

UNIVERSIDADE FEDERAL FLUMINENSE

LUIZ CARLOS DA FONSECA LAGE

**Dívida Técnica de Usabilidade em Projetos de
Software: um Estudo de Casos Múltiplos**

NITERÓI

2019

UNIVERSIDADE FEDERAL FLUMINENSE

LUIZ CARLOS DA FONSECA LAGE

Dívida Técnica de Usabilidade em Projetos de Software: um Estudo de Casos Múltiplos

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Engenharia de Sistemas e Informação.

Orientador:

Prof. Dra. DANIELA GORSKI TREVISAN

NITERÓI

2019

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

L174d Lage, Luiz Carlos da Fonseca
Dívida Técnica de Usabilidade em Projetos de Software: um
Estudo de Casos Múltiplos / Luiz Carlos da Fonseca Lage ;
Daniela Gorski Trevisan, orientadora. Niterói, 2019.
91 f.

Dissertação (mestrado)-Universidade Federal Fluminense,
Niterói, 2019.

DOI: <http://dx.doi.org/10.22409/PGC.2019.m.49045997649>

1. Dívida Técnica. 2. Dívida de usabilidade. 3.
Heurísticas de Nielsen. 4. Estudo de Caso. 5. Produção
intelectual. I. Trevisan, Daniela Gorski, orientadora. II.
Universidade Federal Fluminense. Instituto de Computação.
III. Título.

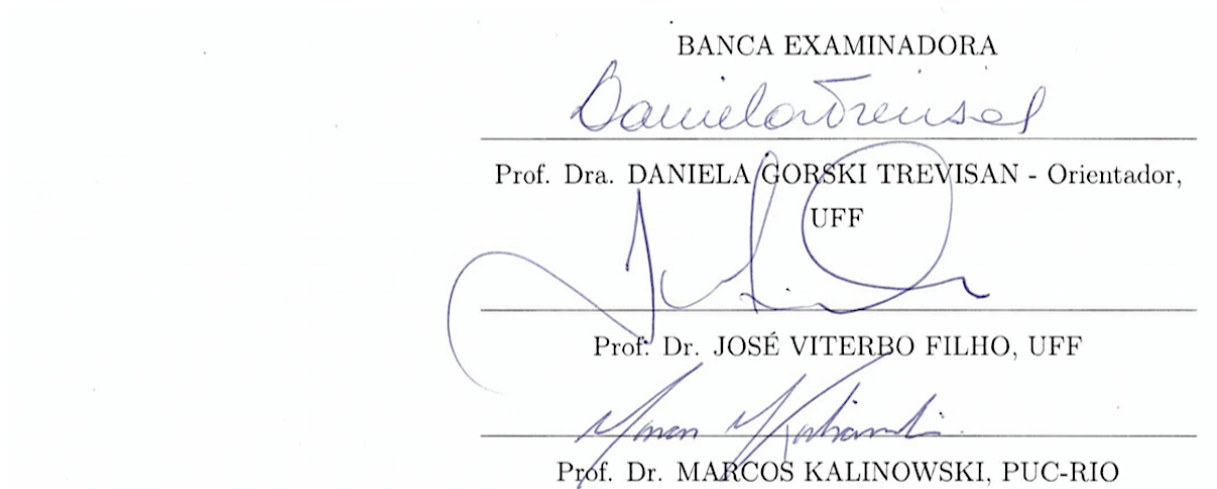
CDD -

LUIZ CARLOS DA FONSECA LAGE

Dívida Técnica de Usabilidade em Projetos de Software: um Estudo de Casos Múltiplos

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Engenharia de Sistemas e Informação.

Aprovada em 05 de 2019.



Niterói

2019

"Eu sou parte de uma equipe. Então quando venço, não sou eu apenas quem vence. De certa forma termino o trabalho de um grupo enorme de pessoas."

Ayrton Senna.

Agradecimentos

Projeto é resultante da atividade de planejamento, em função do desejo de superar um problema, atingir um resultado ou aproveitar uma oportunidade.

Em 2016, me lancei em um projeto de Mestrado, seria ali a realização de um sonho, um desafio, a autossuperação ao enfrentar minhas fraquezas, meus medos e superá-los.

Quero aqui registrar meus sinceros agradecimentos e dar crédito àqueles que me apoiaram e participaram desta jornada comigo.

Minha linda esposa Luciana, minha eterna parceira. Meus filhos Fernanda, Vinícius, Luciana e Lucas que são alicerces da minha vida e me enchem de orgulho.

Aos meus queridos amigos Tânia e Humberto Bosaipo, que não mediram esforços para me ajudar, só Deus para lhes recompensar.

Aos meus irmãos Eduardo e Hideraldo que insistiram para eu trocar Belo Horizonte-MG por Cuiabá-MT e que diretamente mudaram meu destino.

Meus parceiros e amigos André Moraes, Cláudio Ferraz, Marcos dos Anjos e Marcos Azevedo que abraçaram minha causa e foram fundamentais na construção deste projeto.

Ao professor Marcos Kalinowski que iniciou minha orientação, e me deu um susto ao dizer que estava indo para outra instituição. Mas não me desamparou, deixou-me nas mãos da querida e competente professora Daniela Trevisan. Esta dupla competente e dedicada me trouxeram até aqui. Muito obrigado à Dani e Kali por tudo.

Um grupo enorme de pessoas atuaram direta ou indiretamente neste projeto, amigos, familiares, empresas parceiras que abriram as portas para a pesquisa, profissionais que me surpreenderam pela diligência e dedicação.

Que Deus abençoe a todos e que ele coloque mais esse crédito na conta de todos vocês.

Resumo

Ao longo dos anos, vários estudos foram conduzidos com o objetivo de compreender o fenômeno da Dívida Técnica (DT) e suas implicações no desenvolvimento de software. A maioria desses estudos se concentram em tipos de DT relacionados à código-fonte. A ausência de estudos empíricos sobre a dívida de usabilidade motivou nossa pesquisa. O objetivo desta pesquisa é fornecer uma caracterização inicial da dívida técnica de usabilidade em projetos de software com relação à sua ocorrência, tipo e esforço de resolução. Realizamos um estudo de casos múltiplos, analisando itens de DT de cinco projetos de software de quatro empresas publicas brasileiras. Após várias etapas de seleção, classificação e validação, identificamos 145 itens de DT nos sistemas de gerenciamento de mudanças, usados nos projetos. A análise desses itens permitiu observar que 13,8% dos itens do DT se referiam à dívida de usabilidade, variando de 10,4% a 20,8% ao longo dos cinco projetos. Os itens de dívida de usabilidade identificados cobrem uma série de problemas de usabilidade relevantes, violando oito das dez heurísticas de usabilidade de Nielsen. Em relação ao esforço para pagar a DT, medido em horas-homem, estimado pelos gerentes de projeto, os itens de dívida de usabilidade apresentaram um baixo esforço, variando de 5,1% a 6,7% do total do esforço estimado nos projetos analisados. Considerando que os itens de DT de usabilidade são frequentes, dizem respeito a questões relevantes de usabilidade e requerem pouco esforço para seu pagamento, propomos que ações para identificar e pagar este tipo de DT devam receber alta prioridade nas estratégias de gestão de DT.

Palavras-chave: Dívida Técnica, Dívida de Usabilidade, Heurísticas de Nielsen, Estudo de Caso.

Abstract

Over the years, several studies were conducted aiming at understanding the Technical Debt (TD) phenomenon and its implications on software development. Most of these studies focus on source code related TD types. The absence of empirical studies on usability debt motivated our research. The goal of this research is to provide an initial usability debt characterization in software projects regarding its occurrence, type, and resolution effort. We conducted a multi-case study, analyzing TD items of five software projects from four Brazilian public companies. After several steps of selection, classification, and validation, we identified 145 TD items in the change management systems used in the projects. The analysis of these items allowed us to observe that 13.8% of the TD items concerned usability debt (ranging from 10.4% to 20.8% in the five projects). The identified usability debt items cover a range of relevant usability issues, violating eight out of the ten Nielsen usability heuristics. Regarding effort for paying the TD, measured in man hours estimated by the project managers for resolving the TD items, usability debt items require a relatively low effort, ranging from 5.1% to 6.7% of the total TD resolution effort in the analyzed projects. Considering that usability TD items are frequent, concern relevant usability issues and require low effort for their payment, we put forward that actions for identifying and paying this type of TD should receive high priority in TD management strategies.

Keywords: Technical Debt, Usability Debt, Nielsen's Heuristics, Case Study.

Lista de Figuras

1.1	Comparativo entre Tipos DT Vinculados à código-fonte (VCF) x Dívidas não vinculadas à código-fonte (NVCF). Adaptado de Rios et al. [33]	4
1.2	Citações publicadas por tipo de DT ao longo dos anos - Comparativo entre Grupo-A (dividas mais comuns) x Grupo-B (dívidas específicas). Adaptado de Rios et al. [33]	4
1.3	Citações por tipo de DT ao longo dos anos - Comparativo entre DT de Design x Usabilidade. Adaptado de Rios et al. [33]	5
2.1	Interseção entre os estudos para a questão de pesquisa sobre tipos de DT. Adaptado de Rios et al. [33]	14
2.2	Modelo para qualidade externa e interna	18
4.1	Quantidade de itens apurados por tipo de DT	39
4.2	Quantidade de itens de DT apurados por projeto	40

Lista de Tabelas

1.1	Citações publicadas por tipo de DT ao longo dos anos. Fonte:Rios et al. [33]	3
3.1	Modelo para lançamento e classificação de itens de Dívida Técnica.	34
4.1	Análise dos itens de Dívida Técnica	39
4.2	Heurísticas violadas pelos itens de dívida técnica de usabilidade.	43
4.3	Totalização os itens de DT de usabilidade - Heurísticas de Nielsen.	44
4.4	Análise do esforço homem-hora.	46
4.5	Relação do esforço por tipo de Dívida de Técnica.	47
A.1	Lista preliminar Projeto-A - parte 1 de 4.	58
A.2	Lista preliminar Projeto-A - parte 2 de 4.	59
A.3	Lista preliminar Projeto-A - parte 3 de 4.	60
A.4	Lista preliminar Projeto-A - parte 4 de 4.	61
A.5	Lista preliminar Projeto-B	62
A.6	Lista preliminar Projeto-C	63
A.7	Lista preliminar Projeto-D - parte 1 de 2.	64
A.8	Lista preliminar Projeto-D - parte 2 de 2.	65
A.9	Lista preliminar Projeto-E	66
B.1	Lista final de itens de DT do Projeto-A	68
B.2	Lista final de itens de DT do Projeto-B	69
B.3	Lista final de itens de DT do Projeto-C	70
B.4	Lista final de itens de DT do Projeto-D - parte 1 de 2	71
B.5	Lista final de itens de DT do Projeto-D - parte 2 de 2	72

B.6	Lista final de itens de DT do Projeto-E	73
C.1	Estimativa de esforço do Projeto-A	75
C.2	Estimativa de esforço do Projeto-C	76
C.3	Estimativa de esforço do Projeto-D - parte 1 de 2	77
C.4	Estimativa de esforço do Projeto-D - parte 2 de 2	78

Lista de Abreviaturas e Siglas

ACM	:	Association for Computing Machinery
ATD	:	Architectural Technical Debt
BI	:	Business Intelligence
CES	:	Coordenação de Engenharia de Software
DT	:	Dívida Técnica
ESEM	:	Empirical Software Engineering and Measurement
IEEE	:	Institute of Electrical and Electronics Engineers
IIE	:	Internet Information Services
MDS	:	Metodologia de Desenvolvimento de Software
MS	:	Mapeamento Sistemático
MVC	:	Model-View-Controller
RNI	:	Representação de Natureza Interna
RUP	:	Rational Unified Process
SEI	:	Sistema Eletrônico de Informação
SLR	:	Revisão Sistemática da Literatura
STI	:	Secretaria de Tecnologia da Informação
TDM	:	Technical Debt Management
TI	:	Tecnologia da Informação

Sumário

1	INTRODUÇÃO	1
1.1	Declaração do problema	5
1.2	Objetivo da pesquisa	6
1.3	Metodologia de pesquisa	6
1.4	Organização da dissertação	8
2	FUNDAMENTAÇÃO TEÓRICA	10
2.1	Introdução	10
2.2	Dívida Técnica em projetos de software	11
2.3	Taxonomia da Dívida Técnica	13
2.4	Definições de usabilidade	17
2.5	Usabilidade e Dívida Técnica	21
2.6	Usabilidade e as heurísticas de Nielsen	22
3	ESTUDO DE CASO	26
3.1	Questões de pesquisa	26
3.2	Seleção dos projetos de software	27
3.2.1	Projetos selecionados da Empresa-1	28
3.2.2	Projeto selecionado da Empresa-2	29
3.2.3	Projeto selecionado da Empresa-3	30
3.2.4	Projeto selecionado da Empresa-4	31
3.3	Procedimentos para coleta de dados	32

3.3.1	Fase-1 - Capacitação da equipe técnica	33
3.3.2	Fase-2 - Seleção preliminar dos itens de Dívida Técnica	33
3.3.3	Fase-3 - Avaliação e validação dos itens de Dívida Técnica	34
3.3.4	Fase-4 - Fechamento da lista de itens de Dívida Técnica	36
3.3.5	Fase-5 - Cálculo do esforço para resolução da Dívida Técnica	36
4	ANÁLISE DOS DADOS COLETADOS E RESULTADOS OBTIDOS	38
4.1	Introdução	38
4.2	Análise dos tipos de Dívida Técnica	39
4.2.1	RQ1: Itens de dívida de usabilidade são comuns nos projetos de desenvolvimento de software pesquisados?	40
4.3	Análise das dívidas técnicas de usabilidade pela ótica das heurísticas de Nielsen	41
4.3.1	RQ2: Que tipos de dívida de usabilidade tendem a ocorrer nos projetos de desenvolvimento de software pesquisados?	44
4.4	Análise do esforço (homem-hora) por tipo de Dívida Técnica	46
4.4.1	RQ3: Qual o esforço para resolver itens de dívida de usabilidade nos projetos de desenvolvimento software pesquisados?	48
5	CONSIDERAÇÕES FINAIS	50
5.1	Introdução	50
5.2	Resumo das constatações	50
5.3	Implicações	51
5.4	Ameaças à validade	51
5.5	Trabalhos futuros	52
	Referências	53
	Apêndice A – ITENS APURADOS NA FASE PRELIMINAR DA PESQUISA.	57

A.1	Lista de itens de dívida técnica selecionados e classificados preliminarmente, por projeto	57
Apêndice B - ITENS APURADOS AO FINAL DA FASE DE COLETA DE DADOS		67
B.1	Lista de itens de dívida técnica, por projeto, apurados ao final da fase de coleta de dados.	67
Apêndice C - ESTIMATIVA DE ESFORÇO (HOMEM-HORA) PARA RESOLUÇÃO DA DÍVIDA TÉCNICA		74
C.1	Apuração das estimativas de esforço de dívida técnica, por projeto.	74

Capítulo 1

INTRODUÇÃO

As equipes de desenvolvimento de software geralmente enfrentam o desafio de entregar produtos de software sob prazos apertados, enquanto tentam manter a qualidade como padrão. Para lidar com restrições de tempo e recursos, os desenvolvedores de software precisam priorizar, concentrando-se apenas em requisitos cruciais e até mesmo pegar atalhos (pular etapas do processo de desenvolvimento) para obterem vantagens no cronograma [17].

Além disso, durante o desenvolvimento, a qualidade do software tende a diminuir quando se considera aspectos como sua estrutura interna, adesão às normas, documentação e facilidade de entendimento para futuras manutenções [33] [24] [31].

Neste contexto, Cunningham [14] apresentou em 1992, uma comparação entre complexidade técnica e dívida e afirmou que: “entregar um código imaturo é como entrar em dívida” esta alusão à dívida financeira, tonou-se uma metáfora, um marco, na engenharia de software, chamada Dívida Técnica (DT).

O conceito de dívida técnica contextualiza os problemas enfrentados durante a evolução do software, considerando as tarefas que não são realizadas adequadamente durante o seu desenvolvimento [29]. Deste modo, os desenvolvedores intencionalmente ou involuntariamente violam as regras e, portanto, a dívida técnica se acumula ao longo do tempo, tornando o sistema mais difícil de manter [15]. Um outro ponto destacado por Tom et al. [42], é que a dívida técnica pode crescer devido à incapacidade dos desenvolvedores de produzirem aplicativos de alta qualidade .

Esse conceito de DT tem sido estendida para se referir a qualquer artefato mal elaborado durante o ciclo de vida [33]. Desta forma, a DT inclui aquelas tarefas internas que você escolha não fazer agora, mas que correm o risco de causar problemas futuros se não

forem efetuadas.

Assim, é possível que a dívida técnica traga um benefício a curto prazo para o projeto em termos de maior produtividade e menor esforço, mas deixando uma dívida que poderá ter de ser ajustada com juros mais tarde [33][22] [37].

Ao longo dos últimos anos, diversos estudos surgiram buscando entender o conceito em si e suas implicações para o desenvolvimento de software, incluindo a geração de evidências e a criação de taxonomia para consolidar o conhecimento [41] [42] [25] [33] [16] [12].

A literatura atual identificou e desenvolveu algumas ferramentas e práticas para conduzir o gerenciamento da dívida técnica (TDM), o problema é a falta de evidências empíricas em um ambiente de desenvolvimento de software real [25].

É importante reunir evidências sobre o DT e o TDM em situações reais de desenvolvimento de software para entender como o TDM é atualmente percebido por equipes de desenvolvimento e usar esse conhecimento para aprimorar os processos e ferramentas existentes [43].

Os poucos estudos empíricos realizados em cenários reais, tornam-se um indicador de que, para algumas áreas, ainda não compreendemos totalmente todos os custos ou benefícios dos indicadores e estratégias de gestão propostos pela DT.

Apesar de já existir uma taxonomia de tipos de DT praticamente consolidada, muitas dessas propostas exigem uma investigação mais profunda [33].

Tom et al. [42] apontam que os atributos de qualidade, mais frequentemente, comprometidos quando os *trade-offs* que introduzem a dívida técnica são os atributos estruturais de extensibilidade, escalabilidade, capacidade de manutenção, adaptabilidade, desempenho e usabilidade.

Zazworka et al. [45] também percebem a existência da DT de usabilidade quando afirmam: "Ferramentas podem suportar a identificação de defeitos e dívida de design num projeto, mas não outros tipos de dívida que foram encontrados pelos desenvolvedores por exemplo: dívida de documentação, testes e usabilidade".

Em um recente e extenso estudo secundário [33], foram avaliados 100 artigos sobre dívida técnica, publicados no período de 2010 a 2014, (ver Tabela 1.1). A partir dos dados apresentados algumas análise podem ser feitas em relação às dívidas de usabilidade.

A dívida de usabilidade foi citada apenas em dois trabalhos, um por Zazworka [45] e

Tabela 1.1: Citações publicadas por tipo de DT ao longo dos anos. Fonte:Rios et al. [33]

TD Type	2006	2010	2011	2012	2013	2014	Total
Design	1	5	8	11	9	8	42
Architecture	0	2	3	11	5	9	30
Documentation	0	2	4	6	4	12	28
Test	0	2	2	8	6	6	24
(Type not specified) Technical debt	0	1	1	5	6	10	23
Code	0	3	1	9	5	3	21
Defect	0	1	5	3	3	5	17
Requirement	0	0	0	2	0	2	4
Infrastructure	0	1	0	1	1	0	3
People	0	0	0	1	0	2	3
Test automation	0	0	0	0	2	1	3
Process	0	0	0	0	2	1	3
Build	0	0	0	1	0	1	2
Service	0	0	0	0	2	0	2
Usability	0	0	0	0	1	1	2
Versioning	0	0	0	0	1	0	1

outro referenciado por Potdar [32], ambos sem maiores argumentações.

Percebe-se um baixo interesse dos pesquisadores sobre a metáfora da DT antes de 2010, com somente 1 artigo publicado (dívida de design) em 2006.

Há uma alta concentração de estudos de tipos de DT vinculados à código-fonte (design, arquitetura, código e defeito), representando 60% das citações (ver Figura 1.1)

Uma possível explicação para isso é que há uma infinidade de ferramentas que executam análise de código-fonte e podem ser usadas para suportar a detecção de DT do código-fonte [33].

