas em ER

O trabalho de Perera [20] explora a DTR e sua quantificação por meio de uma MSL e do Modelo de Quantificação de DTR (RTDQM). Destaca que a DTR é semelhante à DT de código, mas com componentes únicos. A avaliação usou o *Technology Acceptance Model* (TAM). Os resultados têm o potencial de impactar impactar a prática profissional e a pesquisa acadêmica na área de ER.

Junior e Travassos [12] realizaram um estudo terciário sobre DT em projetos de *software*, analisando 19 estudos que forneceram evidências sobre suas definições, características, gestão, ações e tecnologias. Identificaram causas da DT, como restrições do projeto, decisões técnicas e membros da equipe, além dos impactos no gerenciamento de projetos, equipe, negócios e qualidade do *software*. Por fim, é apresentado um modelo conceitual de DT para apoiar futuras discussões.

O estudo de Kleinwaks [13] propôs uma ontologia para estabelecer uma terminologia unificada na Engenharia de Sistemas, focando em DT. Abordou a falta de definições consistentes e terminologia variada. Destacou a necessidade de uma base comum para discutir riscos sistemáticos e a importância da comunicação eficaz em sistemas complexos e sistemas de sistemas.

O estudo de Rios [23] utilizou diagramas probabilísticos de causa e efeito para organizar informações sobre as causas e efeitos sintetizados do *InsighTD*. Com 105 causas e 85 efeitos de DT identificados, os autores propuseram o uso desses

diagramas entre empresas. Um estudo de caso demonstrou sua eficácia na gestão da DT, facilitando a identificação de causas e efeitos e melhorando agilidade, produtividade e eficácia.

Considerando os trabalhos existentes, nosso modelo conceitual de DTR é complementar aos demais. Um modelo identificado [20] foca na quantificação da DTR, abordando os custos adicionais decorrentes. Em contraste, nossa pesquisa apresenta um modelo conceitual com uma abordagem mais abrangente, visando ampliar a compreensão da DTR e seu gerenciamento ao incorporar causas, impactos e estratégias de gestão. Adicionalmente, buscamos aderência aos conceitos de um modelo existente de DT ES e incorporamos elementos de [20].

5 Conclusões e Trabalhos Futuros

A principal contribuição deste trabalho inclui a criação de um Modelo Conceitual de DTR com o objetivo de proporcionar uma maior organização e compreensão dos termos pela comunidade da ER, facilitando sua aplicação e ensino. A construção do modelo conceitual foi conduzida por um levantamento criterioso da literatura, tendo se baseado numa proposta de Modelo Conceitual de DT para a ES. Assim, esse modelo abrange conceitos fundamentais, propriedades e preocupações centrais relacionadas à compreensão e ao gerenciamento da DTR. Essa abordagem não apenas amplia o entendimento teórico do campo, mas também fornece uma base para a aplicação prática em contextos de desenvolvimento de *software*.

A avaliação do modelo foi realizada por meio de reuniões com especialistas ativos na comunidade da ER e resposta a questionários. Observam-se resultados favoráveis quanto à aceitação, eficiência e aplicabilidade do modelo conceitual proposto. Assim, acredita-se que os objetivos delineados tenham sido alcançados. Para discutir as ameaças à validade, usamos diretrizes propostas por [26] e ampliadas por [18].

- Validade externa. O tamanho da amostra de participantes foi pequena, limitando o resultado do estudo. Contudo, consideramos que foi um ponto de partida da avaliação do modelo. Buscamos minimizar esta ameaça incluindo participantes com vários anos de experiência na pesquisa e ensino da ER e com variações demográficas, dando maior representatividade e elegibilidade dos participantes.
- Validade da conclusão. Procuramos mitigar esta ameaça alinhando a literatura acadêmica em DTR, baseada em RSL e *surveys* com diferentes comunidades. O segundo e terceiro autores revisaram o modelo se baseando em sua especialidade em ER e na realização de projetos práticos.
- Validade interna. Os instrumentos foram elaborados seguindo as diretrizes de [17]. Foi realizado um piloto antes para garantir a compreensão das perguntas. Dividimos o questionário em 2 partes para evitar sobrecarga nos participantes e preservar a qualidade de suas respostas (limitamos o número de perguntas em cada Seção). Porém, pode se identificar que ocorreram dúvidas, por exemplo, a questão 2 parece que não estava clara, impactando nos resultados. Surgiram nas respostas comentários como: "Não consegui enxergar correlações no modelo apresentado." (P1); "Seriam as relações (associações, heranças, agregações...)?

Ou seriam as relações (correlação entre causas, impactos e ações)?”; “Se você realmente se referia a correlações, eu não consegui captá-las." (P3)

- Validade interpretativa. As respostas às perguntas abertas da pesquisa foram codificadas pelo primeiro autor e validadas pelo segundo autor para garantir uma inferência confiável das opiniões dos participantes. Para evitar a perda de aspectos relevantes, o terceiro autor realizou uma leitura adicional.

Como trabalhos futuros, planeja-se aprimorar o modelo conceitual incorporando sugestões específicas, como evoluir o modelo para um catálogo, detalhar a mensuração de custos e suas ferramentas, e explorar ações de prevenção e pagamento de DTR. Al+em disso, o desenvolvimento de uma ferramenta baseada no Modelo Conceitual DTR, e sua aplicação em projetos acadêmicos e na indústria de *software*.