Os dados apresentados na Tabela 1.1, podem ser divididos em dois grupos [33]:

- O Grupo-A contém os tipos considerados mais comuns, consistindo dos seis tipos mais citados nos artigos identificados no mapeamento: Dívida de Arquitetura, Dívida de Código, Dívida de Defeito, Dívida de Design, Dívida de Documentação, Dívida de Teste, representam 88% das citações.
- O Grupo-B formado pelos tipos de dívida menos comuns, que os pesquisadores

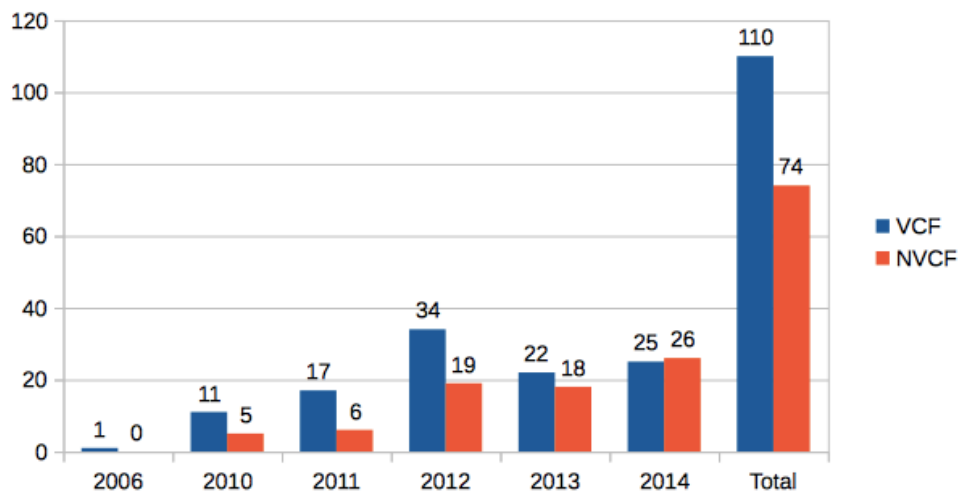


Figura 1.1: Comparativo entre Tipos DT Vinculados à código-fonte (VCF) x Dívidas não vinculadas à código-fonte (NVCF). Adaptado de Rios et al. [33]

chamam de específico apresentando um baixo número de citações em artigos: Dívida de Construção, Dívida de Serviço, Dívida de Infraestrutura, Dívida de Pessoas, Dívida de Processos, Dívida de Automação de Teste, Dívida de Requisitos, Dívida de Usabilidade e Dívida de Versionamento, representam 12% das citações.

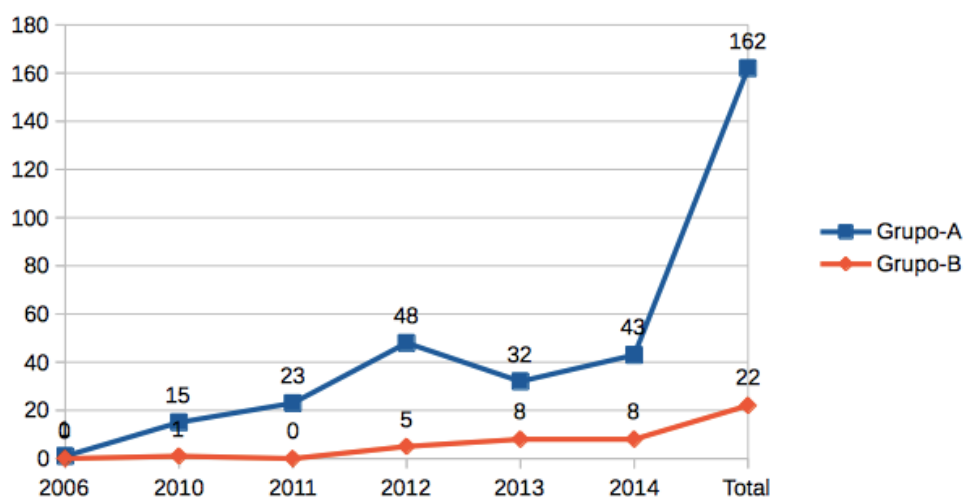


Figura 1.2: Citações publicadas por tipo de DT ao longo dos anos - Comparativo entre Grupo-A (dividas mais comuns) x Grupo-B (dívidas específicas). Adaptado de Rios et al. [33]

Para se ter uma noção da proporção, isso demonstra que a quantidade de citações do Grupo-A é 7 vezes as citações dos tipos de dívidas específicos (Grupo-B) (ver Figura 1.2).

Num comparativo entre a quantidade de citações de dívida de usabilidade e o tipo de dívida mais citada em artigos (dívida de design), foram publicados 42 estudos sobre a

dívida de design enquanto que no mesmo período apenas dois estudos vinculados à dívida de usabilidade (ver Figura 1.3).

Isto demonstra que a dívida de usabilidade é uma área ainda incipiente, pouco explorada num ambiente de pesquisas acadêmicas, no contexto de dívida técnica.

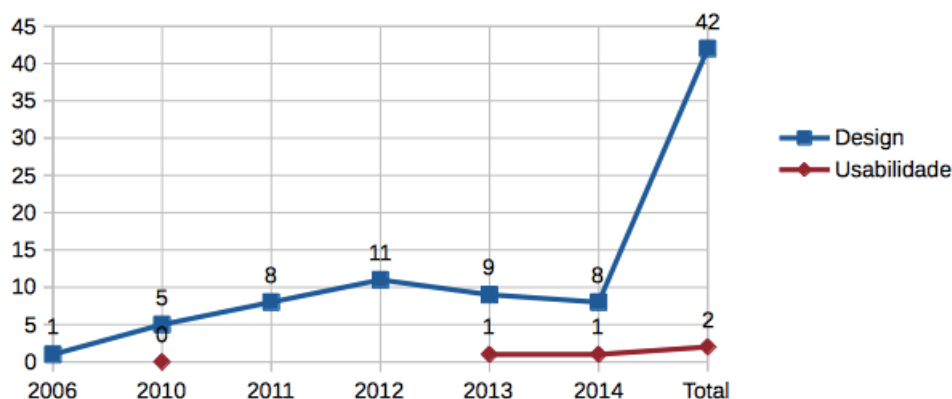


Figura 1.3: Citações por tipo de DT ao longo dos anos - Comparativo entre DT de Design x Usabilidade. Adaptado de Rios et al. [33]

Apesar da utilização das técnicas e conceitos de usabilidade tornarem os sistemas mais fáceis de usar e agradáveis aos usuários, há uma ausência de trabalhos sobre dívida de usabilidade em projetos de desenvolvimento de software. Dar foco a esse fenômeno da dívida de usabilidade em suas perspectivas de ocorrência, natureza e esforço de resolução é a motivação deste trabalho.

1.1 Declaração do problema

Em um recente e extenso estudo terciário Rios et al. [29] identificaram vários tipos de DT. Alguns mais fáceis de identificar através de mecanismos automatizados outros altamente dependentes da percepção humana. Além disso há uma alta concentração nos tipos de dívida relacionados ao código-fonte (design, arquitetura, código e defeito). Uma possível explicação para isso é que há uma infinidade de ferramentas que executam análise de código-fonte e podem ser usadas para suportar a detecção de DT a partir do código-fonte.

De acordo com Rios et al. [33], os tipos DT se expandiram ao longo do tempo e novos foram incluídos, como dívida de serviço, processo, usabilidade e controle de versão. No entanto os trabalhos existentes nesses tipos ainda é muito limitado.

Embora exista um esforço das equipes de desenvolvimento de software para criar

aplicativos fáceis de usar e agradáveis aos usuários finais, não há estudos sobre a dívida de usabilidade em projetos de software [33].

1.2 Objetivo da pesquisa

O objetivo desta pesquisa é investigar o fenômeno da dívida de usabilidade, a fim de lançar alguma luz sobre sua ocorrência, natureza e esforço de resolução. Neste contexto, utilizando uma metodologia específica para trabalhos de estudo de caso [35], em um cenário de projeto de desenvolvimento de software real, corporativo e em plena produção, queremos responder às seguintes Questões de Pesquisa (RQs):

- RQ1: Itens de dívida de usabilidade são comuns nos projetos de desenvolvimento de software pesquisados?
- RQ2: Que tipos de dívida de usabilidade tendem a ocorrer nos projetos de desenvolvimento de software pesquisados?
- RQ3: Qual o esforço para resolver itens de dívida de usabilidade nos projetos de desenvolvimento software pesquisados?

1.3 Metodologia de pesquisa

A metodologia para responder às questões de pesquisa envolveu a realização de um estudo de casos múltiplos em projetos de software a fim de investigar a ocorrência de dívida de usabilidade na prática.

A metodologia do estudo de caso é adequada para muitos tipos de pesquisa de engenharia de software, uma vez que é um método empírico cujos objetos de estudo são fenômenos contemporâneos, em seu contexto natural, difíceis de serem estudados isoladamente [35].

O problema tratado neste trabalho de pesquisa envolve o estudo de um fenômeno contemporâneo (a manifestação de DT de usabilidade) em seu contexto natural (em projetos de desenvolvimento de software).

O estudo de casos múltiplos foi realizado em cinco projetos de quatro empresas públicas brasileiras. Esses projetos dizem respeito a diferentes domínios de aplicações, ta-

manhos de equipe, metodologias, tecnologias de desenvolvimento. Cada estudo de caso envolveu acesso autorizado a sistemas de gestão corporativa e várias reuniões com equipes de projeto, fazendo uso da triangulação de fontes de dados. As diretrizes para conduzir e reportar este estudo de casos múltiplos está de acordo com os propostos por Runeson et al. [35]

Ainda segundo Runeson et al. [35], é relevante levantar a seguinte questão: o que é específico para a engenharia de software que motiva a metodologia de pesquisa especializada? Além das especificidades dos exemplos, as características dos objetos de estudo de engenharia de software são diferentes das ciências sociais e também, em certa medida, dos sistemas de informação.

Seaman et al.[38], corroboram com Runeson et al. [35], quando dizem que a comunidade de pesquisa de engenharia de software tem uma visão pragmática e orientada para resultados sobre metodologia de pesquisa, ao invés de uma posição filosófica.

Runeson et al. [35], estabelecem duas perspectivas a serem alcançadas num trabalho envolvendo estudos de caso: a perspectiva dos objetos de estudo e a perspectiva das etapas principais a serem cumpridas na condução de um estudo de caso. Este estudo de casos múltiplos atendeu plenamente ambas as perspectivas.

Perspectiva dos objetos de estudo:

1. Empresas privadas ou unidades de órgãos públicos que desenvolvem software em vez de órgãos públicos ou corporações privadas usando sistemas de software;
2. Orientado para o projeto em vez de orientado por linha ou função; e
3. O trabalho estudado é um trabalho de engenharia avançada conduzido por pessoas altamente instruídas e não por trabalhos de rotina.

Na perspectiva dos objetos de estudo foram selecionadas empresas públicas nos âmbitos federal e estadual com atuação na área tecnológica. Foram selecionados projetos reais em produção há mais de um ano e todas as empresas selecionadas contam com equipes altamente qualificadas e com larga experiência em desenvolvimento de soluções tecnológicas, e em constante processo de capacitação.

Na perspectiva das etapas do processo, a pesquisa do estudo de casos é conduzida por iteração, em um conjunto de fases. Ao conduzir um estudo de caso, há cinco etapas principais do processo a serem percorridas [35]:

1. Desenho do estudo de caso: objetivos são definidos e o estudo de caso é planejado.
2. Preparação para coleta de dados: são definidos procedimentos e protocolos para coleta de dados.
3. Coleta de evidências: execução da coleta de dados no caso estudado.
4. Análise dos dados coletados.
5. Relatório.

Este estudo de casos múltiplos também cumpre as etapas determinadas por esta metodologia, na perspectiva das etapas do processo, da seguinte forma: as etapas da pesquisa foram bem definidas, houveram inúmeras reuniões com as equipes para alinhamento das informações, modelos padronizados para coleta dos dados, análises e validação dos dados coletados pela equipe técnica e líderes, e apresentação dos resultados.

Um tipo de metodologia de pesquisa não se ajusta a todos os propósitos. Diferentes metodologias de pesquisa servem a diferentes propósitos. Nós distinguimos entre quatro tipos de propósitos para pesquisa baseados na classificação de Robson et al. [34]:

- Exploratório - descobrir o que está acontecendo, buscando novos *insights* e gerando ideias e hipóteses para novas pesquisas.
- Descritivo - retratando uma situação ou fenômeno.
- Explicativo - buscando uma explicação de uma situação ou um problema, principalmente, mas não necessário, sob a forma de uma relação causal.
- Melhorado - tentando melhorar um certo aspecto do fenômeno estudado.

Ancorados nesta classificação podemos inferir que este estudo de casos múltiplos, pode ser classificado como exploratório, descritivo e com caráter quantitativo [35] [36].

1.4 Organização da dissertação

O restante desta dissertação está organizado da seguinte forma.

O Capítulo 2 apresenta a fundamentação teórica a respeito dos tipos de DT em projetos de software e descreve usabilidade dentro dos contextos de Dívida Técnica, das heurísticas de Nielsen e das normas ISO.

No Capítulo 3 o estudo de caso é descrito. Seguindo as diretrizes propostas por Runeson et al.[36], são apresentados as questões de pesquisa a serem respondidas, as empresas e os projetos selecionados para o estudo de múltiplos casos e os procedimentos para coleta de dados

O Capítulo 4 apresenta a análise dos dados, os resultados obtidos e as respostas às questões de pesquisa.

O Capítulo 5 contém as considerações finais, com o resumo das constatações, ameaças à validade, seus impactos e implicações e trabalhos futuros.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados a fundamentação teórica a respeito dos tipos de DT existentes em projetos de software, descreve usabilidade e sua relação com a dívida técnica, aborda definições de usabilidade no contexto das normas ISO e por fim a relação de usabilidade com as Heurísticas de Nielsen utilizada para a classificação dos itens de DT de usabilidade.

2.1 Introdução

O uso das tecnologias da informação se tornou rotineiro nas atividades profissionais, pessoais e de lazer, tornando praticamente impossível de se imaginar a vida sem estas tecnologias. Cada vez mais indivíduos e nações dependem do avanço de sistemas de software no controle de sua infraestrutura, produção industrial, sistema financeiro e entretenimento [39]. Esta percepção ficou latente no final do século XX e consolidou-se já nos primeiros anos do século XXI.

O início da década de 1950 foi um período pioneiro na programação. Os programas não sofrem de desgaste, rasgo, corrosão ou poluição, eles não mudam a menos que e até as pessoas mudá-los, e isso é feito sempre que o comportamento atual de um programa em execução é considerado errado, inadequado ou muito restrito [23].

Lehman et al. [23], já indicava que programas em evolução contínua aumentavam sua complexidade e conseqüentemente deterioram a sua estrutura, a menos que algo fosse feito para mantê-los. Neste contexto, o autor cita que existe uma clara necessidade de metodologias e ferramentas adicionais, uma necessidade que surge principalmente da manutenção do programa, e ressalta com dados históricos estatísticos a relação entre manutenção e

o desenvolvimento de programas: “Do total de gastos dos EUA para 1977, cerca de 70% foram gastos na manutenção do programa e apenas cerca de 30% no desenvolvimento do programa. Esta relação é geralmente aceita pela comunidade de software como característica do estado da arte.”Lehman et al. [23].

Apesar dos grandes avanços na metodologia de programação, o setor ainda enfrenta grandes problemas no desenvolvimento e manutenção de softwares que implementam aplicativos de computador no mundo real (de E-type) [24].

Ward Cunningham [14], apresentou uma comparação entre complexidade técnica e dívida e afirmou que: “entregar um código imaturo é como entrar em dívida”.

Desde então cunhou-se o termo “Dívida Técnica” (DT) em alusão à dívida financeira, que se tonou uma metáfora, um marco, na engenharia de software.

Mesmo após este marco, o desenvolvimento de software continuou acelerado e cada vez mais os processos deixavam de ser manuais e passavam a ser comandados por computadores. Porém, estudos mais contundentes sobre a metáfora da “Dívida Técnica”, só intensificaram uma década após a publicação do artigo de Cunningham, fato relatado por Tom et al. [42] e confirmado por Li et al. [25] onde afirmam que: “somente na última década intensificou-se o interesse pela metáfora da dívida técnica entre os profissionais e a academia”.

2.2 Dívida Técnica em projetos de software

Em um mundo de recursos finitos onde priorização e escolhas são constantemente necessárias, é inevitável que de alguma maneira o processo de desenvolvimento de software seja comprometido, gerando custos ao longo do tempo [42].

A dívida técnica utiliza a dívida financeira como uma metáfora para descrever o fenômeno dos crescentes custos de desenvolvimento de software ao longo do tempo [42] [45].

A dívida técnica é reconhecida como uma questão crítica no setor de desenvolvimento de software: a dívida técnica global foi estimada pelo Gartner em US\$ 500 bilhões em 2010, com o potencial de dobrar em cinco anos [40] [42].

O valor estimado pela Gartner [40] [42], refere-se a uma entrevista intitulada “*Ted Theodoropoulos on Managing Technical Debt Successfully*” conduzida pela professora Carolyn Seaman, em 2012, na qual o entrevistado, Ted Theodoropoulos, afirma que : “A

metáfora, surgiu em torno de 20 anos e as coisas começaram a aquecer realmente no verão de 2010, quando a Gardner disse que havia 500 bilhões de dólares em Dívida Técnica globalmente e que iria crescer para 1,5 trilhões até 2015”.

A partir da publicação desta estimativa, verificou-se, um aumento do volume de pesquisas, fato relatado por Tom et al.[42] e confirmado por Li et al. [25].

Embora a dívida técnica esteja sendo cada vez mais discutida, conforme relatado por trends.google.com, que indica que, nos últimos sete anos, mais e mais usuários do Google têm procurado o termo "dívida técnica", ainda é difícil ter uma ampla compreensão de área porque a informação sobre isso ainda está espalhada na literatura técnica [33].

Dentre as diversas áreas e motivações para pesquisas sobre DT, nos últimos anos diversos artigos relatando revisão sistemática (RS), mapeamentos sistemáticos (MS) e revisão sistemática de literatura (RSL) vinculados a DT tem sido publicados:

- *"A consolidated understanding of technical debt."* [41]. Estudo preliminar, desenvolveu uma revisão sistemática da literatura para entender o estado da pesquisa acadêmica sobre dívida técnica.
- *"An exploration of technical debt."* [42]. Revisão de literatura multivocal para criar uma taxonomia do fenômeno da dívida técnica.
- *"A systematic mapping study on technical debt and its management."* [25]. Mapeamento sistemático visa identificar atividades a serem realizadas na gestão técnica da dívida
- *"Identification and management of technical debt: A systematic mapping study."* [33]. Mapeamento sistemático para identificar os tipos de dívida técnica e os métodos utilizados para a gestão técnica da dívida.
- *"Identification and analysis of the elements required to manage technical debt by means of a systematic mapping study."* [16]. Especificamente interessados nos elementos necessários para gerenciar a dívida técnica.

Além dos já citados, vale ainda ressaltar Ampatzoglou et al. [12] realizaram uma revisão sistemática da literatura para estudar os aspectos financeiros da dívida técnica.

Diversas pesquisas nessa área tem investigado propostas de solução envolvendo ferramentas e práticas para uma efetiva identificação e gerenciamento da DT [33].

De acordo com um recente mapeamento, há uma falta de evidências empíricas sobre o gerenciamento da DT em ambientes de desenvolvimento de software reais [25].

Assim, é importante reunir evidências sobre DT e o seu gerenciamento em situações reais de desenvolvimento de software para entender como o gerenciamento da DT é percebido por equipes de desenvolvimento de software na prática e usar esse conhecimento para aprimorar as propostas de solução existentes [43].

2.3 Taxonomia da Dívida Técnica

Nos últimos anos, diversos trabalhos de pesquisa utilizando revisões e Mapeamentos Sistemáticos (MS) foram publicados. A partir destes trabalhos foi possível definir uma ontologia dos tipos de DT existentes e na medida em que novos estudos foram sendo publicados, novos tipos de DT foram surgindo. Há uma sobreposição parcial entre os estudos. A sobreposição está associada aos tipos de dívida técnica. Entretanto, analisando os resultados de cada estudo de forma mais ampla, é possível observar uma relação complementar entre eles [33]. Dentre estes trabalhos, destacam-se cronologicamente:

- Em 2012 Tom et al.[41] desenvolveram uma estrutura teórica abrangente para facilitar pesquisas futuras. O quadro teórico resultante retrata uma visão holística da dívida técnica, que serviu de referência para outros trabalhos seguintes. Em 2013 Tom et al. [42] ampliaram o trabalho de 2012, desenvolvendo uma revisão de literatura multi-vocal para criar uma taxonomia do fenômeno da DT, com o objetivo de fazer um exame crítico e consolidar a compreensão da natureza da dívida técnica e suas implicações para o desenvolvimento de software. No MS de 2012 foram classificados os seguintes tipos de DT: documentação, requisitos, teste, design, arquitetura, código, infraestrutura e defeito. No mapeamento seguinte, 2013, foi acrescentado um novo tipo de DT: processo.
- Em 2015 Li et al.[25] publicaram um novo MS ampliando o mapeamento de Tom et al. [41], que visa identificar atividades a serem realizadas na gestão da DT, com o objetivo coletar estudos sobre o manejo de DT e fazer uma classificação e análise temática sobre esses estudos, para obter um entendimento abrangente sobre o conceito de DT e uma visão geral sobre o estado atual das pesquisas sobre gestão de dívida técnica (TDM). Neste estudo a DT foi classificada em 10 tipos, foram percebidos dois novos tipos de DT: versionamento e construção. Estes dois novos tipos de DT somam-se aos 8 tipos classificados por TOM et al. [41].
- Em 2016 um novo MS, ainda mais completo, buscou ampliar a ontologia de tipos de DT, este MS visa identificar os tipos de DT e os métodos utilizados para gestão

da DT [33]. Assim, esta taxonomia proposta complementa outras categorizações relevantes de DT realizadas nos últimos anos, acrescentando os seguintes tipos de DT: pessoas, usabilidade, automatização de testes e serviço, (Figura 2.1).

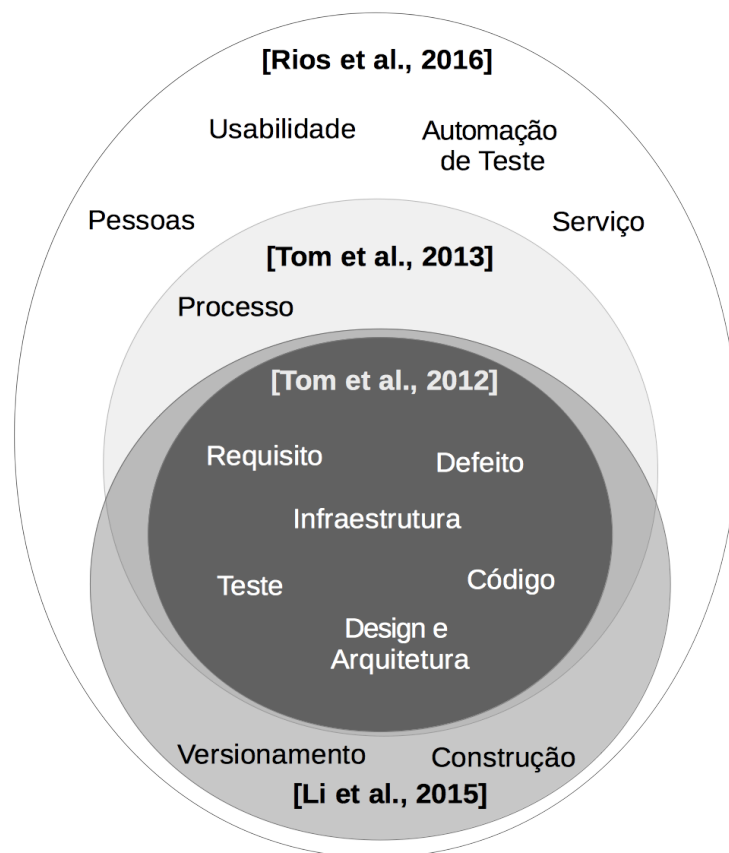


Figura 2.1: Interseção entre os estudos para a questão de pesquisa sobre tipos de DT. Adaptado de Rios et al. [33]

Além destes, outros MS foram desenvolvidos nos últimos anos, com focos bem específicos, dentre eles: o de Fernández-Sánchez et al. [16], interessados especificamente nos elementos necessários para gerenciar a DT e Ampatzoglou et al. [12] cujo objetivo foi analisar os esforços de pesquisa sobre dívida técnica, enfocando seu aspecto financeiro.

Em nosso trabalho utilizaremos, como referência, a taxonomia e definições de DT apresentadas no mapeamento sistemático de Rios et al. [33]:

- **Dívida de design:** Refere-se à dívida que pode ser descoberta através da análise do código-fonte e da identificação de violações dos princípios do bom design orientado a objetos (por exemplo, classes muito grandes ou fortemente acopladas);
- **Dívida de arquitetura:** Refere-se aos problemas encontrados na arquitetura do produto, por exemplo, violação da modularidade, que pode afetar os requisitos arquiteturais (desempenho, robustez, entre outros). Normalmente, este tipo de dívida não pode ser pago com intervenções simples no código, implicando em atividades de desenvolvimento mais extensas;
- **Dívida da documentação:** Refere-se aos problemas encontrados na documentação do projeto de software e pode ser identificada pela procura de documentação faltante, inadequada ou incompleta de qualquer tipo;
- **Dívida de teste:** Refere-se a problemas encontrados em atividades de teste que podem afetar a qualidade dessas atividades. Exemplos desse tipo de dívida são testes planejados que não foram executados ou deficiências conhecidas no conjunto de testes (por exemplo, baixa cobertura de código);
- **Dívida de código:** Refere-se aos problemas encontrados no código-fonte que podem afetar negativamente a legibilidade do código, dificultando sua manutenção. Normalmente, essa dívida pode ser identificada examinando o código-fonte para questões relacionadas a práticas de codificação incorreta;
- **Dívida por defeito:** Refere-se a defeitos conhecidos, geralmente identificados por atividades de teste ou pelo usuário e relatados em sistemas de rastreamento de bugs, que o CCB (*Change Control Board*), ou comitê de controle de configuração, concorda que devem ser reparados mas que devido a prioridades concorrentes e recursos limitados, tem de ser adiados para mais tarde. As decisões tomadas pelo CCB para adiar o tratamento de defeitos podem acumular uma quantidade significativa de DT em um produto, tornando mais difícil corrigi-los mais tarde;
- **Dívida de requisitos:** Refere-se a compensações feitas com relação a quais requisitos a equipe de desenvolvimento precisa implementar ou como implementá-los. Alguns exemplos desse tipo de dívida são: requisitos parcialmente implementados, requisitos implementados mas não para todos os casos, requisitos que são implementados mas que não satisfazem totalmente todos os requisitos não funcionais (por exemplo, segurança, desempenho, etc.);

- **Dívida de infraestrutura:** Refere-se a problemas de infraestrutura que, se presentes na organização de software, podem atrasar ou dificultar algumas atividades de desenvolvimento. Alguns exemplos desse tipo de dívida estão atrasando uma atualização ou correção de infraestrutura;
- **Dívida de pessoas:** Refere-se a questões de pessoas que, se presentes na organização do software, podem atrasar ou dificultar algumas atividades de desenvolvimento. Um exemplo deste tipo de dívida é a especialização concentrada em poucas pessoas, como efeito do atraso no treinamento e / ou contratação;
- **Débito da automação de testes:** Refere-se ao trabalho envolvido na automação de testes de funcionalidades previamente desenvolvidas para suportar integração contínua e ciclos de desenvolvimento mais rápidos. Essa dívida pode ser considerada um subtipo de dívida de teste;
- **Dívida de processo:** Refere-se a processos ineficientes, por exemplo, o que o processo foi projetado para lidar pode não ser mais apropriado;
- **Dívida de construção:** Refere-se a problemas que dificultam a tarefa de construção e consomem tempo desnecessariamente. O processo de construção pode envolver código que não contribui para o valor para o cliente. Além disso, se o processo de compilação precisar executar dependências mal definidas, o processo se tornará desnecessariamente lento. Quando isso ocorre, pode-se identificar a dívida da construção;
- **Dívida de serviço:** Refere-se à seleção e substituição inadequadas de serviços da Web que levam à incompatibilidade dos recursos de serviços e aplicativos. Além disso, esse tipo de dívida também leva à subutilização ou superutilização do sistema integrando um serviço que não utiliza os recursos do sistema da maneira esperada (por exemplo, falta de memória devido a um serviço que não segue o processamento de dados esperado). Processo, ou falta de desempenho devido a um serviço que não usa a memória disponível para a tarefa). Esse tipo de dívida é relevante para sistemas com arquiteturas orientadas a serviços;
- **Dívida de usabilidade:** Refere-se a decisões de usabilidade inadequadas que precisarão ser ajustadas posteriormente. Exemplos dessa dívida são a falta de padrão de usabilidade e inconsistência entre os aspectos de navegação do software;
- **Dívida de versionamento:** Refere-se a problemas na versão do código-fonte, como forks de código desnecessários.

Em relação à DT de usabilidade, foco desta pesquisa, diversos autores reconhecem sua importância na prática. Zazworka et al. [45], por exemplo, afirmam que: "Ferramentas podem suportar a identificação de dívida de defeitos e de design em um projeto, mas não outros tipos de dívida que foram encontrados pelos desenvolvedores por exemplo: dívida de documentação, testes e usabilidade".

Entretanto, não conseguimos encontrar estudos que investigassem esse tipo de dívida técnica, o que torna o presente estudo o primeiro a explorar o fenômeno da dívida técnica de usabilidade na prática.

2.4 Definições de usabilidade

O tema usabilidade ganhou destaque com a publicação da norma NBR ISO/IEC 9126 [3] que foi a primeira norma a definir o termo usabilidade. A seguir outras normas trataram o tema como a norma ISO/IEC 12119:1994 que trata da avaliação de pacotes de software e descreve profundamente as características e subcaracterísticas mencionadas na norma ISO/IEC 9126. Neste contexto de norma aparece também a norma ISO/IEC 9241-11 (1998), que redefine usabilidade como: "a capacidade de um produto ser usado por usuários específicos para atingir objetivos específicos como eficácia, eficiência e satisfação em um contexto específico de uso." Outras normas também trataram o tema usabilidade, porém para o contexto deste trabalho trataremos das normas mais diretamente ligadas à usabilidade.

Além das definições das normas que tratam o tema usabilidade, esta seção explicará a relação existente entre usabilidade e as heurísticas de Nielsen. Relatará como o tema usabilidade tem sido apresentados nos artigos publicas cujo foco é dívida técnica.

As normas ISO que tratam o tema usabilidade:

- ISO/IEC 9126 (1991) - NBR 13596
- ISO/IEC 12119 (1994)
- ISO/IEC 9241-11 (1998)

A NBR ISO/IEC 9126 - Engenharia de software - Qualidade de produto [19], fornece um modelo de propósito geral que define seis características de qualidade de software: Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade, Figura 2.2. Foi a primeira norma a definir o termo usabilidade.

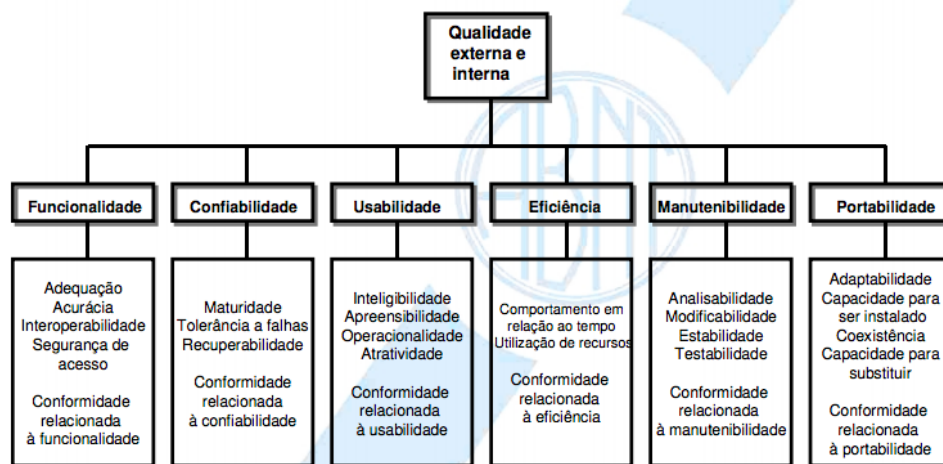


Figura 2.2: Modelo para qualidade externa e interna

O modelo é dividido em duas partes: a) qualidade interna e externa e b) qualidade de uso.

Qualidade interna é a totalidade das características do produto de software do ponto de vista interno. A qualidade interna é medida e avaliada com relação aos requisitos de qualidade interna. Detalhes da qualidade do produto de software podem ser melhorados durante a implementação do código, revisão e teste, mas a natureza fundamental da qualidade do produto de software representada pela qualidade interna mantém-se inalterada, a menos que seja reprojeta.

Qualidade externa é a totalidade das características do produto de software do ponto de vista externo. É a qualidade quando o software é executado, o qual é tipicamente medido e avaliado enquanto está sendo testado num ambiente simulado, com dados simulados e usando métricas externas. Durante os testes, convém que a maioria dos defeitos seja descoberta e eliminada. Entretanto, alguns defeitos podem permanecer após o teste. Como é difícil corrigir a arquitetura do software ou outro aspecto básico do projeto do software, a base do projeto usualmente permanece inalterada ao longo do teste.

Qualidade em uso é a visão da qualidade do produto de software do ponto de vista do usuário, quando este produto é usado em um ambiente e um contexto de uso especificados. Ela mede o quanto usuários podem atingir seus objetivos num determinado ambiente e não as propriedades do software em si.

Segundo a NBR ISO/IEC 9126 [19], usabilidade é a capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições especificadas.

As métricas de usabilidade são definidas no modelo de qualidade externa e interna como sub-características:

- **Inteligibilidade** Capacidade do produto de software de possibilitar ao usuário compreender se o software é apropriado e como ele pode ser usado para tarefas e condições de uso específicas. (É fácil entender os conceitos utilizados?).
- **Apreensibilidade** Capacidade do produto de software de possibilitar ao usuário aprender sua aplicação. (É fácil aprender a usar?).
- **Operacionalidade** Capacidade do produto de software de possibilitar ao usuário operá-lo e controlá-lo. (É fácil operar e controlar a operação?).
- **Atratividade** Capacidade do produto de software de ser atraente ao usuário (Isto refere-se a atributos de software que possuem a intenção de tornar o software mais atraente para o usuário, como o uso de cores e da natureza do projeto gráfico).
- **Conformidade relacionada à usabilidade** Capacidade do produto de software de estar de acordo com normas, convenções, guias de estilo ou regulamentações relacionadas à usabilidade.

A norma ISO/IEC 9126 sofreu três revisões:

- 9126-1, inclui definições e sub-características de qualidade
- 9126-2, descrevem métricas externas, relativas ao uso do produto;
- 9126-3, internas, relativas à arquitetura do produto

Em 2011 a norma ISO/IEC 9126 foi substituída pela norma ISO/IEC/25010, que define modelos de avaliação da qualidade de software e sistemas, acrescentando as características de segurança e compatibilidade [20].

A norma ISO/IEC 12119:1994 - Tecnologia da informação - Pacotes de software-Requisitos de qualidade e testes. Aplicável a pacotes de software, estabelece requisitos para pacotes de software e instruções sobre como testar um pacote de software em relação a esses requisitos. Lida apenas com pacotes de software oferecidos e entregues. Não lida com o processo de produção. O sistema de qualidade de um fornecedor está fora do escopo desta norma [18].

Descreve profundamente as características e sub-características mencionadas na norma ISO/IEC 9126, no tocante a usabilidade, inclui alguns detalhes que devem estar presentes no produto, tais como:

- Consistência de vocabulário entre as margens e a documentação;
- Mensagens de erro com informações necessárias para a solução da situação de erro;
- Diferenciação dos tipos de mensagem: confirmação, consulta, advertência e erro;
- Capacidade de reverter funções de efeito drástico;
- Alertas claros para consequências de uma determinada confirmação;
- Identificação da função do programa que esta sendo executada no momento;

Na avaliação de usabilidade de sistemas interativos, o padrão internacional mais comum é a norma ISO/IEC 9241, que considera mais o ponto de vista do usuário e seu contexto de uso do que as características ergonômicas do produto.

A ISO/IEC 9241-11 (1998), redefine usabilidade como: "a capacidade de um produto ser usado por usuários específicos para atingir objetivos específicos como eficácia, eficiência e satisfação em um contexto específico de uso."

Para um melhor entendimento a norma ISO/IEC 9241-11 definiu alguns conceitos:

- Usuário - pessoa que interage como produto;
- Contexto de uso - usuários, tarefas, equipamentos (hardware, software e materiais), ambiente físico e social em que o produto é usado.
- Eficácia - precisão e completeza com que os usuários atingem objetivos específicos, acessando a informação correta ou gerando os resultados esperados.
- Eficiência - precisão e completeza com que os usuários atingem seus objetivos, em relação a quantidade de recursos gastos;
- Satisfação - conforto e aceitabilidade do produto, medidos por meio de métodos subjetivos e/ou objetivos.

2.5 Usabilidade e Dívida Técnica

No recente e amplo trabalho de mapeamento sistemático apresentado por Rios et al. [33], onde foram avaliados 100 estudos sobre DT no período compreendido entre 2010 e 2014, foram encontrados somente 2 artigos citando dívida de usabilidade, Zazworka et al. [45] e Potdar et al. [32].

Resultados de estudos secundários ajudam a identificar lacunas, capazes de sugerir pesquisas futuras e prover um guia para posicionar adequadamente novas atividades de pesquisa [21].

O artigo de Zazworka et al. [45], teve como objetivo verificar se ferramentas automáticas encontram os mesmos itens de DT que a equipe do projeto e quantificam a sobreposição de itens. Neste contexto chegou-se ao resultado que as ferramentas podem suportar a identificação de dividas de defeitos, design, porém não apontam DT percebidas por desenvolvedores como dividas de documentação, testes e usabilidade.

A única referência feita por Zazworka et al. [45] em relação ao termo DT usabilidade, foi ao afirmar nos resultados da pesquisa: "Note-se que um novo tipo DT, dívida de usabilidade, foi introduzida para descrever a falta de um modelo de interface de usuário comum".

Já o artigo de Potdar et al. [32] utilizaram comentários de código fonte em quatro grandes projetos de software de fonte aberta - Eclipse, Chromium OS, Servidor HTTP Apache e ArgoUML para identificar dívida técnica auto-admitida.

A única referência que faz sobre usabilidade é a citação do trabalho de Zazworka et al. [45], no item trabalhos relacionados.

Para comprovarmos esta situação, durante nossas pesquisas, fizemos uma simulação na biblioteca Scopus, utilizando a mesma string de pesquisa que Rios et al. [33] usou no seu MS ((“Software”) AND (“Technical Debt”)), busca por título do artigo, resumo e palavra-chave, e no período entre 2015 e 2018. Com estes parâmetros a busca trouxe 376 resultados. Entretanto, não é foco deste trabalho analisar o conteúdo desses resultados.

Utilizando os mesmos parâmetros da busca anterior alterando somente a string, ou seja acrescentando o termo "Usabilidade" na pesquisa:

```
((“Software”) AND (“Technical Debt”) and (“Usability”))
```

a pesquisa trouxe apenas 2 resultados.

Um dos resultados foi um artigo publicado em 2015 na 12ª Conferência IEEE / IFIP sobre arquitetura de software, com o título *"Architectural Technical Debt Identification Based on Architecture Decisions and Change Scenarios"* cujo objetivo é avaliar a eficácia e usabilidade da abordagem de identificação do Dívida Técnica de Arquitetura (*Architectural Technical Debt - ATD*) baseada em decisões de arquitetura e cenários de mudança, através de um estudo de caso [44].

O segundo resultado refere-se a um artigo publicado em 2017 *ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, com o título *"Formative Evaluation of a Tool for Managing Software Quality"*, cujo objetivo é avaliar a qualidade de uma ferramenta de Gerenciamento de Qualidade de Software, ou seja, avaliar a qualidade do protótipo ProDebt. O objetivo do protótipo ProDebt é fornecer aos profissionais suporte para gerenciar a qualidade do software e a dívida técnica [26].

Tal fato comprova a ausência de estudos sobre dívida técnica vinculada à usabilidade e corrobora com esta pesquisa de estudo de casos múltiplos cujo objetivo é dar foco ao fenômeno da dívida de usabilidade em suas perspectivas de ocorrência, natureza e esforço de resolução.

2.6 Usabilidade e as heurísticas de Nielsen

Após compilar diversas definições do termo "Heurística" podemos definir, por inferência, para o contexto de informática, que se trata de um método de investigação baseado na aproximação progressiva de um dado problema. Segundo Nielsen et al. [5] estes princípios são chamados de "heurística" porque são regras gerais e não são diretrizes específicas de usabilidade.

Os 10 princípios gerais de Jakob Nielsen para design de interação [5], serão utilizados como referência em nosso trabalho para a classificação dos itens de DT de usabilidade, gerando subsídios para auxiliar a resposta da questão de pesquisa RQ2.

Nielsen et al. [5], originalmente desenvolveu os 10 princípios para avaliação heurística em colaboração com Rolf Molich [28], em 1990.

Desde então foram refinadas com base em uma análise fatorial de 249 problemas de usabilidade [30], para derivar um conjunto de heurísticas com o máximo poder explicativo, resultando neste conjunto revisado de heurísticas [27]:

1. Visibilidade do status do sistema

O sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de feedback apropriado dentro de um prazo razoável.

Ex.: Ao ouvir uma playlist, a interface deve exibir: qual o vídeo está sendo ouvido, o próximo da lista e em destaque os vídeos que já foram ouvidos.

2. Correspondência entre o sistema e o mundo real

O sistema deve falar a linguagem dos usuários, com palavras, frases e conceitos familiares ao usuário, em vez de termos orientados pelo sistema. Siga as convenções do mundo real, fazendo as informações aparecerem em uma ordem natural e lógica.

Ex.: O sistema deve destacar em vermelho algo negativo. O usuário naturalmente sabe que algo destacado em vermelho deve ser visto com atenção.

3. Controle do usuário e liberdade

Os usuários geralmente escolhem as funções do sistema por engano e precisarão de uma "saída de emergência" claramente marcada para deixar o estado indesejado sem ter que passar por um diálogo extenso. Suporte desfazer e refazer.

Ex.: Em um aplicativo de mensagem o sistema deve permitir que o usuário exclua uma mensagem mesmo após o envio. O sistema deve permitir que um engano seja reparado de imediato, isto causa um conforto ao usuário e evita tensões desnecessárias.

4. Consistência e padrões

Os usuários não devem se perguntar se palavras, situações ou ações diferentes significam a mesma coisa. Siga as convenções da plataforma .

Ex.: O botão para enviar a resposta deve ser sempre da mesma cor, tamanho e texto, assim como o botão para cancelar, deve ser o mesmo em todo o sistema. O usuário identifica a função do botão em todo sistema facilmente.

5. Prevenção de erros

Ainda melhor do que boas mensagens de erro é um projeto cuidadoso que impede que um problema ocorra em primeiro lugar. Elimine as condições propensas a erros ou verifique-as e apresente aos usuários uma opção de confirmação antes de se comprometerem com a ação.

Ex.: A interface deve sempre evitar que o usuário erre. Ao digitar um texto o sistema deve corrigir erros ortográficos automaticamente, durante a digitação e sugerir opções oriundas do contexto.

6. Reconhecimento ao invés de lembrar

Minimize a carga de memória do usuário, tornando os objetos, ações e opções visíveis. O usuário não deve ter que lembrar informações de uma parte do diálogo para outra. As instruções de uso do sistema devem ser visíveis ou facilmente recuperáveis sempre que apropriado.