Agradecimentos. Este trabalho foi parcialmente suportado pela Fundação de Amparo à Ciência e Tecnologia de PE (FACEPE), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) código de financiamento 001, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Referências

1. Alves, N. S., Mendes, T. S., De Mendonça, M. G., Spínola, R. O., Shull, F., & Seaman, C. (2016). Identification and management of technical debt: A systematic mapping study. *Information and Software Technology*, 70, 100-121.
2. Alves, N. S. R. (2021). Organização de um conjunto de descobertas experimentais sobre causas e efeitos da dívida técnica através de uma família de surveys globalmente distribuída.
3. Avgeriou, P., Kruchten, P., Ozkaya, I., & Seaman, C. (2016). Managing technical debt in software engineering (dagstuhl seminar 16162).
4. Baltes, S., & Ralph, P. (2022). Sampling in software engineering research: A critical review and guidelines. *Empirical Software Engineering*, 27(4), 94.
5. Barbosa, L., Freire, S., Rios, N., Ramač, R., Taušan, N., Pérez, B., ... & Spínola, R. (2022). Organizing the TD management landscape for requirements and requirements documentation debt. UMBC Faculty Collection.
6. Cunningham, W. (1992). The WyCash portfolio management system. *ACM Sigplan Oops Messenger*, 4(2), 29-30.
7. Ernst, N. A. (2012, June). On the role of requirements in understanding and managing technical debt. In 2012 Third International Workshop on Managing Technical Debt (MTD) (pp. 61-64). IEEE.
8. Frattini, J., Fucci, D., Mendez, D., Spinola, R., Mandić, V., Taušan, N., ... & Gonzalez-Huerta, J. (2023). An initial theory to understand and manage requirements engineering debt in practice. *Information and Software Technology*, 159, 107201.
9. Freire, S., Rios, N., Mendonça, M., Falessi, D., Seaman, C., Izurieta, C., & Spínola, R. O. (2020, March). Actions and impediments for technical debt prevention: results from a global family of industrial surveys. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing* (pp. 1548-1555).
10. Fernández-Sánchez, C., Garbajosa, J., Vidal, C., & Yagüe, A. (2015, May). An analysis of techniques and methods for technical debt management: a reflection from the architecture

- perspective. In 2015 IEEE/ACM 2nd International Workshop on Software Architecture and Metrics (pp. 22-28). IEEE.
11. Hernández, J. E., Mula, J., & Ferriols, F. J. (2008). A reference model for conceptual modelling of production planning processes. *Production Planning and Control*, 19(8), 725-734.
 12. Junior, H. J., & Travassos, G. H. (2022). Consolidating a common perspective on Technical Debt and its Management through a Tertiary Study. *Information and Software Technology*, 149, 106964.
 13. Kleinwaks, H., Batchelor, A., & Bradley, T. H. (2023). An Ontology for Technical Debt in Systems Engineering. *IEEE Open Journal of Systems Engineering*.
 14. Lenarduzzi, V., & Fucci, D. (2019, September). Towards a holistic definition of requirements debt. In 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (pp. 1-5). IEEE.
 15. Li, Z., Avgeriou, P., & Liang, P. (2015). A systematic mapping study on technical debt and its management. *Journal of Systems and Software*, 101, 193-220.
 16. de Melo, A. C. C., Fagundes, R., Lima, J. V. V., Alencar, F., & Santos, W. (2021). Identificação e Mensuração da Dívida Técnica de Requisitos: um survey na indústria de software. In *Anais do WER21-Workshop em Engenharia de Requisitos*.
 17. Melo, A., Fagundes, R., Lenarduzzi, V., & Santos, W. B. (2022). Identification and measurement of Requirements Technical Debt in software development: A systematic literature review. *Journal of Systems and Software*, 194, 111483.
 18. Molléri, J. S., Petersen, K., & Mendes, E. (2020). An empirically evaluated checklist for surveys in software engineering. *Information and Software Technology*, 119, 106240.
 19. Nascimento, R., Guimarães, E., & Lucena, M. (2021). Requirements Smells como Indicador de Qualidade para Histórias de Usuários: Estudo Exploratório. In WER.
 20. Perera, J., Tempero, E., Tu, Y. C., & Blincoe, K. (2023, September). Quantifying Requirements Technical Debt: A Systematic Mapping Study and a Conceptual Model. In 2023 IEEE 31st International Requirements Engineering Conference (RE) (pp. 123-133). IEEE.
 21. Ramač, R., Mandić, V., Taušan, N., Rios, N., Freire, S., Pérez, B., ... & Spínola, R. (2022). Prevalence, common causes and effects of technical debt: Results from a family of surveys with the IT industry. *Journal of Systems and Software*, 184, 111114.
 22. Rios, N., de Mendonça Neto, M. G., & Spínola, R. O. (2018). A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. *Information and Software Technology*, 102, 117-145.
 23. Rios, N., Spínola, R. O., Mendonça, M., & Seaman, C. (2019, May). Supporting analysis of technical debt causes and effects with cross-company probabilistic cause-effect diagrams. In 2019 IEEE/ACM International Conference on Technical Debt (TechDebt) (pp. 3-12). IEEE.
 24. Wattanakriengkrai, S., Maipradit, R., Hata, H., Choetkiertikul, M., Sunetnanta, T., & Matsumoto, K. (2018, December). Identifying design and requirement self-admitted technical debt using n-gram idf. In 2018 9th International Workshop on Empirical Software Engineering in Practice (IWESEP) (pp. 7-12). IEEE.
 25. Wiegers, K. E., & Beatty, J. (2013). *Software requirements*. Pearson Education.
 26. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.
 27. Silva, J. (2024). Aplicação do conceito de dívida técnica de requisitos em projetos acadêmicos: Uma abordagem para uma formação alinhada com o mercado de trabalho (Dissertação de mestrado). Programa de Engenharia da Computação da UPE, Pernambuco.