Ex.: O sistema deve sempre exibir o caminho que o usuário fez até onde está, o chamado "*breadcrumb*".

7. Flexibilidade e eficiência de uso

Os aceleradores - nunca vistos pelo usuário iniciante - podem acelerar a interação do usuário especialista, de modo que o sistema possa atender a usuários inexperientes e experientes. Permitir que os usuários personalizem ações frequentes.

Ex.: Sempre que possível o sistema deve proporcionar atalhos rápidos, que podem ajudar aos mais experientes ações mais rápidas. O uso de atalhos de teclados, preenchimento automático a partir de dados anteriores são sempre úteis.

8. Design estético e minimalista

Os diálogos não devem conter informações irrelevantes ou raramente necessárias. Cada unidade extra de informação em um diálogo compete com as unidades relevantes de informação e diminui sua visibilidade relativa.

Ex.: O sistema deve ser claro e objetivo na sua proposta, excesso de informações podem confundir, escassez de informações geram insegurança, não deixe o design esteticamente poluído, um exemplo clássico de site eficiente é o Google, simples, direto e prático. Um dos sites mais acessados do planeta.

9. Ajude os usuários a reconhecer, diagnosticar e recuperar erros

As mensagens de erro devem ser expressas em linguagem simples (sem códigos), indicar precisamente o problema e sugerir construtivamente uma solução.

Ex.: Quando o usuário estiver digitando um campo no formulário, e houver inconsistência, o sistema deve exibir de imediato uma mensagem, clara e objetiva,: "Senha inválida - A senha deve ter entre 6 e 8 dígitos, letras e números".

10. Ajuda e documentação

Mesmo que seja melhor se o sistema puder ser usado sem documentação, pode ser necessário fornecer ajuda e documentação. Qualquer informação desse tipo deve ser fácil de pesquisar, focada na tarefa do usuário, listar etapas concretas a serem executadas e não ser muito grande.

Ex.: O sistema deve deixar, próximo ao usuário uma documentação em ações que haja necessidade de uma explicação mais detalhada, como: Código de identificação do cartão de crédito, deve ter uma imagem do verso do cartão destacando o local onde se encontra o número.

Capítulo 3

ESTUDO DE CASO

Neste capítulo o estudo de caso é descrito. Seguindo as diretrizes propostas por Runeson et al.[36], são apresentados as questões de pesquisa a serem respondidas, as empresas e os projetos selecionados para o estudo de múltiplos casos e os procedimentos empregados para a coleta de dados.

3.1 Questões de pesquisa

Motivado pelo fato de que se tem muito pouco conhecimento prático a respeito das dividas técnicas de usabilidade, este trabalho se propõe a responder às seguintes questões de pesquisa:

RQ1: Itens de dívida de usabilidade são comuns nos projetos de desenvolvimento de software pesquisados?

Esta questão tem o propósito de mostrar qual a frequência que itens de dívida de usabilidade são registrados e relatados em cada projeto pesquisado. Ao final será apurada a proporção de itens de dívida de usabilidade em relação a outros tipos de dívidas. Essa proporção estabelece a frequência de itens de DT de usabilidade nos projetos.

RQ2: Que tipos de dívida de usabilidade tendem a ocorrer nos projetos de desenvolvimento de software pesquisados?

Após serem apurados todos os itens de DT de usabilidade, estes serão classificados com base nas Heurísticas de Usabilidade de Nielsen [5]. Esta classificação pode ajudar a determinar as possíveis origens da existência deste tipo de DT nos projetos.

RQ3: Qual o esforço para resolver itens de dívida de usabilidade nos pro-

jetos de desenvolvimento software pesquisados?

Esta questão busca apurar o esforço total para a resolução dos item de DT dos projetos (medido através do esforço homem-hora), ou seja, o esforço a ser investido para o pagamento das dívidas. A partir deste esforço, vamos analisar o esforço total para resolução da DT, em relação ao esforço investido na resolução específica de DT de usabilidade.

3.2 Seleção dos projetos de software

É importante usar várias fontes de dados em um estudo de caso para limitar os efeitos de interpretação de uma única fonte de dados. Se a mesma conclusão pode ser tirada de várias fontes de informação, isto é, triangulação, essa conclusão é mais forte do que uma conclusão baseada em uma única fonte [36], neste cenário conduzimos esse estudo de casos múltiplos.

Foram selecionadas quatro empresas públicas brasileiras, todas com equipes de tecnologia da informação qualificadas envolvendo profissionais graduados, pós-graduados, mestres e doutorandos. Adicionalmente, para cada empresa, os dados obtidos do acesso autorizado às bases de dados corporativas foram validados em reuniões com as equipes dos projetos. No total foram selecionados cinco projetos de desenvolvimento de software.

Em todos os projetos pesquisados haviam sistemas de gestão de tarefas para o registro de demandas (*issue tracking tool*). Alguns sistemas possuíam bases mais ricas em detalhes do que outras, mas todas atendiam plenamente às necessidades de informação básicas para a pesquisa, tendo o número da demanda, autor e data do cadastramento, descrição, situação atual e comentários. Em todas as bases pesquisadas o critério adotado para seleção das demandas eram itens com características de DT, registrados com data de cadastramento anterior a 2017 e que estivessem com situação atual em aberto.

Vale ressaltar que, como são projetos que estão em ambiente de produção e em constante evolução, pode ocorrer que ao longo da pesquisa algum item de DT selecionado, tenha sido pago. Para efeito da pesquisa, foi considerada a situação do item de DT no momento de sua seleção.

A seguir serão feitas breves apresentações das empresas, o ambiente tecnológico e os respectivos projetos envolvidos. Para fins de privacidade os nomes de algumas empresas bem como os projetos selecionados serão tratados por pseudônimos.

3.2.1 Projetos selecionados da Empresa-1

A Empresa-1 é uma empresa pública brasileira, do estado do Mato Grosso, com mais de 60 anos de existência que atua no controle da gestão dos recursos públicos. Tem uma área de TI altamente atuante e capacitada em diversos projetos, algumas áreas tendo certificação ISO-9001.

Nesta empresa foram selecionados dois projetos, que aqui trataremos como Projeto-A e Projeto-B.

Ambos projetos são referentes a sistemas técnicos de controle externo, desenvolvidos como aplicações web em Java e baseados no *framework* MVC, utilizando banco de dados Oracle 11g [7] e servidor de aplicação JBoss EAP [3]. A metodologia de desenvolvimento empregada foi baseada nas melhores práticas para desenvolvimento de software - RUP [2].

Há aproximadamente 29 profissionais de diversas especialidades envolvidos nas equipes dos projetos A e B, destes, 12 profissionais atuam em ambas as equipes: Analista de BI - (*Business Intelligence*) (3), Analista de Infraestrutura (2), Analista de Qualidade (3), Analista de Suporte (3), Desenvolvedor (1).

Vale ressaltar que, não há uma relação de um para um entre profissionais e atividade desenvolvida. Neste contexto, alguns profissionais exercem mais de uma atividade nos projetos, atualmente 7 profissionais atuam em mais de uma atividade.

A equipe técnica do projeto A conta com 20 profissionais envolvidos: Analista de BI - (*Business Intelligence*) (3), Analista de Infraestrutura (2), Analista de Qualidade (4), Analista de Sistemas (5), Analista de Suporte (6), Desenvolvedor (5), Analista de Requisitos (2), Líder do projeto (1).

A equipe técnica do projeto B conta com 22 profissionais envolvidos: Analista de BI - (*Business Intelligence*) (3), Analista de Infraestrutura (3), Analista de Qualidade (6), Analista de Sistemas (5), Analista de Suporte (3), Desenvolvedor (1), Analista de Banco de Dados (2), Líder do projeto (1).

As demandas oriundas dos projetos são registradas em um sistema de gestão de tarefas chamado RedMine [8]. Este sistema permite registrar, controlar e acompanhar todas as demandas encaminhadas ao projeto, mantendo os históricos das evoluções. O primeiro autor teve acesso autorizado a base de dados, com o perfil "somente leitura".

O trabalho de coleta de dados junto a Empresa-1 (Projetos A e B) teve início em

novembro de 2017 e finalizou em outubro de 2018. Somente demandas registradas anteriores a 2017, com características de DT (demandas úteis e necessárias ao sistema ainda não implementada), e com situação em aberto foram selecionadas.

Projeto-A: O objetivo deste projeto é a automatização de todo o processo de construção de relatórios de auditoria: pesquisa e coleta de dados *in loco*, integração de bases, cruzamento de informações, apontamento de indícios de irregularidades, análise, reanálise, conclusão e entrega. O desenvolvimento deste projeto iniciou em 2012 com a primeira entrega em produção em 2014, encontra-se ainda em produção e desenvolvimento, é considerado pela empresa como um sistema de grande porte e alta complexidade. É um projeto que atende mais de 100 auditores, utiliza inúmeras fórmulas de cálculos e estimativas, cruza informações de várias bases de dados, gerando gráficos, tabelas e projeções, tendo como resultado final um relatório de auditoria completo.

Projeto-B: O objetivo deste projeto é a automatização na geração da Representação de Natureza Interna - RNI e consequentemente geração de multas, a fiscalizados que estejam inadimplentes de envio de documentos e informações obrigatórias por lei. Este projeto iniciou seu desenvolvimento em 2012 com a primeira entrega em 2013, encontra-se atualmente em produção e em franco desenvolvimento de novas funcionalidades.

3.2.2 Projeto selecionado da Empresa-2

A Empresa-2 é uma empresa pública brasileira, do estado do Mato Grosso, órgão do poder legislativo, com mais de 180 anos de existência que atua na criação de leis para o estado e fiscalização do poder executivo estadual.

Possui uma área de TI atuante e capacitada, centralizada na Coordenadoria de Informática, composta por quatro áreas: Governança, Atendimento, Infraestrutura e Desenvolvimento de Sistemas.

A equipe de desenvolvimento de sistemas é composta por um gerente, sete desenvolvedores, um designer e um testador.

As solicitações de demandas para os projetos são analisadas, classificadas e inseridas num *backlog* até sua efetiva análise e execução. Atualmente utilizam duas ferramentas de gerenciamento de demandas *open-source*: GLPI [1] e OpenProject [6].

O ambiente tecnológico é heterogêneo, contando com aplicações próprias, de terceiros e legadas. A equipe de desenvolvimento de software é especializada em PHP com o

Symfony *Framework* [10] e banco de dados MySQL, utilizando a metodologia ágil Scrum [9] para desenvolvimentos dos seus projetos.

Nesta empresa o projeto em estudo será referenciado por Projeto-C e os pesquisadores deste trabalho não tiveram acesso direto à sua base de dados. O trabalho de coleta de dados junto a Empresa-1 (Projetos A e B) teve início em setembro de 2018 e finalizou em novembro de 2018.

Projeto-C: O Projeto-C é formado por um conjunto de sistemas Web (Intranet, Web site, RH, entre outros). O principal sistema do projeto é a "Intranet" que concentra a maior quantidade de soluções e é de maior importância para o negócio, haja vista ser um forte canal de comunicação entre o órgão e a sociedade.

O desenvolvimento do projeto iniciou em 2008 e sofre constantes manutenções, agregando novas tecnologias no intuito de torná-lo um canal simples, atrativo e eficiente, neste mesmo ano este projeto entrou em ambiente de produção.

3.2.3 Projeto selecionado da Empresa-3

A Empresa-3 foi o Laboratório de Documentação Ativa e Design Inteligente (ADDLabs) do Instituto de Computação da Universidade Federal Fluminense (UFF) que, desde 1996, desenvolve pesquisa e tecnologia em Inteligência Artificial e Interação Homem-Computador.

O ADDLabs desenvolve projetos, sistemas computacionais e tecnologia de ponta, nas áreas de óleo e gás, de gestão e de web.

A equipe de desenvolvimento do laboratório utiliza as ferramentas Microsoft Project para auxílio no gerenciamento de projetos, Rational Rose para UML, Visual C++ para programação, soluções de Inteligência Artificial e Processamento de imagens em Visual C++ em seu ambiente tecnológico.

A metodologia utilizada é o Scrum [9]. Scrum é um *framework* de trabalho que emprega diversas ferramentas para o desenvolvimento iterativo e incremental utilizado no gerenciamento de projetos diversos e no desenvolvimento de software ágil. O sistema de gerenciador de tarefas é o Trac [11]. O Trac é uma simples ferramenta, *open source* e de interface web para controle de mudanças em projetos de desenvolvimento de software.

A equipe de desenvolvimento deste projeto é composta por um gerente, quatro analistas, dois programadores (estagiários, um especificamente para testes) e mais um estagiário de engenharia. Os estagiários de engenharia além de testar os sistemas também executam

tarefas específicas de engenharia.

Nesta empresa o projeto em estudo será referenciado por Projeto-D.

Projeto-D: Trata-se do desenvolvimento de um protótipo, contratado por uma empresa petrolífera nacional, para apoio à avaliação de consistência de modelos a partir das informações distribuídas nos bancos de dados: fluxogramas de engenharia, projeto de instrumentação e modelos 3D. O desenvolvimento do Projeto-D iniciou em Dezembro de 2012 e a versão em produção foi entregue para a contratante em Junho de 2014.

Foi concedido aos pesquisadores, acesso de leitura à base de dados do sistema de gestão de tarefas. O período de coleta de dados, classificação, validação e a estimativa de esforço para resolução de itens de DT, junto a equipe envolvida no projeto, durou de 26 a 30/11/2018.

3.2.4 Projeto selecionado da Empresa-4

A Empresa-4 refere-se à Secretaria de Tecnologia da Informação e da Comunicação Aplicadas a Educação (STI) da Universidade Federal do Mato Grosso (UFMT).

A Coordenação de Engenharia de Software (CES), é a responsável pelo desenvolvimento de novos sistemas e manutenção dos existentes. A equipe conta um gerente, 12 analistas e um técnico (área de desenvolvimento), oito analistas de rede e mais quatro analistas responsáveis pelos servidores. Esta equipe é responsável por mais de 50 projetos (novos e legados) que rodam em toda a Universidade.

O sistema de gestão de *bugs* e registro de tarefas é o RedMine [8]. Também é utilizado o Sistema Eletrônico de Informações (SEI), que é o sistema de processos administrativos da UFMT. Todas as demandas são iniciadas através de um processo registrado nesse sistema.

O ambiente tecnológico para desenvolvimento de sistemas é formado pelas tecnologias C# e o *framework* .NET, utilizando o banco de dados SQL Server 2008 (sistema gerenciador de Banco de dados relacional da Microsoft) e servidor de aplicação IIS [4].

A metodologia utilizada para desenvolvimento de sistemas foi desenvolvida pela própria empresa denominada "Metodologia para Desenvolvimento de Sistemas - MDS" tendo como referência as melhores práticas para desenvolvimento de software - RUP [2].

O projeto desta empresa que foi selecionado para fazer parte deste estudo de múltiplos casos é aqui denominado como Projeto-E.

Projeto-E: Este projeto é formado por demandas de um conjunto de sistemas acadêmicos Web (sistemas acadêmicos de extensão, de monitoria, de pós-graduação, sistema de gestão de contratos e convênios, dentre outros) agrupadas no sistema de gestão de tarefas RedMine [8]. O desenvolvimento dos projetos acadêmicos iniciaram em 2001, as primeiras versões entraram em produção no ano seguinte, desde então vem sendo muito utilizado e sofrendo manutenções evolutivas, corretivas e novas funcionalidades, adequando-se às necessidades.

O período de coleta de dados junto a equipe do projeto foi de outubro a dezembro de 2018 e não foi concedido aos pesquisadores acesso à base de dados. Todas as informações foram repassadas pelo líder do projeto.

3.3 Procedimentos para coleta de dados

O escopo deste trabalho se limita a seleção de cinco projetos reais de desenvolvimento de software, em produção, a partir de quatro empresas, conforme apresentado na Seção 3.2.

Todos os projetos seguiram o mesmo roteiro de procedimentos para coleta de dados, com objetivos bem definidos:

- **Capacitação da equipe técnica:** O objetivo desta fase é capacitar as equipes envolvidas nos projetos, quanto aos conceitos de dívida técnica e o entendimento da taxonomia de tipos de DT. Ao final da capacitação, os profissionais, das equipes técnicas, devem estar habilitados a reconhecer um item de DT e classificá-los por tipo, utilizando taxonomia adotada de Rios et al. [33]. As capacitações foram ministradas pelo primeiro autor, no local de trabalho das empresas participantes.
- **Seleção preliminar dos itens de Dívida Técnica** O objetivo desta fase é selecionar, preliminarmente, itens com características de DT e classificá-los por tipo, conforme taxonomia adotada. Ao final desta fase deve ser gerado uma lista com os itens de DT selecionados e classificados preliminarmente.
- **Avaliação e validação dos itens de Dívida Técnica** O objetivo desta fase é que a equipe técnica se reúna analise e valide os itens selecionados, previamente, pelo primeiro autor ou pelo gerente de projeto. Ao final desta fase a lista de DT deve ter sido validada pela equipe técnica.
- **Fechamento da lista de itens de Dívida Técnica** O objetivo desta fase é que o líder do projeto e o primeiro autor, em reunião, analisem e aprovem a lista validada

pela equipe técnica na etapa anterior. O propósito desta fase é dirimir possíveis dúvidas e fazer o fechamento final da lista de itens de DT.

- **Cálculo do esforço para pagamento da Dívida Técnica** Nesta etapa o gerente do projeto, utiliza de sua expertise para, estimar o esforço para pagamento dos item de DT, utilizando o indexador "homem-hora". Ao final desta etapa todos os itens selecionados devem ter seus esforços estimados.

A seguir são apresentados os detalhamentos dos procedimentos para coleta de dados.

3.3.1 Fase-1 - Capacitação da equipe técnica

Foram selecionados, pelas empresas, servidores para acompanhar as pesquisas. O termo "Equipe Técnica" faz referência a esses profissionais de TI disponibilizados e envolvidos neste estudo.

De uma forma geral os profissionais assimilaram muito bem os conceitos e a taxonomia, haja vista serem profissionais capacitados, apesar de alguns confessarem serem pouco familiarizados com os conceitos de dívida técnica.

O ponto a se destacar foi a criação de uma nova forma de comunicação entre os membros das equipes técnicas utilizando a metáfora da DT.

O único projeto que não houve capacitação foi o Projeto-D, por se tratar de um ambiente acadêmico onde os profissionais envolvidos já tinham ampla vivência com os conceitos de DT e não sentiram necessidade de capacitação.

3.3.2 Fase-2 - Seleção preliminar dos itens de Dívida Técnica

O filtro padrão adotado para seleção de itens de dívida técnica, em todos os projetos, foram os seguintes:

- itens cadastrados, em seus respectivos repositórios, com data anterior a 2017 e
- itens com status "em aberto" no momento da seleção, (encontram-se pendentes de resolução) e
- itens que se enquadrem nos conceitos de dívida técnica.

O primeiro autor teve acesso, com perfil somente leitura, às bases de dados dos projetos A, B e D. Foram selecionados e classificados, preliminarmente, nos projetos A, B e D respectivamente 120, 24 e 64 itens.

Para dar suporte à pesquisa foi desenvolvida uma planilha modelo para coleta de dados (Tabela 3.1).

Tabela 3.1: Modelo para lançamento e classificação de itens de Dívida Técnica.

Relação de demandas classificadas por Tipo de Dívida Técnica				
#	Projeto	Tipo de DT	Título	Descrição

Nos projetos C e E, a seleção, preliminar, ficou a cargo dos seus respectivos gerentes de projetos, pois o primeiro autor não teve acesso autorizado à base de dados.

A classificação (feita após a seleção preliminar) em todos os projetos, obedecendo a taxonomia adotada, foi executada pelo primeiro autor.

Neste contexto, foram selecionados e classificados preliminarmente nos projetos C e D, 27 e 20 itens respectivamente.

Nesta fase foram selecionados e classificados, ao longo dos 5 projetos, um total preliminar de 255 itens que se enquadravam nos conceitos de DT.

3.3.3 Fase-3 - Avaliação e validação dos itens de Dívida Técnica

Nesta fase a equipe técnica analisa a planilha com os itens de DT selecionados e classificados. Fazem uma nova rodada de avaliação dos itens selecionados, fazendo os ajustes que julgarem necessários. A equipe técnica tem a prerrogativa de incluir, alterar ou excluir itens, e até reclassificá-los, segundo seu entendimento, sempre tendo como referência a taxonomia adotada.

No Projeto-A, essa fase foi realizada de forma bastante detalhada, o que não se repetiu nos demais projetos. Isso se justifica porque foi o primeiro projeto a ser pesquisado, com a equipe muito dedicada. Neste projeto foi estruturada a pesquisa de tal forma que o formato construído aqui foi replicado na sequência da pesquisa, nos demais projetos.

A equipe técnica dividiu essa fase em duas etapas: a primeira etapa foi analisar a classificação de tipos de DT com o claro objetivo de consolidar o conhecimento na taxonomia adotada. Neste contexto, 13 itens foram reclassificados e foram acrescentados a lista mais 7 novos itens, a lista então passou de 120 para 127 itens.

Na segunda etapa, o foco foi analisar a real situação do item dentro do contexto do projeto. A equipe técnica entendeu que havia uma quantidade excessiva de itens classificados, fora do contexto e que não deveriam estar ali. Foi estabelecido o seguinte critério para exclusão de itens da lista:

- itens já finalizados em outras demandas;
- itens suspensos por decisão superior (diretoria não autorizou a implementação destes itens)
- itens fora do escopo do projeto (obsoletos).

Utilizando este critério, a equipe técnica do Projeto-A eliminou 79,53% da lista inicial (101 itens), restando 26 itens (20,47%) que foram analisados e comprovadamente representam itens de DT.

No Projeto-B, a lista contendo 24 itens de DT, foi encaminhada à equipe técnica para validação. Após a análises e discussões os itens foram validados sem ressalvas. Ou seja, a equipe técnica validou os 24 itens da lista original.

No Projeto-C, o líder do projeto questionou três itens, 2 classificados como sendo de processo e 1 classificado como de código. Após a reunião os dois itens classificados como de processo foram mantidos e o item classificado como de código foi alterado para usabilidade, os demais foram validados. Assim, ao final desta etapa a lista de itens de DT manteve os 27 itens que seguiram para próxima etapa.

No Projeto-D, foram excluídos 19 itens, incluídos 3 novos itens e 11 itens foram reclassificados, gerando uma nova lista com 48 itens de DT. Segundo o líder do projeto 19 itens foram excluídos por já terem sido pagos em outras oportunidades por demandas similares e outros por estarem obsoletos para o projeto. Foram incluídos 3 novos itens (1 do tipo código e 2 do tipo requisitos), outros 11 itens tiveram a classificação alterada.

No Projeto-E, os itens foram selecionados pelo líder do projeto e classificados pelo primeiro autor, que os enviou para a equipe técnica validar. Após a reunião de validação a equipe técnica manteve os 20 itens da lista original sem alterações.

Foram levantados preliminarmente por esta pesquisa um total de 255 itens de dívida técnica, a lista completa dos itens bem como sua descrição e classificação encontra-se no Apêndice A.1.

Ao final desta etapa de validação o total de itens selecionados passou de 255 para 145 itens, uma redução de 43,13%.

Embora o número de itens tenha reduzido, optou-se por ter itens validados e representativos, de acordo com as equipes envolvidas nos projetos.

Vale ressaltar que nessa etapa houveram várias reuniões de alinhamento, até chegar a lista consolidada dos itens.

3.3.4 Fase-4 - Fechamento da lista de itens de Dívida Técnica

Nesta fase, reuniões foram realizadas individualmente após agendamento prévio, com cada líder de projeto. Algumas reuniões foram somente entre o líder do projeto e o primeiro autor e outras tiveram a participação da equipe técnica. Todos os itens dos projetos obtidos na fase 2 e ajustados na fase 3 foram validados como pertinentes e corretamente classificados.

3.3.5 Fase-5 - Cálculo do esforço para resolução da Dívida Técnica

Durante as reuniões para alinhamento das estimativas de esforços, logo no primeiro projeto pesquisado (Projeto-A), ocorreu uma situação interessante: abriu-se a discussão da melhor forma de estimar os esforços para pagamento da DT dos itens.

Durante as discussões, surgiu a ideia de dividir a execução da tarefa em atividades menores e valorar o esforço na execução de cada atividade em homem-hora. Como os gerentes dos projetos são muito experientes esta ideia tornou-se real. Algo similar já era rotina para a equipe técnica em outros projetos de estimativa de esforço.

Para esta estimativa é necessário que o Gerente conheça o valor HH (homem/hora) de cada cargo e a quantidade de horas que serão necessárias para cada etapa. Para estimativas iniciais, estima-se uma margem de erro de 25%. A medida que os profissionais são alocados ao projeto, o detalhamento das atividades será maior e o grau de assertividade do orçamento também. Vale lembrar que a experiência deste gerente em projetos anteriores fará muita diferença [13].

Foram adotadas seis atividades, utilizadas na estimativa de custo (homem-hora). Estas atividades foram inspiradas na metodologia RUP [2], por dois motivos: 1) Estas atividades estão presentes na maioria das metodologias tradicionais de desenvolvimento de software. 2) A maioria dos projetos pesquisados utilizam RUP, e mesmo os projetos que não utilizam esta metodologia, tem amplo conhecimento dos objetivos e ações vinculadas em cada uma das seguintes atividades:

- Análise e Projeto
- Arquitetura e Banco de Dados
- Desenvolvimento
- Testes
- Implantação e treinamento
- Gerência de projeto

Foi desenvolvido pelo primeiro autor uma planilha eletrônica para auxiliar o gerente de projeto. O gerente estimou as horas por atividade e os cálculos dos totais eram feitos automaticamente pela planilha. Vale ressaltar que cada item analisado tem suas particularidades e que o líder estimou o esforço homem-hora somente as atividades que julgou necessárias para a resolução do item.

Esta etapa de estimativas de esforços para resolução dos itens de DT, foi aplicada somente nos projetos A, C e D. Nos projetos B e E, a equipe não realizou esta etapa, por considerar a estimativa muito complexa.

Capítulo 4

ANÁLISE DOS DADOS COLETADOS E RESULTADOS OBTIDOS

Neste capítulo serão apresentados as análises dos dados da pesquisa (pela perspectiva da frequência e origem e da perspectiva do esforço para pagamento das DT de usabilidade), os resultados obtidos e as respostas às questões de pesquisa.

4.1 Introdução

Há duas perspectivas fundamentais para as análises: (i) a perspectiva da existência de itens de DT de usabilidade nas amostras analisadas, num contexto de frequência e origem, gerando subsídios para responder as questões de pesquisa RQ1 e RQ2; e (ii) a perspectiva do impacto, relativo ao esforço, para pagamento das DT de usabilidade em relação a outros tipos de dívidas, dando subsídios para responder a questão de pesquisa RQ3. Neste contexto, os resultados das análises serão apresentados através das respostas às questões da pesquisa.

Para o melhor entendimento das análises seguintes, na Tabela 4.1 são apresentados dois subtotais. O Grupo-1 que representa aqueles tipos de dívida com vinculação à código-fonte, considerados os mais pesquisados, com maior número de ferramentas de pesquisa, mais comuns e com mais artigos publicados. O Grupo-2, por sua vez, representa os tipos de dívida sem vinculação com código-fonte, que são mais específicos e que ainda carecem de estudos, segundo Rios et al. [33].

Tabela 4.1: Análise dos itens de Dívida Técnica

Tipos de DT	Apuração da quantidade de itens de DT										Acumulado	
	Projeto-A		Projeto-B		Projeto C		Projeto-D		Projeto-E			
	Itens	%	Itens	%	Itens	%	Itens	%	Itens	%	Itens	%
Arquitetura	2	7,7%	1	4,2%			4	8,3%	1	5,0%	8	5,52%
Código	11	42,3%	5	20,8%	4	14,8%	6	12,5%	2	10,0%	28	19,31%
Defeito	2	7,7%	2	8,3%	5	18,5%	11	22,9%	3	15,0%	23	15,86%
Design							6	12,5%	2	10,0%	8	5,52%
Subtotal Grupo-1											67	46,21%
Documentação	3	11,5%					4	8,3%	1	5,0%	8	5,52%
Infraestrutura					11	40,7%					11	7,59%
Processo					2	7,4%			1	5,0%	3	2,07%
Requisitos	4	15,4%	10	41,7%	2	7,4%	11	22,9%	7	35,0%	34	23,45%
Serviço			1	4,2%							1	0,69%
Teste							1	2,1%			1	0,69%
Usabilidade	4	15,4%	5	20,8%	3	11,1%	5	10,4%	3	15,0%	20	13,79%
Subtotal Grupo-2											78	53,79%
Total	26	100,0%	24	100,0%	27	100,0%	48	100,0%	20	100,0%	145	100,0%

4.2 Análise dos tipos de Dívida Técnica

Foram selecionados 145 itens de DT em 5 projetos pesquisados. A lista completa dos itens de DT pode ser visualizada no Apêndice B.1. e a Tabela 4.1 mostra a quantidade de itens selecionados e respectivos percentuais, por tipo de DT, conforme a taxonomia adotada.

Os itens classificados, por tipo de DT, ficaram assim distribuídos: Arquitetura (8), Código (28), Defeito (23), Design (8), Documentação (8), Infraestrutura (11), Processo (3), Requisitos (34), Serviço (1), Teste (1) e Usabilidade (20), (ver Figura 4.1).

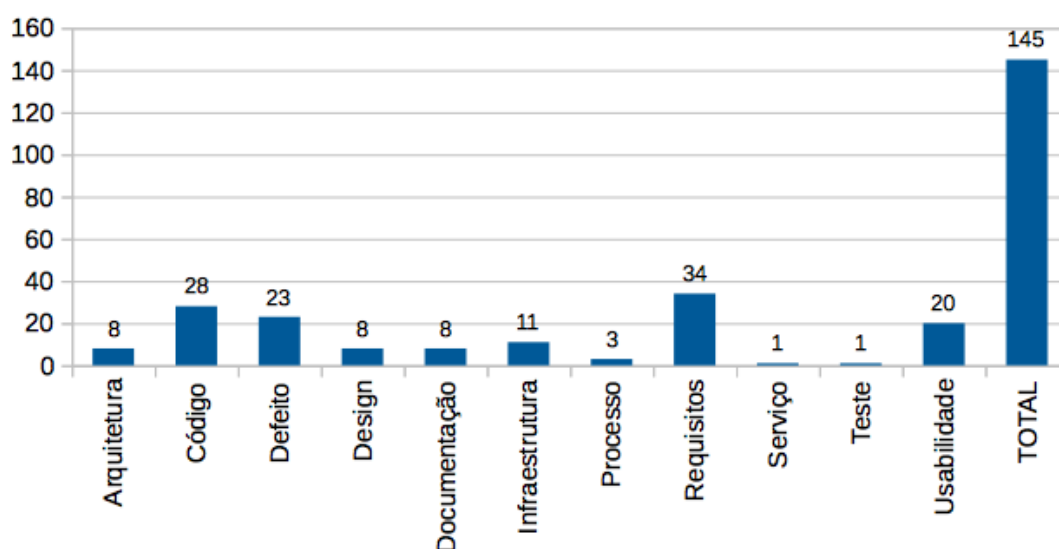


Figura 4.1: Quantidade de itens apurados por tipo de DT

Numa análise por projetos, a quantidade de itens de DT ficou assim definida: Projeto-

A (26), Projeto-B (24), Projeto-C (27), Projeto-D (48) e Projeto-E (20), (ver Figura 4.2)..

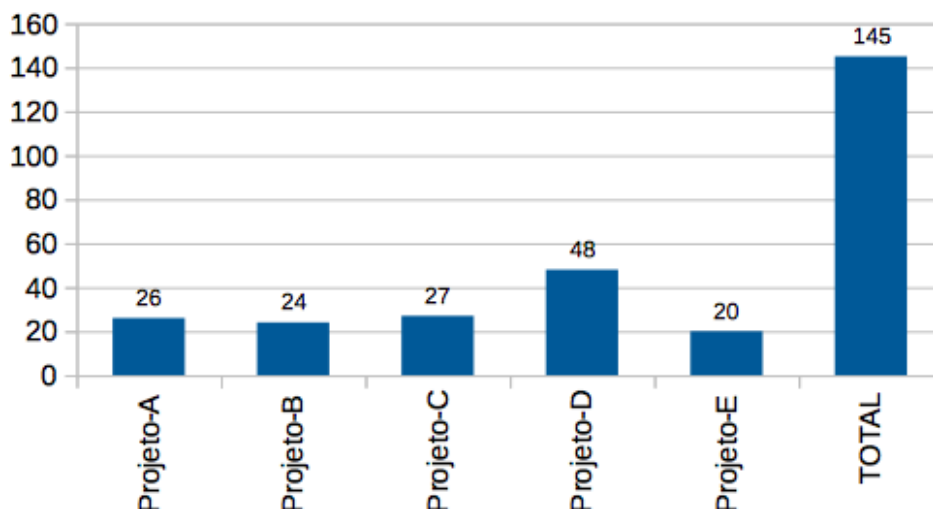


Figura 4.2: Quantidade de itens de DT apurados por projeto

Os 4 tipos de dívida do Grupo-1 representam 46,21%, enquanto os outros 7 tipos identificados do Grupo-2, representam 53,79% dos itens.

Somente 4 tipos de dívida estavam presentes e com percentuais significativos em todas as bases de dados pesquisadas: Requisitos (23,45%), Código (19,31%), Defeito (15,86%) e Usabilidade (13,79%).

Tal fato demonstra que, apesar da carência de estudos sobre itens de DT de usabilidade, estes são frequentes e representativos em projetos de desenvolvimento de software. O que nos fornece subsídios para responder à primeira questão de pesquisa RQ1.

4.2.1 RQ1: Itens de dívida de usabilidade são comuns nos projetos de desenvolvimento de software pesquisados?

Para responder esta questão foi utilizada a quantidade de itens de DT de usabilidade apurados nos cinco projetos analisados, em comparação aos demais tipos de DT (Tabela 4.1).

No Projeto-A, 15,4% dos itens de dívida técnica eram de usabilidade, a mesma quantidade de itens de requisitos, ambos em 2º lugar, atrás somente de itens de DT de Código com 42,3%.

No Projeto-B, os itens de DT de usabilidade totalizaram 20,8%, igualando aos itens de código, ambos em 2º lugar, atrás somente dos itens de requisitos com 41,7%.

No Projeto-C, os itens de DT de usabilidade foram de 11,1% do total, ficando em 4º lugar, atrás dos itens de infraestrutura com 40,7%, defeito com 18,5% e código com 14,8%.

No Projeto-D, itens de DT de usabilidade representaram 10,4% do total, ficando na 5ª posição, atrás dos itens de defeito e requisitos com 22,9% e código e design com 12,5%.

No Projeto-E, 15% dos itens eram de DT de usabilidade. A mesma quantidade de itens DT de defeito, ficando em 2º lugar e perdendo somente para itens de requisitos com 35%.

No acumulado de todos os projetos, os itens de DT de usabilidade representaram 13,79% do total, ficando em 4º lugar, atrás de itens de DT de requisitos (23,45%), código (19,31%) e defeito (15,86%).

Se analisarmos somente os itens do Grupo-2, a frequência dos itens de DT de usabilidade alterna entre o primeiro e o segundo lugar em todos os projetos pesquisados. No acumulado deste grupo fica na segunda posição com 13,79%, somente atrás de itens de Requisitos 23,45%.

Uma outra constatação é que o tipo de DT de usabilidade sozinho, tem frequência (13,79%) maior que a média do Grupo-1 (11,55%).

As análises confirmam a representatividade de itens de DT de usabilidade no contexto geral da pesquisa. Foi possível ainda observar que dívida técnica de usabilidade foi um dos quatro tipos de dívida com ocorrência em todos os projetos.

Portanto, com base nos cinco projetos analisados, é possível afirmar que "Sim", itens de dívida técnica de usabilidade são frequentes e representativos em projetos de desenvolvimento de software.

4.3 Análise das dívidas técnicas de usabilidade pela ótica das heurísticas de Nielsen

Esta análise busca entender qual seria o impacto dos itens de DT de usabilidade em projetos de software. Para isso, ao final da fase de coleta de dados, os itens de usabilidade foram classificados segundo as Heurísticas de Jacob Nielsen e Rolf Molich [28] [5]. Tais heurísticas representam os 10 princípios gerais para design de interação, e foram utilizados em nosso trabalho para analisar os itens de DT de usabilidade. São elas:

1. Visibilidade do status do sistema
2. Correspondência entre o sistema e o mundo real
3. Liberdade de controle fácil para o usuário
4. Consistência e padrões
5. Prevenção de erros
6. Reconhecimento em vez de memorização
7. Flexibilidade e eficiência de uso
8. Estética e design minimalista
9. Ajuda aos usuários para reconhecer, diagnosticar e recuperar erros
10. Ajuda e documentação

Ao fazer a avaliação dos itens, o especialista deve levar em conta os princípios gerais de um bom design de interfaces, visando identificar situações onde as heurísticas são violadas, baseadas na sua expertise [27].

Ao todo, foram identificados 20 itens do tipo dívida técnica de usabilidade, representando 13,79% do total de itens apurados nos 5 projetos. A identificação das Heurísticas violadas nos itens de usabilidade, foi realizada pelo primeiro autor, e apresentada em reunião aos gerentes de cada um dos 5 projetos para validação. O resultado das heurísticas identificadas pode ser visto na Tabela 4.2.

Sabe-se que um dado problema de interação (aqui representado por um item de DT de usabilidade) pode estar violando mais de uma heurística. No nosso estudo de caso observamos que três itens violaram mais de uma heurística conforme descrito abaixo e destacado na Tabela 4.2.

- Projeto-C - item nº 190 (violou as heurísticas 8 e 9),
- Projeto-D - item nº 970 (violou as heurísticas 3 e 5)
- Projeto-E - item nº 13-SI (violou as heurísticas 1 e 6).

De uma forma geral os itens de DT de usabilidade encontrados apontam problemas de interação relacionados principalmente com Flexibilidade e eficiência de uso, com 7 itens

Tabela 4.2: Heurísticas violadas pelos itens de dívida técnica de usabilidade.

#	HEURÍSTICA NIELSEN	Ticket	Projeto	Summary
1	1-Visibilidade Status Sistema	13-SI	Projeto-E	Mostrar quando a ação não foi realizada sem clicar em detalhes
2		1261	Projeto-D	Exportar para excel - Cursor de espera
3	2-Correspondência entre o sistema e o mundo real	6635	Projeto-B	Correção na descrição das Propostas de RNI de 2015
4	3-Liberdade de controle fácil para o usuário	970	Projeto-D	Restaurando a ordenação padrão do demonstrativo de inconsistências
5		11-SI	Projeto-E	Listar projetos por unidade/ Responsável por unidade visualizar todas as propostas
6	5-Prevenção de Erro	1293	Projeto-B	Possibilitar salvamento dos textos de forma automática
7		860	Projeto-D	Ajustar sensibilidade do clique de acionamento do filtro [Tela Parâmetros Fusíveis]
8		970	Projeto-D	Restaurando a ordenação padrão do demonstrativo de inconsistências
9		965	Projeto-D	Desabilitar botão SALVAR após o salvamento do projeto
10	6-Reconhecimento ao invés de lembrar	891	Projeto-B	Filtro de propostas
11		821	Projeto-D	TAGs ligeiramente diferentes (Adicionar identificador nos tags)
12		13-SI	Projeto-E	Mostrar quando a ação não foi realizada sem clicar em detalhes
13	7-Flexibilidade e eficiência de uso	639	Projeto-B	Relatório gerencial - RNI
14		2666	Projeto-B	Geração automática de cadastro de inadimplência
15		7035	Projeto-A	Permitir Ordenar de Itens de Verificação em tópicos diferente
16		4510	Projeto-A	Link para trazer PDF do APLIC
17		2956	Projeto-A	Gráficos (A partir de quadros permitir a geração de gráficos (pizza, colunas, linhas, barras)).
18		43	Projeto-C	Acesso a determinados sistemas quando o usuário for Gestor de Unidade
19		12-SI	Projeto-E	Cadastrar função a partir da tela de membro de equipe
20	8-Design estético e minimalista	29	Projeto-C	Trocar o Template dos Relatórios do SAHRA
21		190	Projeto-C	Páginas de Exceção mais Elegantes
22	9-Ajude o usuário a reconhecer, diagnosticar e recuperar erros	2349	Projeto-A	Correção de mensagem de Item de Verificação em Processamento de Relatório Técnico
23		190	Projeto-C	Páginas de Exceção mais Elegantes

(30,4%), seguida de Prevenção de erro com 4 itens (17,4%) e Reconhecimento ao invés de lembrar com 3 itens (13%) (Tabela 4.3).

A seguir as descrições das heurísticas citadas para um melhor entendimento desta análise:

- **7-Flexibilidade e eficiência de uso:** As ações de interface devem ter diferentes formas de ser acionadas. Em nosso estudo de caso os itens classificados por esta heurísticas foram solicitações de usuários experientes requerendo uma ação ou uma alternativa mais eficaz para uma funcionalidade dos sistemas.

Tabela 4.3: Totalização os itens de DT de usabilidade - Heurísticas de Nielsen.

HEURÍSTICAS NIELSEN	Qtd	%
1-Visibilidade do status do sistema	2	8,7%
2-Correspondência entre o sistema e o mundo real	1	4,3%
3-Liberdade de controle fácil para o usuário	2	8,7%
5-Prevenção de erros	4	17,4%
6-Reconhecimento em vez de memorização	3	13,0%
7-Flexibilidade e eficiência de uso	7	30,4%
8-Estética e design minimalista	2	8,7%
9-Ajudar aos usuários para reconhecer, diagnosticar e recuperar erros	2	8,7%
Totalização	23	100,0%

- **5-Prevenção de erro:** O sistema deve evitar que enganos e erros ocorram, sempre que possível. As demandas registradas, neste estudo de caso, visam levantar uma situação no sistema que possa levar a algum erro, ou esteja causando confusão de entendimento por necessidade de ajustes.
- **6-Reconhecimento em vez de memorização:** A interface deve apresentar claramente os objetos, ações e opções, pois o usuário não deve precisar "decorar" formas de acionamento do sistema. As demandas, no nosso estudo de caso, selecionadas nesta categoria, tinham em comum uma dificuldade do usuário no entendimento de uma ação a ser executada nos sistemas através do uso de sua interface.

Esta análise subsidia a resposta à nossa segunda questão de pesquisa, que se encontra descrita a seguir.

4.3.1 RQ2: Que tipos de dívida de usabilidade tendem a ocorrer nos projetos de desenvolvimento de software pesquisados?

Os 20 itens de dívida técnica representam a violação de 8 das 10 heurísticas de usabilidade (Tabela 4.2). Percebe-se que não há concentração em uma determinada heurística, porém 3 heurísticas concentram aproximadamente 60,8% dos itens selecionados:

- **7-Flexibilidade e eficiência de uso:** Foi apontada em 30,4% dos itens, esta heurística diz que os aceleradores - nunca vistos pelo usuário iniciante - podem acelerar a interação do usuário especialista, de modo que o sistema possa atender a usuários inexperientes e experientes. Permitir que os usuários personalizem ações frequentes [5].

Em nosso trabalho os itens classificados por esta heurísticas foram solicitações de usuários experientes requerendo uma ação ou uma alternativa mais eficaz para uma funcionalidade dos sistemas. Nesta seleção de itens fica claro a intensão dos usuários em tornar suas atividades rotineiras mais eficientes, quando por exemplo, solicitam acesso mais rápidos a determinados sistemas, *links*, geração automática de cadastros, etc. (ver Tabela 4.2).

- **5-Prevenção de erro:** Apontada em 17,4% dos itens. Esta heurística diz que o sistema deve evitar que enganos e erros ocorram, sempre que possível. Ainda melhor do que boas mensagens de erro é um projeto cuidadoso que impeça que um problema ocorra. Elimine as condições propensas a erros ou verifique-as e apresente aos usuários um opção de confirmação antes de se comprometerem com a ação [5].

As demandas registradas, nesta pesquisa, visam levantar uma situação no sistema que possa levar a algum erro, ou esteja causando confusão de entendimento por necessidade de ajustes. Neste contexto, na seleção e classificação de itens fica evidente a preocupação em evitar erros, quando por exemplo, solicitam o salvamento de texto de forma automática, desabilitar o botão <Salvar> após o salvamento do projeto, entre outras, (ver Tabela 4.2).

- **6-Reconhecimento em vez de memorização:** Apontada em 13% dos itens. Esta heurística diz que a interface deve apresentar claramente os objetos, ações e opções, pois o usuário não deve precisar "decorar" formas de acionamento do sistema. Minimize a carga de memória do usuário, tornando os objetos, ações e opções visíveis. O usuário não deve ter que lembrar informações de uma parte do diálogo para outra. As instruções de uso do sistema devem ser visíveis ou facilmente recuperáveis sempre que apropriado [5].

As demandas selecionadas nesta categoria tinham em comum uma dificuldade do usuário no entendimento de uma das ações de um dos sistemas. Neste sentido, na seleção e classificação de itens fica latente esta preocupação quando são registradas demandas como por exemplo: adicionar identificador nas *Tag's*, sinalizar quando a ação não foi realizada sem ter que clicar em detalhes, etc. (ver Tabela 4.2).

Como as Heurísticas de Nielsen são abrangentes e cada projeto tem seu contexto único, fica difícil definir, a partir das amostras colhidas nesta pesquisa, quais os tipos de dívida técnica de usabilidade ocorrem em projetos de desenvolvimento de software de forma genérica.

No contexto deste trabalho três heurísticas se destacaram dentre as demais, conforme demonstrado.

Porém, pode-se inferir, a partir dos dados apresentados, que "Flexibilidade e eficiência de uso" tem uma tendência a ocorrer, devido às características do tipo DT de usabilidade, que em sua essência, busca tornar a experiência do usuário mais agradável. Neste contexto, usuários experientes, de uma forma geral, utilizando os sistemas por um tempo, tornam-se aptos a solicitar mais recursos ao sistema de modo a torná-lo mais eficiente.

4.4 Análise do esforço (homem-hora) por tipo de Dívida Técnica

Uma outra perspectiva de análise realizada neste estudo é o esforço para pagamento da DT. Nesta pesquisa foi adotada a métrica homem-hora. Neste contexto, somente os projetos A, C e D forneceram dados, totalizando 101 itens e 6.254 homem-hora apuradas (Tabela 4.4). A lista completa da apuração da estimativa de esforço dos itens de DT pode ser visualizada no Apêndice C.1.

Tabela 4.4: Análise do esforço homem-hora.

	Tipos DT	Estimativa de esforço homem-hora									Acumulado		
		Projeto-A			Projeto-C			Projeto-D			Qt Itens	H/H	%
		Qt Itens	H/H	%	Qt Itens	H/H	%	Qt Itens	H/H	%			
Grupo-1	Arquitetura	2	1.680	58,80%				4	108	17,70%	6	1.788	28,59%
	Código	11	198	6,93%	4	382	13,71%	6	74	12,13%	21	654	10,46%
	Defeito	2	32	1,12%	5	373	13,38%	11	175	28,69%	18	580	9,27%
	Design							6	27	4,43%	6	27	0,43%
	Subtotal Group-1										51	3.049	48,75%
Grupo-2	Documentação	3	292	10,22%				4	20	3,28%	7	312	4,99%
	Infraestrutura				11	885	31,75%				11	885	14,15%
	Processo				2	884	31,72%				2	884	14,13%
	Requisitos	4	464	16,24%	2	90	3,23%	11	170	27,87%	17	724	11,58%
	Teste							1	5	0,82%	1	5	0,08%
	Usabilidade	4	191	6,69%	3	173	6,21%	5	31	5,08%	12	395	6,32%
	Subtotal Group-2										50	3.205	51,25%
Total		26	2.857	100%	27	2.787	100%	48	610	100%	101	6.254	100%

Percebe-se uma tendência ao equilíbrio de forças entre os grupos 1 e 2, tanto na quantidade de itens analisados (67 no Grupo 1 e 78 no Grupo 2), quanto na quantidade de homem-hora necessária para resolver os itens (3049 homem-hora no Grupo 1 e 3205 homem-hora no Grupo 2).

Neste cenário o esforço médio para pagamento da DT de um item do Grupo 1 é de 45,5 homem-hora, enquanto o para pagamento de um item do Grupo 2 é bem próximo, 41,1 homem-hora.

Tabela 4.5: Relação do esforço por tipo de Dívida de Técnica.

Relação entre Custo hora/homem x Qtd itens					
Custo DT em Hora/Homem			Qtd itens de DT		
#	%	Tipo DT	#	%	Tipo DT
1º	28,59%	Arquitetura	1º	23,45%	Requisitos
2º	14,15%	Infraestrutura	2º	19,31%	Código
3º	14,13%	Processo	3º	15,86%	Defeito
4º	11,58%	Requisitos	4º	13,79%	Usabilidade
5º	10,46%	Código	5º	7,59%	Infraestrutura
6º	9,27%	Defeito	6º	5,52%	Arquitetura
7º	6,32%	Usabilidade	7º		Design
8º	4,99%	Documentação	8º		Documentação
9º	0,43%	Design	9º	2,07%	Processo
10º	0,08%	Teste	10º	0,69%	Teste
			11º		Serviço

Analisando somente itens de DT de usabilidade (Tabela 4.5), percebe-se uma clara dicotomia.

Numa primeira análise focada na quantidade de itens encontrados, os itens de DT de usabilidade aparecem sempre bem posicionados alcançando ao final a 4º posição geral com 13,79% do total de itens, apenas 1,7 vezes menor que o item com maior quantidade apurada (Requisitos - 23,45%).

Em outra análise focada no esforço para pagamento da DT de usabilidade, a DT de usabilidade apresenta um esforço mais baixo (ocupa somente a 7º posição geral) com apenas 6,32% do esforço total apurado. Quase 5 vezes menor que o item de DT de maior esforço (Arquitetura - 28,59%).

De posse de tais dados podemos observar que, apesar dos itens de DT de usabilidade aparecerem com frequência em todos os projetos analisados neste estudo, o esforço para resolvê-los é baixo em relação a outros tipos de itens de DT.

Pela ótica do gerenciamento de DT, neste contexto, itens de DT de usabilidade são representativos e tem um baixo esforço para pagamento. Esta análise fornece os subsídios para responder a nossa terceira e última questão de pesquisa.

4.4.1 RQ3: Qual o esforço para resolver itens de dívida de usabilidade nos projetos de desenvolvimento software pesquisados?

Para responder esta questão foram analisados os 3 maiores projetos, perfazendo 101 dos 145 itens pesquisados, aproximadamente 70% dos itens. Então foi pedido aos líderes dos projetos que estimassem o esforço para pagamento de cada item, em homem-hora, conforme descrito na Seção 4.4.

No Projeto-A, dos 6 tipos de DT identificados, o esforço para pagamento dos itens de usabilidade ficou em 6,7% do esforço total. Foi o penúltimo, à frente somente dos itens de defeito com 1,1%. O esforço para pagamento das dívidas técnicas de usabilidade ficou aproximadamente 9 vezes menor que o esforço para pagamento das dívidas de Arquitetura (58,8%), o maior esforço apurado (Tabela 4.4).

No Projeto-C, também foram identificados 6 tipos de DT, o esforço para pagamento das dívidas técnicas de usabilidade ficou em 6,2%. Penúltima posição, à frente somente das dívidas de requisitos com 3,2%. Seguindo o mesmo raciocínio, o esforço para pagamento das dívidas técnicas de usabilidade ficou aproximadamente 5 vezes menor que o maior esforço que foi a dívida de infraestrutura (31,8%), (Tabela 4.4).

Por fim, no Projeto-D, foram identificados 8 tipos de DT. A dívida técnica de usabilidade representou 5,1% do total da dívida apurada para o projeto, à frente das dívidas de design (4,4%), documentação (3,3%) e teste (0,8%). O esforço para pagamento das dívidas técnicas de usabilidade ficou aproximadamente 6 vezes menor que o maior esforço do projeto, referente a DT de defeito (28,7%).

No valor acumulado o esforço para pagamento das dívidas técnicas de usabilidade ficou em 6,32%, à frente somente das dívidas de documentação (4,99%), design (0,43%) e teste (0,08%). Aproximadamente 5 vezes menor que o maior esforço geral, dívida de arquitetura (28,59%) (Tabela 4.4).

A apuração total dos projetos contemplou 10 tipos de DT. A dívida técnica de usabilidade ficou em 7º lugar no esforço para pagamento da dívida, ou seja um dos menores esforços apurados, com 6,32% (Tabela 4.5).

Como descrito na seção 4.4, percebe-se uma dicotomia nos itens de dívida de usabilidade em relação as demais dívidas.

Na perspectiva da quantidade de itens apurados e da sua frequência os itens de dívida

de usabilidade alcançam uma posição de destaque em 4º lugar. Ou seja, são itens frequentes. Já na perspectiva de esforço para pagamento das dívidas técnicas de usabilidade, ele é baixo, alcançando somente a 7ª posição (Tabela 4.5).

Com base nos dados apresentados, o esforço para a resolução de DT de usabilidade em projetos de desenvolvimento de software, apresenta-se baixo, 6,32% do esforço total dos projetos analisados, ao mesmo tempo que se apresenta como um tipo de dívida frequente em projetos de desenvolvimento de software.

Este cenário indica que pela ótica do gerenciamento de dívida técnica é interessante pagar essa dívida, eliminando-as do portfólio de itens de DT dos projetos e beneficiando a experiência do usuário.

Capítulo 5

CONSIDERAÇÕES FINAIS

5.1 Introdução

O objetivo deste trabalho foi, a partir da verificação de uma carência de estudos para caracterizar a ocorrência da dívida técnica de usabilidade, no contexto de desenvolvimento de software, realizar um estudo de casos múltiplos, analisando itens de DT de cinco projetos de software de quatro empresas públicas brasileiras. As subseções a seguir apresentam o resumo das constatações, suas implicações práticas, as ameaças à validade desta pesquisa e trabalhos futuros.

5.2 Resumo das constatações

Esta pesquisa discutiu a dívida técnica de usabilidade no contexto de projetos de desenvolvimento de software, em comparação a outros tipos de dívidas técnicas.

Foi constatado que itens de dívida técnica de usabilidade ocorrem com frequência em projetos de software. Neste pesquisa, estes itens tiveram ocorrência em todos os projetos, atingindo o 4º lugar dos itens mais frequentes com 13,79%. Em contrapartida o esforço apurado para pagamento das dívidas técnicas de usabilidade foi um dos mais baixos, ficando em 7º lugar, com 6,32% do esforço total.

Isso é um indicador de que pode ser interessante liquidar as dívidas técnicas de usabilidade, eliminando-as do portfólio das dívidas técnicas dos projetos, gerando um possível impacto positivo junto ao usuário. De fato, nesta pesquisa, em geral, as dívidas técnicas de usabilidade nasceram de solicitações de usuários experientes, no anseio de tornar a sua experiência com o sistema mais fácil e produtiva.

Analisamos ainda os tipos de dívida técnica de usabilidade. Foi possível observar a presença de itens de dívida técnica referentes à violação de 8 das 10 heurísticas de usabilidade propostas por Nielsen [5]. Entretanto, embora tenhamos descrito os dados observados, acreditamos que o tipo da dívida técnica de usabilidade seja fortemente dependente do tipo de projeto sendo conduzido.

5.3 Implicações

Os resultados deste trabalho de pesquisa aplicada contribuem para analisar e entender o fenômeno da dívida técnica de usabilidade. Isto é de particular importância, tendo em vista a carência de estudos em dívida técnica de usabilidade.

Os resultados indicam, no contexto desta pesquisa, que os itens de dívida técnica de usabilidade são frequentes, mas tem um baixo esforço para pagamento. Estes resultados mostram um cenário num ambiente específico de empresas sólidas e com equipes qualificadas, o que pode não ocorrer em outros cenários. Entretanto, recomendamos a gerentes de projeto que identifiquem esse tipo de dívida (frequente) e priorizem sua resolução (que envolve pouco esforço) para melhorar a experiência do usuário com o sistema sendo desenvolvido.

Este trabalho apresentou um panorama da dívida técnica de usabilidade referente à frequência, tipo e esforço de resolução. Desta forma, acreditamos ainda estar abrindo novas avenidas de investigação para trabalhos futuros nesta área.

5.4 Ameaças à validade

A seguir discutimos ameaças à validade desta pesquisa em relação à validade interna, de construção, externa e de conclusão.

- **Validade interna:** este estudo de caso foi ancorado em informações oriundas de projetos reais de empresas que utilizam as informações referentes a suas tarefas de maneira organizada, rotineira e metódica. Todas as classificações realizadas foram validadas com representantes das empresas. Neste contexto as ameaças à validade interna foram minimizadas. As estimativas de esforço estão sujeitas a erros, que poderiam comprometer a validade interna. Entretanto, foram realizadas por profissionais experientes das empresas, seguindo sua forma tradicional de realizar esta tarefa.

- **Validade de construção:** O objetivo deste estudo de casos múltiplos foi categorizar os tipos de DT de empresas distintas e analisar o esforço de cada tipo de DT e compará-los aos itens de DT de usabilidade. A métrica utilizada, homem-hora, é amplamente utilizada em projetos de software. Vale ressaltar que, para reforçar a validade de construção, em momento algum da pesquisa foi dito que o foco da pesquisa era DT de usabilidade, este cuidado foi tomando para evitar que houvesse uma seleção e ou classificação tendenciosa. Para todas as equipes o foco da pesquisa era apuração da DT em projetos de desenvolvimento de software.
- **Validade de conclusão:** Embora tenhamos realizado um estudo de casos múltiplos, envolvendo cinco projetos de quatro empresas, e tenhamos observados tendências comuns, não é possível afirmar que a saturação teórica foi alcançada. Assim, a validade de conclusão pode ser reforçada com replicações do estudo.
- **Validade externa:** O estudo envolveu cinco projetos reais e atuais que estão em plena produção em quatro empresas públicas (duas do âmbito federal e duas de âmbito regional), com equipes de TI capacitadas. No entanto, os resultados não podem ser generalizados. Para ampliar a validade externa é preciso replicar o estudo em outros projetos, com contextos diferentes, permitindo ampliar a representatividade dos resultados do estudo.

5.5 Trabalhos futuros

Segundo Rios et al. [33] há poucos estudos empíricos referentes a dívida técnica realizados em cenários reais. No caso da dívida técnica de usabilidade não existem outros trabalhos nessa direção [33]. Assim, há muito a ser pesquisado sobre dívida técnica, principalmente no contexto de dívida técnica de usabilidade.

Este trabalho apurou frequência, tipo e esforço de resolução de dívida técnica de usabilidade num ambiente específico de empresas públicas. Em trabalhos futuros este estudo poderia ser ampliado com empresas do setor privado e de portes diferentes, visando ampliar a validade de conclusão e externa da pesquisa aqui apresentada. Poderiam ainda ser conduzidas novas pesquisas, com estudos longitudinais, envolvendo estudos de caso, acompanhando a gerência e resolução de itens de dívida técnica de usabilidade ao longo do tempo para ampliar a compreensão do fenômeno.

Referências

- [1] GLPI Gerenciamento de TI. Disponível em <https://glpi-project.org/pt-br/>. Acesso em: 13-05-2019.
- [2] IBM Rational Unified Process - Best Practices for Software Development Teams. . Disponível em https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf Acesso em: 25-11-2018.
- [3] JBoss Enterprise Application Platform . Disponível em <https://www.redhat.com/pt-br/technologies/jboss-middleware/application-platform> Acesso em: 13-05-2019.
- [4] Microsoft Internet Information Services . Disponível em <https://www.iis.net/> Acesso em: 14-05-2019.
- [5] Nielsen Norman Group 10 Heuristics for User Interface Design: Article by Jakob Nielsen. Disponível em <http://www.nngroup.com/articles/ten-usability-heuristics/> Acesso em: 28-06-2018.
- [6] OpenProject Collaborative Project Management . Disponível em <https://www.openproject.org/> Acesso em: 13-05-2019.
- [7] Oracle Database . Disponível em <https://www.oracle.com/database/> Acesso em: 13-05-2019.
- [8] RedMine Project management web application. Disponível em <https://www.redmine.org/> Acesso em: 07-01-2019.
- [9] Scrum Framework . Disponível em <https://www.scrum.org/> Acesso em: 14-05-2019.
- [10] Symfony Framework. Disponível em <https://symfony.com/> Acesso em: 13-05-2019.
- [11] Trac Integrated SCM Project Management . Disponível em <https://trac.edgewall.org/> Acesso em: 14-05-2019.
- [12] AMPATZOGLOU, A.; AMPATZOGLOU, A.; CHATZIGEORGIOU, A.; AVGERIOU, P. The financial aspect of managing technical debt: A systematic literature review. *Information and Software Technology* 64, 1 (2015), 52–73.
- [13] CARVALHO, G. Passo a passo do gerenciamento de projetos. *Gestão Tecnologia de Projetos* 2, 1 (2007), 124–138.

- [14] CUNNINGHAM, W. The WyCash portfolio management system. In *Addendum to the Proceedings on Object-oriented Programming Systems, Languages, and Applications (Addendum)* (1992), OOPSLA '92, ACM, pp. 29–30.
- [15] FALESSI, D.; KRUCHTEN, P.; AVGERIOU, P. Introduction to the special issue on technical debt in software systems. *Journal of Systems and Software* 120 (2016), 154–155.
- [16] FERNÁNDEZ-SÁNCHEZ, C.; GARBAJOSA, J.; YAGÜE, A.; PEREZ, J. Identification and analysis of the elements required to manage technical debt by means of a systematic mapping study. *Journal of Systems and Software* 124 (2017), 22–38.
- [17] GUO, Y.; SEAMAN, C.; Q.B. DA SILVA, F. Costs and obstacles encountered in technical debt management – a case study. *Journal of Systems and Software* 120 (2016), 156–169.
- [18] INTERNATIONAL STANDARDS ORGANIZATION. *ISO/IEC 12119:1994 - Tecnologia da informação - Pacotes de software - Requisitos e testes de qualidade*, 1994.
- [19] INTERNATIONAL STANDARDS ORGANIZATION. *ISO/IEC 9126-1:2001 – Software engineering – Product quality – Part 1: Quality model*, 2001.
- [20] INTERNATIONAL STANDARDS ORGANIZATION. *ISO/IEC 25010:2011(en) – Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*, 2011.
- [21] KITCHENHAM, B.A., C. S. *Guidelines for performing systematic literature reviews in software engineering*. EBSE Technical Report - Keele University, 2007.
- [22] KRUCHTEN, P.; NORD, R.; OZKAYA, I. Technical debt: From metaphor to theory and practice. *IEEE Software* 29, 6 (2012), 18–21.
- [23] LEHMAN, M. M. On understanding laws, evolution, and conservation in the large-program life cycle. *Journal of Systems and Software* 1 (1979), 213–221.
- [24] LEHMAN, M. M. Feedback in the software evolution process. *Information and Software Technology* 38 (2012), 681–686.
- [25] LI, Z.; AVGERIOU, P.; LIANG, P. A systematic mapping study on technical debt and its management. *Journal of Systems and Software* 101 (2015), 193–220.
- [26] LILIANA GUZMAN, ANNA MARIA VOLLMER, M. C. M. G. Formative evaluation of a tool for managing software quality. 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 297–306.
- [27] MACK, R. NIELSEN, J. Usability inspection methods. New York, NY: John Wiley Sons, 1994.
- [28] MOLICH, R.; NIELSEN, J. Improving a human-computer dialogue. *Commun. ACM* 33, 3 (1990), 338–348.
- [29] N. RIOS, M. M.; SPINOLA, R. *A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners*, vol. 102. Information and Software Technology, 2018.

- [30] NIELSEN, J. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, ACM, pp. 152–158.
- [31] PARNAS, D. Software aging. In *IEEE Proceedings of 16th International Conference on Software Engineering* (Sorrento, Italy, Italy, May 1994), pp. 279–287.
- [32] POTDAR, A.; SHIHAB, E. An exploratory study on self-admitted technical debt. In *IEEE 30th International Conference on Software Maintenance and Evolution, ICSME 2014*; (Victoria, BC, Canadá, September 2014), pp. 91–100.
- [33] RIOS, N. S. R.; MENDES, T. S.; DE MENDONÇA, M. G.; SPÍNOLA, R. O.; SHULL, F.; SEAMAN, C. Identification and management of technical debt: A systematic mapping study. *Information and Software Technology* 70 (2016), 100–121.
- [34] ROBSON, C. K. M. *Real world research*. John Wiley Sons Ltd, London, 2011.
- [35] RUNESON, P. HOST, M. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering* 14, 2 (2009), 131.
- [36] RUNESON, PER; HOST, M. R. A. R. B. *Case Study Research in Software Engineering - Guidelines and Examples*. John Wiley Sons, Inc, 2012.
- [37] SEAMAN, C.; GUO, Y. Chapter 2 - measuring and monitoring technical debt. In *Advances in Computers*, M. V. Zelkowitz, Ed., vol. 82. Elsevier, 2011, pp. 25–46.
- [38] SEAMAN, C. B. Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering* 25 (07 1999), 557–572.
- [39] SOMMERVILLE, I. *Software Engineering*. Addison Wesley Publishing Company, 2005.
- [40] SZYKARSKI, A. Ted theodoropoulos on managing technical debt successfully, 2012. Disponível em <http://www.ontechnicaldebt.com/blog/ted-theodoropoulos-on-managing-technical-debt-successfully/> Acesso em: 17-09-2018.
- [41] TOM, E.; AURUM, A.; VIDGEN, R. A consolidated understanding of technical debt. In *Proceedings of the 20th European Conference on Information Systems (ECIS'12)*, AIS Electronic Library (2012), pp. 1–13.
- [42] TOM, E.; AURUM, A.; VIDGEN, R. An exploration of technical debt. *Journal of Systems and Software* 86, 6 (2013), 1498–1516.
- [43] YLI-HUUMO, J.; MAGLYAS, A.; SMOLANDER, K. How do software development teams manage technical debt? – an empirical study. *Journal of Systems and Software* 120 (2016), 195–218.
- [44] Z. LI, P. LIANG, P. A. Architectural technical debt identification based on architecture decisions and change scenarios. In *12th Work. IEEEIFIP Conf. Softw. Archit. WICSA*. (2015), vol. 101, Journal of Systems and Software, pp. 193–220.

-
- [45] ZAZWORKA, N.; SPÍNOLA, R. O.; VETRO', A.; SHULL, F.; SEAMAN, C. A case study on effectively identifying technical debt. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering* (2013), EASE '13, ACM, pp. 42–47.

APÊNDICE A - ITENS APURADOS NA FASE PRELIMINAR DA PESQUISA.

A.1 Lista de itens de dívida técnica selecionados e classificados preliminarmente, por projeto

Nessa seção serão exibidos os itens de dívida técnica selecionados e classificados preliminarmente, descritos detalhadamente na fase 2 (ver subseção 3.3.2) dos procedimentos para coleta de dados.

Cada tabela representa a lista de itens de DT de um projeto pesquisado, classificado por tipo de DT de acordo com a taxonomia adotada[33]:

- Tabela A.1: Lista preliminar Projeto-A - parte 1 de 4.
- Tabela A.2: Lista preliminar Projeto-A - parte 2 de 4.
- Tabela A.3: Lista preliminar Projeto-A - parte 3 de 4.
- Tabela A.4: Lista preliminar Projeto-A - parte 4 de 4.
- Tabela A.5: Lista preliminar Projeto-B.
- Tabela A.6: Lista preliminar Projeto-C.
- Tabela A.7: Lista preliminar Projeto-D - parte 1 de 2.
- Tabela A.8: Lista preliminar Projeto-D - parte 2 de 2.
- Tabela A.9: Lista preliminar Projeto-E.

Tabela A.1: Lista preliminar Projeto-A - parte 1 de 4.

	#	Tipo	Dívida Técnica	Prioridade	Descrição
1	2174	Outras Manutenções	Processos	Alta	REVERTER A SITUAÇÃO DA ORDEM DE Serviço 002077/2014
2	2239	Outras Manutenções	Processos	Alta	REVERTER A SITUAÇÃO DA ORDEM DE Serviço 002077/2014
3	4492	Suporte	Processos	Alta	SEC. DE CONTROLE EXTERNO - Script para Alteração de responsável da O.S
4	4684	Suporte	Processos	Alta	SEC. DE CONTROLE EXTERNO- Alterar Data das O.S
5	3954	Outras Manutenções	Processos	Alta	Criar script para permitir a edição de células de quadros bloqueados para edição
6	8610	Nova Funcionalidade	Processos	Alta	Fórmula Despesa Autorizada - Exceto intraorçamentária
7	9419	Manutenção Evolutiva	Processos	Alta	Ajuste na Fórmula: Valor da receita do FUNDEB e Gastos com magistério (Nova contabilidade)
8	11867	Manutenção Evolutiva	Processos	Alta	Quadro 1.1 ? Créditos Adicionais do Período
9	11869	Manutenção Evolutiva	Processos	Alta	Quadro 1.3 - Créditos Adicionais - por Fonte de Financiamento
10	5305	Outras Manutenções	Processos	Alta	Publicação Conex-e War em produção – VERSÃO 3.0.0.6
11	3588	Nova Demanda	Processos	Baixa	Quartzo - incluir perfil
12	2228	Outras Manutenções	Processos	Normal	REVERTER A SITUAÇÃO DAS ORDEM DE ServiçoS/relatórioTÉCNICO.
13	5094	Nova Funcionalidade	Processos	Normal	Criar um perfil para a Ouvidoria
14	8843	Manutenção Evolutiva	Processos	Normal	CONTAS GOVERNO - RESULTADO ORÇAMENTÁRIO
15	7890	Manutenção Evolutiva	Processos	Urgente	Atendimento do chamado #7882
16	8488	Manutenção Evolutiva	Processos	Urgente	Quadro - Quociente de disponibilidade financeira - Exceto RPPS
17	1476	Nova Funcionalidade	Arquitetura	Normal	Alterar a forma de envio de e-mails
18	1477	Nova Funcionalidade	Arquitetura	Normal	Evitar duplicidade de OS
19	893	Manutenção Evolutiva	Arquitetura	Normal	Mudança nas atividades referentes ao Multas
20	7573	Nova Funcionalidade	Arquitetura	Normal	Implementar uma das formas do padrão de projetos "Singleton" para o EntityManagerFactory.
21	2884	Manutenção Evolutiva	Código	Alta	Alteração da fórmula do quadro Tempo de Contribuição para o RPPS
22	3395	Nova Funcionalidade	Código	Alta	Vinculação entre modelos de relatório para permitir a importação de conteúdos no relatório técnico
23	4195	Nova Funcionalidade	Código	Alta	Criar funcionalidade para registro de erros/exceções na aplicação bem como gerar alertas por e-mail
24	4460	Manutenção Evolutiva	Código	Alta	Relatórios de auditoria (tempestivo) para importação do Contas de Gestão
25	861	Manutenção Evolutiva	Código	Normal	Atividades da OS
26	265	Nova Funcionalidade	Código	Normal	Alteração de data legal para envios sem data certa
27	639	Manutenção Evolutiva	Código	Normal	Relatório gerencial ? RNI
28	483	Manutenção Evolutiva	Código	Normal	Consulta de RNI
29	1288	Manutenção Evolutiva	Código	Normal	Perfil Cadastrar Ordem de Serviços
30	1754	Outras Manutenções	Código	Normal	Atualização de dados da tabela [param_despacho]
31	2098	Manutenção Evolutiva	Código	Normal	Alteração script de fórmula Conex-e - Atendimento do redmine #2097
32	2429	Nova Funcionalidade	Código	Normal	Script - produção - atendimento do Redmine 2419
33	1283	Nova Funcionalidade	Código	Normal	Implementação das regras novas para o pdf dos líderes
34	1260	Manutenção Evolutiva	Código	Normal	Modelo da informação dos líderes
35	3218	Outras Manutenções	Código	Normal	Script - produção - atendimento do Redmine 3187
36	2971	Manutenção Evolutiva	Código	Normal	Tipo Célula
37	3445	Manutenção Evolutiva	Código	Normal	Nova variável

Tabela A.2: Lista preliminar Projeto-A - parte 2 de 4.

38	4224	Nova Funcionalidade	Código	Normal	Criação de atributo
39	4899	Manutenção Evolutiva	Código	Normal	Boletos - 2014 - Engenharia
40	4050	Manutenção Evolutiva	Código	Normal	Alteração da forma de cobrança das inadimplências
41	2963	Nova Funcionalidade	Código	Normal	Alteração da forma de cobrança das inadimplências
42	713	Manutenção Evolutiva	Código	Normal	Cadastramento de inadimplência sem data certa
43	681	Nova Funcionalidade	Código	Normal	relatório de acompanhamento das RNI's lançadas
44	2666	Nova Funcionalidade	Código	Normal	Geração automática de cadastro de inadimplência
45	2930	Nova Demanda	Código	Normal	Prorrogação de prazo - Arquivos tempestivos
46	1241	Manutenção Evolutiva	Código	Normal	Exclusão de inadimplência incluída manualmente (por dentro da proposta)
47	3918	Manutenção Evolutiva	Código	Urgente	Contas de Governo - com o LRF
48	7712	Manutenção Evolutiva	Código	Urgente	Atendimento do chamado #7598 (Genexus)
49	11527	Outras Manutenções	Código	Urgente	relatório de todas as irregularidades nos relatórios de contas de governo 2015
50	2290	Outras Manutenções	Defeito	Alta	RECEITAS DE CAPITAL – OPERAÇÕES DE CRÉDITO
51	2311	Outras Manutenções	Defeito	Alta	Anexo: RECEITA Quadro: Deduções não aparece os valores
52	2424	Outras Manutenções	Defeito	Alta	Script - produção - atendimento do Redmine 2422
53	2432	Outras Manutenções	Defeito	Alta	Script - produção - atendimento do Redmine 2105
54	2597	Manutenção Evolutiva	Defeito	Alta	Script - produção - atendimento do Redmine 2568
55	3400	Manutenção Corretiva (B)	Defeito	Alta	Link inserido não está funcionando
56	2977	Manutenção Corretiva (B)	Defeito	Alta	Link inserido não está funcionando
57	3642	Manutenção Corretiva (B)	Defeito	Alta	não está exibindo irregularidade na conclusão do relatório PDF
58	4168	Manutenção Evolutiva	Defeito	Alta	Atendimento do chamado #4110 E-mail automático - OS Finalizada
59	5200	Manutenção Corretiva (B)	Defeito	Alta	A tabela de Entidade está ficando com registro "Lockado"
60	5028	Outras Manutenções	Defeito	Alta	Tratamento das mensagens no PDF do chamado #4991 Erro na base de homologação
61	7887	Manutenção Corretiva (B)	Defeito	Alta	Atendimento do chamado #7885
62	7919	Manutenção Corretiva (B)	Defeito	Alta	Atendimento do chamado #7895 Corrigir Sumário – relatório Contas de Governo
63	2151	Manutenção Corretiva (B)	Defeito	Imediata	O sistema não permite adicionar uma irregularidade no item de verificação automático
64	3475	Manutenção Corretiva (B)	Defeito	Imediata	Listagem de relatórios para revisão e informação do secretário-subsecretário
65	10297	Manutenção Corretiva (B)	Defeito	Imediata	Gravação de documentos na tabela documento elaborado
66	8990	Manutenção Evolutiva	Defeito	Imediata	Erro Cálculo do item de verificação Contas de Governo
67	2160	Manutenção Evolutiva	Defeito	Normal	Alterar função Sub Secretário
68	2777	Manutenção Evolutiva	Defeito	Normal	Ajustar exibição de irregularidade no RT quando a mesma recebe classificação diferentes
69	3636	Manutenção Corretiva (B)	Defeito	Normal	Erro ao finalizar relatório Técnico
70	3693	Manutenção Corretiva (B)	Defeito	Normal	Erro ao adicionar nova irregularidade em um item de Verificação Manual.
71	3833	Manutenção Corretiva (B)	Defeito	Normal	Remover justificativa automática dos textos inseridos nos campos de texto.
72	3929	Manutenção Corretiva (B)	Defeito	Normal	Erro de sessão na tela de login
73	1304	Manutenção Corretiva (B)	Defeito	Normal	Alterações nos PDF's da defesa
74	4526	Manutenção Corretiva (B)	Defeito	Normal	não esta salvando a ordenação dos apêndices.
75	4548	Manutenção Evolutiva	Defeito	Normal	Contagem incorreta de prazo
76	4858	Manutenção Corretiva (B)	Defeito	Normal	O sistema está permitindo finalizar o RT sem informar os dados na análise de Defesa/Redefesa!
77	5022	Manutenção Corretiva (B)	Defeito	Normal	Erro ao inserir informação do secretário - base de homologação

Tabela A.3: Lista preliminar Projeto-A - parte 3 de 4.

78	3327	Manutenção Corretiva (B)	Defeito	Normal	não está cadastrando arquivos modelo padrão pelo coordenador
79	3102	Manutenção Evolutiva	Defeito	Urgente	Favor elaborar os scripts que tratam o concurso
80	4254	Manutenção Evolutiva	Design	Normal	PDF do relatório de análise de defesa
81	2801	Nova Demanda	Documentação	Alta	Construção da matriz de Riscos e Impactos
82	630	Documentação	Documentação	Normal	Quadros 7.1 e 7.2
83	2830	Documentação	Documentação	Normal	Documentação Enterprise Architect
84	4751	Nova Demanda	Documentação	Normal	Video aula - Assinatura dentro do sistema
85	4484	Documentação	Documentação	Normal	Solicito elaboração de manual para divulgação do LOG
86	2023	Outras Manutenções	Infraestrutura	Normal	Alteração da situação e desbloqueio dos relatórios técnicos
87	7281	Manutenção Evolutiva	Requisitos	Alta	Nova funcionalidade - Cadastro de e-mail para alertas
88	6156	Nova Demanda	Requisitos	Alta	Extrato de Informações para MPC
89	11690	Manutenção Evolutiva	Requisitos	Alta	Manutenção para alterações dos processos no SIMULTANEO
90	3589	Nova Funcionalidade	Requisitos	Baixa	Quartzo - incluir perfil - terceirizados
91	4476	Manutenção Evolutiva	Requisitos	Imediata	- QUADROS - Aplicar as alterações da estrutura/modelo nos relatórios PROCESSADOS;
92	4456	Manutenção Evolutiva	Requisitos	Imediata	Reprocessar quadros
93	4412	Nova Funcionalidade	Requisitos	Imediata	Notificações, Comunicação
94	880	Manutenção Evolutiva	Requisitos	Normal	Cadastro de OS
95	2959	Nova Funcionalidade	Requisitos	Normal	Gerar Boleto - Cobrança de envios na situação FORA DO PRAZO + não ENVIADO no caso dos arquivos tempestivos
96	4510	Nova Funcionalidade	Requisitos	Normal	Link para trazer PDF do sistema base
97	4283	Manutenção Evolutiva	Requisitos	Normal	Sistema base - ESTADO
98	9444	Nova Demanda	Requisitos	Normal	Especificação de Requisitos de Negócio - Nova Ordem de Serviço
99	2807	Suporte	Requisitos	Urgente	Problema ao alterar responsável na ordem de Serviço vencida (situação - REVISADO).
100	3979	Manutenção Evolutiva	Requisitos	Urgente	Atendimento do chamado - #3978 Tomar opcional a inclusão de um arquivo (modelo)
101	1533	Teste	Testes	Alta	Verificar e testar o problema "Distribuição de O.S. para RNI" do chamado da Secex Atos de pessoal
102	1578	Teste	Testes	Alta	Executar roteiro de teste da Versão 2.0.0
103	3105	Nova Demanda	Testes	Imediata	Elaborar roteiros de Teste - Ordem de Serviço Manual e Vinculação entre Modelos de Relatório
104	3272	Nova Demanda	Testes	Imediata	Elaborar roteiros de Teste - Importação entre relatórios
105	3273	Nova Demanda	Testes	Imediata	Elaborar roteiros de Teste - Gerar Representação de Natureza Inter - RNI
106	2366	Outras Manutenções	Usabilidade	Alta	Script produção - Atualização das situações de Ordem de Serviço e relatório técnico (Finalizada x Concluído)
107	3831	Manutenção Evolutiva	Usabilidade	Alta	Atendimento do chamado - #3619 Alteração de item obrigatório no cadastro de responsáveis
108	3845	Manutenção Evolutiva	Usabilidade	Alta	Atendimento do chamado - Manutenção Evolutiva #3807 Incluir novo local de atividade
109	5265	Manutenção Evolutiva	Usabilidade	Alta	Atendimento do chamado - Manutenção Evolutiva #3236: Novo parâmetro no modelo - não SE APLICA Tópico não se aplica
110	3399	Manutenção Evolutiva	Usabilidade	Alta	Gráficos nos quadros
111	6635	Manutenção Evolutiva	Usabilidade	Alta	Correção na descrição das Propostas de RNI de 2015
112	2956	Manutenção Evolutiva	Usabilidade	Alta	Gráficos
113	891	Manutenção Evolutiva	Usabilidade	Baixa	Filtro de propostas
114	2349	Nova Funcionalidade	Usabilidade	Normal	Correção de mensagem de Item de Verificação em Processamento de relatório Técnico
115	6178	Manutenção Evolutiva	Usabilidade	Normal	Informar dados processo cadastramento da OS

Tabela A.4: Lista preliminar Projeto-A - parte 4 de 4.

116	4973	Manutenção Evolutiva	Usabilidade	Normal	Criar validação para "permitir não se aplica" quando não houver nova citação - DEFESA
117	1293	Manutenção Evolutiva	Usabilidade	Normal	Possibilitar salvamento dos textos de forma automática
118	1305	Manutenção Evolutiva	Usabilidade	Normal	Criar parâmetro para os textos do PDF da proposta e da defesa
119	2972	Manutenção Evolutiva	Usabilidade	Normal	Texto da conclusão
120	7035	Manutenção Evolutiva	Usabilidade	Normal	Permitir Ordenar de Itens de Verificação em tópicos diferentes
121	7959	Nova Funcionalidade	Documentação	Normal	Reestruturação da Ordem de Serviço - Análise, requisitos e arquitetura
122	9646	Nova Funcionalidade	Documentação	Normal	(NOVA OS) Projeto Nova Ordem de Serviço
123	19927	Documentação	Documentação	Normal	Atualizar Roteiros de Teste
124	4996	Documentação	Documentação	Normal	Atualização da Documentação no Enterprise Architect
125	20801	Manutenção Corretiva (B)	Defeito	Urgente	CONCLUSÃO DE OS FINALIZADA POR PRAZO
126	18310	Manutenção Evolutiva	Usabilidade	Imediata	Informar a Nota de Conformidade na tela do Feedback e no PDF
127	11917	Nova Demanda	Requisitos	Normal	OS de Auditoria para integração com o SIS-AUD

Tabela A.5: Lista preliminar Projeto-B

	#	Tipo de DT	Tipo	Prioridade	Título
1	861	Arquitetura	Manutenção Evolutiva	Normal	Atividades da OS
2	265	Código	Nova Funcionalidade	Normal	Alteração de data legal para envios sem data certa
3	1241	Código	Manutenção Evolutiva	Normal	Exclusão de inadimplência incluída manualmente (por dentro da proposta)
4	1283	Código	Nova Funcionalidade	Normal	Implementação das regras novas para o pdf dos líderes
5	1305	Código	Manutenção Evolutiva	Normal	Criar parâmetro para os textos do PDF da proposta e da defesa
6	4548	Defeito	Manutenção Evolutiva	Normal	Contagem incorreta de prazo
7	4899	Defeito	Manutenção Evolutiva	Normal	Boletos - 2014 - Engenharia
8	681	Requisitos	Nova Funcionalidade	Normal	Relatório de acompanhamento das RNI's lançadas
9	713	Requisitos	Manutenção Evolutiva	Normal	Cadastramento de inadimplência sem data certa
10	893	Requisitos	Manutenção Evolutiva	Normal	Mudança nas atividades referentes ao Multas
11	1260	Requisitos	Manutenção Evolutiva	Normal	Modelo da informação dos líderes
12	2930	Requisitos	Nova Demanda	Normal	Prorrogação de prazo - Arquivos tempestivos
13	2959	Requisitos	Nova Funcionalidade	Normal	Gerar Boletos - Cobrança de envios na situação FORA DO PRAZO + Não ENVIADO no caso dos
14	2972	Requisitos	Manutenção Evolutiva	Normal	Texto da conclusão
15	3918	Requisitos	Manutenção Evolutiva	Urgente	Contas de Governo - com o LRF
16	4050	Requisitos	Manutenção Evolutiva	Normal	Alteração da forma de cobrança das Inadimplências
17	4283	Requisitos	Manutenção Evolutiva	Normal	Sistema Base - ESTADO
18	2963	Serviço	Nova Funcionalidade	Normal	Criar serviço para atualização de Inadimplências "Manuais"
19	639	Usabilidade	Manutenção Evolutiva	Normal	Relatório gerencial - RNI
20	891	Usabilidade	Manutenção Evolutiva	Baixa	Filtro de propostas
21	1293	Usabilidade	Manutenção Evolutiva	Normal	Possibilitar salvamento dos textos de forma automática
22	2666	Usabilidade	Nova Funcionalidade	Normal	Geração automática de cadastro de inadimplência
23	4254	Usabilidade	Manutenção Evolutiva	Normal	PDF do relatório de análise de defesa
24	6635	Usabilidade	Manutenção Evolutiva	Alta	Correção na descrição das Propostas de RNI de 2015

Tabela A.6: Lista preliminar Projeto-C

#	Ticket	Projeto	Tipo de DT	Título
1	60	Projeto-C	Código	Inclusão das Moções na página de Projetos em Tramitação do Website
2	71	Projeto-C	Código	Integração do sistema de Catraca com o Ponto
3	41	Projeto-C	Código	Participação Popular na Intranet/Website
4	45	Projeto-C	Código	Ajuste nos Critérios do Motivo Afastamento
5	230	Projeto-C	Defeito	Relatório de Indenização Trabalhista está Inconsistente
6	231	Projeto-C	Defeito	Os PDF's de algumas proposições não existem
7	95	Projeto-C	Defeito	Corrigir Importação da qualificação cadastral para o eSocial
8	G195	Projeto-C	Defeito	Leis de 2018 não aparecem no Aplicativo da Fidelity
9	165	Projeto-C	Defeito	Garantir Consistência entre o Dump SQL e o Banco de Dados da Intranet
10	180	Projeto-C	Infraestrutura	Atualização de Versão do Framework Symfony para a 3.4
11	188	Projeto-C	Infraestrutura	Substituição do Bower pelo Yarn + Webpack
12	179	Projeto-C	Infraestrutura	Atualização de Versão das Chamadas em Symfony para a 3.4
13	178	Projeto-C	Infraestrutura	Atualização de Versão das Chamadas em Symfony para a 3.4
14	177	Projeto-C	Infraestrutura	Atualização de Versão do Framework Symfony para a 3.4
15	176	Projeto-C	Infraestrutura	Atualização de Versão das Chamadas em Symfony para a 3.4
16	175	Projeto-C	Infraestrutura	Atualização de Versão do Framework Symfony para a 3.4
17	189	Projeto-C	Infraestrutura	Substituição do Bower pelo Yarn + Webpack
18	174	Projeto-C	Infraestrutura	Atualização de Versão do Framework Symfony para a 3.4
19	173	Projeto-C	Infraestrutura	Atualização de Versão do Framework Symfony para a 3.4
20	100	Projeto-C	Infraestrutura	Gerador de Código
21	37	Projeto-C	Processos	Funcionalidade de Férias
22	52	Projeto-C	Processos	Funcionalidade de Licença Prêmio
23	30	Projeto-C	Requisitos	Catálogo de Telefones por Unidade Organizacional
24	51	Projeto-C	Requisitos	Funcionalidade de Agenda Médica integrada com Aposentados e Pensionistas
25	29	Projeto-C	Usabilidade	Trocar o Template dos Relatórios
26	190	Projeto-C	Usabilidade	Páginas de Exceção mais Elegantes
27	43	Projeto-C	Usabilidade	Acesso a determinados sistemas quando o usuário for Gestor de Unidade

Tabela A.7: Lista preliminar Projeto-D - parte 1 de 2.

#	id	Projetos	DT	Descrição
1	865	Projeto-D	Codigo	Demora na hora de filtrar dados em listagem extensa
2	62	Projeto-D	Codigo	Gerar log no sistema
3	1286	Projeto-D	Codigo	Configuração - Equipamentos não analisados
4	121	Projeto-D	Codigo	Correções do Sonar
5	173	Projeto-D	Codigo	Verificar deleção das instâncias dos controladores
6	205	Projeto-D	Codigo	Verificar lentidão ao trocar aba de análise
7	455	Projeto-D	Codigo	Verificar lentidão na criação do novo projeto
8	1153	Projeto-D	Codigo	Inconsistência interna SPI - Ajuste da contagem das inconsistências
9	955	Projeto-D	Codigo	Implementar aviso (email) ao administrador quando o agente de carga parar de executar
10	1504	Projeto-D	Codigo	Ajustes de eixos do gráfico de barras
11	430	Projeto-D	Codigo	Implementar Parametrização na execução debug
12	1110	Projeto-D	Codigo	Parâmetros Fusíveis - Mudança de lugar das colunas
13	854	Projeto-D	Defeito	Utilização de filtro para registro vazio
14	860	Projeto-D	Defeito	Ajustar sensibilidade do clique de acionamento do filtro [Tela Parâmetros Fusíveis]
15	997	Projeto-D	Defeito	Lentidão no filtro de detalhamento de inconsistências entre bases
16	1120	Projeto-D	Defeito	Relatório - Gráfico plotado em branco.
17	620	Projeto-D	Defeito	Erro no método de inserir aba da TabWidget
18	861	Projeto-D	Defeito	Inserir tooltip com título do grid [Janela Parâmetros Fusíveis]
19	323	Projeto-D	Defeito	Testar atualização dos objetos do Modelo
20	723	Projeto-D	Defeito	Verificar atualização dos modelos de engenharias.
21	1104	Projeto-D	Defeito	Demonstrativo de Inconsistência P-76 - Problema no gráfico
22	1121	Projeto-D	Defeito	Problema ao fechar aplicação
23	1134	Projeto-D	Defeito	Inconsistência entre bases - Problema nos números do gráfico após marcação
24	1018	Projeto-D	Defeito	Desfazer ordenação da janela de Inconsistências
25	983	Projeto-D	Defeito	Primeiro item do check-box do filtro esta vazio.
26	969	Projeto-D	Defeito	Problema na ordenação do diâmetro no detalhamento de Inconsistências
27	974	Projeto-D	Defeito	Problema no botão ""Exportar gráfico"" do histórico de análises
28	1031	Projeto-D	Defeito	Filtro aplicado em uma Análise influenciando outra Análise
29	1135	Projeto-D	Defeito	Configuração - Modelo 3D - Habilitação do botão Tabelas Customizadas
30	1155	Projeto-D	Defeito	Melhoria na Configuração dos Usuários e Contratos
31	1506	Projeto-D	Design	Atualização da TabRMP
32	1114	Projeto-D	Design	Configurar Relatório - Alteração de nome das bases
33	1154	Projeto-D	Design	Carga SM3D - Inclusão do campo ""system path""
34	263	Projeto-D	Design	Verificar deleção dos Ctrls no Agente de Carga
35	1200	Projeto-D	Documentacao	Help - Atualiza as telas com as abas de análise
36	1202	Projeto-D	Documentacao	Help - Atualizar nome da análise ""Instrumentos com loop
37	1182	Projeto-D	Documentacao	Help - Atualização (filtro coluna status)

Tabela A.8: Lista preliminar Projeto-D - parte 2 de 2.

38	136	Projeto-D	Documentacao	Documentar as classes, comentar o código existente e gerar documentação de apoio ao desenvolvimento.
39	417	Projeto-D	Documentacao	[Teste] Criação de testes de interface
40	161	Projeto-D	Requisitos	Implementar Funcionalidade de Salvar Como
41	557	Projeto-D	Requisitos	Filtragem de inconsistências dos itens especiais de tubulação do SPPID x SM3D
42	561	Projeto-D	Requisitos	Filtragem de inconsistências dos pontos de tie-in de tubulação do SPPID x SM3D
43	562	Projeto-D	Requisitos	Filtragem de inconsistências dos sistemas de comissionamento do SPPID x SM3D
44	645	Projeto-D	Requisitos	Criar modelo de planilha excel (arquivo exportado)
45	882	Projeto-D	Requisitos	Link das barras do histórico para as análises
46	1199	Projeto-D	Requisitos	Abas de análise - Duplo clique para renomear aba
47	943	Projeto-D	Requisitos	Implementar exibição dos resultados agrupados no demonstrativo de inconsistências
48	1173	Projeto-D	Requisitos	Comentário - Implementar histórico
49	1190	Projeto-D	Requisitos	Organizar exportação para xls de inconsistências
50	1186	Projeto-D	Requisitos	Incluir no gráfico de inconsistências itens não-importantes
51	351	Projeto-D	Requisitos	Concatenar nome da base com nome do servidor para conexão com a configuração editada pelo usuário
52	413	Projeto-D	Requisitos	Modificações nas configurações do projeto só podem ser feitas após teste de perfil de Usuário do Controle de Acesso
53	126	Projeto-D	Teste	Criação de testes unitários para verificar todas as conexões de banco
54	821	Projeto-D	Usabilidade	TAGs ligeiramente diferentes (Adicionar identificador nos tags)
55	920	Projeto-D	Usabilidade	Padronizar titulo dos elementos pai no grid de parâmetro fusíveis
56	970	Projeto-D	Usabilidade	Restaurando a ordenação padrão do demonstrativo de inconsistências
57	203	Projeto-D	Usabilidade	Estilizar botões de configurar relatório
58	583	Projeto-D	Usabilidade	Estilizar TabWidget
59	1261	Projeto-D	Usabilidade	Exportar para excel - Cursor de espera
60	965	Projeto-D	Usabilidade	Desabilitar botão SALVAR apos o salvamento do projeto
61	1035	Projeto-D	Usabilidade	Exibir demonstrativo de inconsistência mesmo sem gráfico.
62	1168	Projeto-D	Usabilidade	Justificativa da marcação - Interface
63	1359	Projeto-D	Usabilidade	Inserir cursor de espera
64	1499	Projeto-D	Usabilidade	Modificações de interface

Tabela A.9: Lista preliminar Projeto-E

	# Ticket	Projeto	Tipo de DT	Título
3	3-SI	Projeto-E	Código	Visualizar coordenadores inativos de um projeto
4	4-SI	Projeto-E	Defeito	Corrigir mensagem quando o projeto é retomado e submetido novamente
5	5-SI	Projeto-E	Defeito	Opção para o coordenador do programa desvincular projetos do programa
6	6-SI	Projeto-E	Defeito	Retirar quantitativo de participantes quando a ação não for realizada
7	1-SG	Projeto-E	Design	Baixa abstração
8	1-SIG	Projeto-E	Design	Baixa abstração
9	7-SI	Projeto-E	Documentação	Adicionar caixas de texto sobre cada página e/ou botão do sistema expediente
10	8-SI	Projeto-E	Processo	Abrir relatório completo ao Inserir Relatório Final
11	2-SG	Projeto-E	Requisitos	Disponibilizar dados abertos
12	3-SG	Projeto-E	Requisitos	Avaliação de relatórios de iniciação científica
13	2-SIG	Projeto-E	Requisitos	Melhorias no sistema
14	1-MN	Projeto-E	Requisitos	Alteração de cadastro
15	1-AL	Projeto-E	Requisitos	Controle de acesso
16	9-SI	Projeto-E	Requisitos	Enviar notificação ao responsável da avaliação quando nova proposta e relatório estiver disponível
17	10-SI	Projeto-E	Requisitos	Opção das Instâncias Superiores enviar para Avaliador (Parecerista) a proposta e relatório
18	11-SI	Projeto-E	Usabilidade	Listar projetos por unidade/ Responsável por unidade visualizar todas as propostas
19	12-SI	Projeto-E	Usabilidade	Cadastrar função a partir da tela de membro de equipe
20	13-SI	Projeto-E	Usabilidade	Mostrar quando a ação não foi realizada sem clicar em detalhes

APÊNDICE B - ITENS APURADOS AO FINAL DA FASE DE COLETA DE DADOS

B.1 Lista de itens de dívida técnica, por projeto, apurados ao final da fase de coleta de dados.

Nessa seção serão exibidas as listas finais de itens de dívida técnica por projeto. A lista final corresponde a lista validada pela equipe técnica, líderes e pesquisadores na fase 4 (ver subseção 3.3.4) dos procedimentos para coleta de dados:

- Tabela B.1: Lista final de itens de DT do Projeto-A.
- Tabela B.2: Lista final de itens de DT do Projeto-B.
- Tabela B.3: Lista final de itens de DT do Projeto-C.
- Tabela B.4: Lista final de itens de DT do Projeto-D - parte 1 de 2.
- Tabela B.5: Lista final de itens de DT do Projeto-D - parte 2 de 2
- Tabela B.6: Lista final de itens de DT do Projeto-E.

Tabela B.1: Lista final de itens de DT do Projeto-A

#	# Ticket	Projeto	Heurísticas de Nielsen	Tipo de DT	Descrição
1	7959	Projeto-A		Arquitetura	Reestruturação da Ordem de Serviço - Análise, requisitos e arquitetura
2	9646	Projeto-A		Arquitetura	(NOVA OS) Projeto Nova Ordem de Serviço
3	1476	Projeto-A		Código	Alterar a forma de envio de e-mails
4	8610	Projeto-A		Código	Fórmula Despesa Autorizada - Exceto intraorçamentária
5	9419	Projeto-A		Código	Ajuste na Fórmula: Valor da receita do FUNDEB e Gastos com magistério (Nova contabilidade)
6	11867	Projeto-A		Código	Quadro 1.1 ? Créditos Adicionais do Período
7	11869	Projeto-A		Código	Quadro 1.3 - Créditos Adicionais - por Fonte de Financiamento
8	8843	Projeto-A		Código	CONTAS GOVERNO - RESULTADO ORÇAMENTÁRIO
9	7882	Projeto-A		Código	Cancelamento de OS Planejada
10	8488	Projeto-A		Código	Quadro - Quociente de disponibilidade financeira - Exceto RPPS
11	11527	Projeto-A		Código	relatório de todas as irregularidades nos relatórios de contas de governo 2015
12	11690	Projeto-A		Código	Manutenção para alterações dos processos no...SIMULTANEO
13	6178	Projeto-A		Código	Criar botão ao lado do ano do processo no cadastramento da O.S
14	4548	Projeto-A		Defeito	Contagem incorreta de prazo
15	8990	Projeto-A		Defeito	Erro Cálculo do item de verificação Contas de Governo
16	19927	Projeto-A		Documentação	Atualizar Roteiros de Teste
17	4996	Projeto-A		Documentação	Atualização da Documentação no Enterprise Architect
18	4751	Projeto-A		Documentação	Assinatura dentro do Sistema – Vídeoaula
19	4476	Projeto-A		Requisitos	QUADROS - Aplicar as alterações da estrutura/modelo nos relatórios PROCESSADOS
20	4456	Projeto-A		Requisitos	Reprocessar quadros
21	9419	Projeto-A		Requisitos	OS de Auditoria para integração com o Modulo AUD
22	6156	Projeto-A		Requisitos	Extrato de informações para MPC
23	7035	Projeto-A	7	Usabilidade	Permitir Ordenar de Itens de Verificação em tópicos diferente
24	4510	Projeto-A	7	Usabilidade	Link para trazer PDF do sistema base
25	2956	Projeto-A	7	Usabilidade	Gráficos (A partir de quadros permitir a geração de gráficos (pizza, colunas, linhas, barras).
26	2349	Projeto-A	9	Usabilidade	Correção de mensagem de Item de Verificação em Processamento de Relatório Técnico

Tabela B.2: Lista final de itens de DT do Projeto-B

#	Ticket	Projeto	Heurísticas de Nielsen	Tipo de DT	Título
1	861	Projeto B		Arquitetura	Atividades da OS
2	265	Projeto B		Código	Alteração de data legal para envios sem data certa
3	1241	Projeto B		Código	Exclusão de inadimplência incluída manualmente (por dentro da proposta)
4	1283	Projeto B		Código	Implementação das regras novas para o pdf dos líderes
5	1305	Projeto B		Código	Criar parâmetro para os textos do PDF da proposta e da defesa
6	4254	Projeto B		Código	PDF do relatório de análise de defesa
7	4548	Projeto B		Defeito	Contagem incorreta de prazo
8	4899	Projeto B		Defeito	Boletos - 2014 - Engenharia
9	681	Projeto B		Requisitos	Relatório de acompanhamento das RNI's lançadas
10	713	Projeto B		Requisitos	Cadastramento de inadimplência sem data certa
11	893	Projeto B		Requisitos	Mudança nas atividades referentes as Multas
12	1260	Projeto B		Requisitos	Modelo da informação dos líderes
13	2930	Projeto B		Requisitos	Prorrogação de prazo - Arquivos tempestivos
14	2959	Projeto B		Requisitos	Gerar Boleto - Cobrança de envios na situação FORA DO PRAZO + NÃO ENVIADO no caso dos arquivos tempestivos
15	2972	Projeto B		Requisitos	Texto da conclusão
16	3918	Projeto B		Requisitos	Contas de Governo - com o LRF
17	4050	Projeto B		Requisitos	Alteração da forma de cobrança das Inadimplências
18	4283	Projeto B		Requisitos	Criar um serviço para que as Inadimplências manuais sejam atualizadas automaticamente sistema de origem.
19	2963	Projeto B		Serviço	Criar serviço para atualização de Inadimplências "Manuais"
20	639	Projeto B	7	Usabilidade	Relatório gerencial - RNI
21	891	Projeto B	6	Usabilidade	Filtro de propostas
22	1293	Projeto B	5	Usabilidade	Possibilitar salvamento dos textos de forma automática
23	2666	Projeto B	7	Usabilidade	Geração automática de cadastro de inadimplência
24	6635	Projeto B	2	Usabilidade	Correção na descrição das Propostas de RNI de 2015

Tabela B.3: Lista final de itens de DT do Projeto-C

#	# Ticket	Projeto	Heurísticas de Nielsen	Tipo de DT	Descrição
1	60	Projeto-C		Código	Inclusão das Moções na página de Projetos em Tramitação do Website
2	71	Projeto-C		Código	Integração do sistema de Catraca com o Ponto
3	41	Projeto-C		Código	Participação Popular na Intranet/Website
4	45	Projeto-C		Código	Ajuste nos Critérios do Motivo Afastamento
5	230	Projeto-C		Defeito	Relatório de Indenização Trabalhista está Inconsistente
6	231	Projeto-C		Defeito	Os PDF's de algumas proposições não existem
7	95	Projeto-C		Defeito	Corrigir Importação da qualificação cadastral para o eSocial
8	G195	Projeto-C		Defeito	Leis de 2018 não aparecem no Aplicativo da Fidelity
9	165	Projeto-C		Defeito	Garantir Consistência entre o Dump SQL e o Banco de Dados da Intranet
10	180	Projeto-C		Infraestrutura	Atualização de Versão do Framework Symphony para a 3.4
11	188	Projeto-C		Infraestrutura	Substituição do Bower pelo Yarn + Webpack
12	179	Projeto-C		Infraestrutura	Atualização de Versão das Chamadas em Symphony para a 3.4
13	178	Projeto-C		Infraestrutura	Atualização de Versão das Chamadas em Symphony para a 3.4
14	177	Projeto-C		Infraestrutura	Atualização de Versão do Framework Symphony para a 3.4
15	176	Projeto-C		Infraestrutura	Atualização de Versão das Chamadas em Symphony para a 3.4
16	175	Projeto-C		Infraestrutura	Atualização de Versão do Framework Symphony para a 3.4
17	189	Projeto-C		Infraestrutura	Substituição do Bower pelo Yarn + Webpack
18	174	Projeto-C		Infraestrutura	Atualização de Versão do Framework Symphony para a 3.4
19	173	Projeto-C		Infraestrutura	Atualização de Versão do Framework Symphony para a 3.4
20	100	Projeto-C		Infraestrutura	Gerador de Código
21	37	Projeto-C		Processo	Funcionalidade de Férias
22	52	Projeto-C		Processo	Funcionalidade de Licença Prêmio
23	30	Projeto-C		Requisitos	Catálogo de Telefones por Unidade Organizacional
24	51	Projeto-C		Requisitos	Funcionalidade de Agenda Médica integrada com Aposentados e Pensionistas
25	29	Projeto-C	8	Usabilidade	Trocar o Template dos Relatórios do SAHRA
26	190	Projeto-C	8,9	Usabilidade	Páginas de Exceção mais Elegantes
27	43	Projeto-C	7	Usabilidade	Acesso a determinados sistemas quando o usuário for Gestor de Unidade

Tabela B.4: Lista final de itens de DT do Projeto-D - parte 1 de 2

#	# Ticket	Projeto	Heurística Nielsen	Tipo DT	Descrição
1	997	Projeto-D		Arquitetura	Lentidão no filtro de detalhamento de inconsistências entre bases
2	865	Projeto-D		Arquitetura	Demora na hora de filtrar dados em listagem extensa
3	1154	Projeto-D		Arquitetura	Carga SM3D - Inclusão do campo ""system path"
4	1155	Projeto-D		Arquitetura	Melhoria na Configuração dos Usuários e Contratos
5	62	Projeto-D		Código	Gerar log no sistema
6	173	Projeto-D		Código	Verificar deleção das instâncias dos controladores
7	205	Projeto-D		Código	Verificar lentidão ao trocar aba de análise
8	263	Projeto-D		Código	Verificar deleção dos Ctrls no Agente de Carga
9	455	Projeto-D		Código	Verificar lentidão na criação do novo projeto
10	1153	Projeto-D		Código	Inconsistência interna SPI - Ajuste da contagem das inconsistências
11	854	Projeto-D		Defeito	Utilização de filtro para registro vazio
12	1120	Projeto-D		Defeito	Relatório - Gráfico plotado em branco.
13	620	Projeto-D		Defeito	Erro no método de inserir aba da TabWidget
14	723	Projeto-D		Defeito	Verificar atualização dos modelos de engenharias.
15	1104	Projeto-D		Defeito	Demonstrativo de Inconsistência P-76 - Problema no gráfico
16	1121	Projeto-D		Defeito	Problema ao fechar aplicação
17	1134	Projeto-D		Defeito	Inconsistência entre bases - Problema nos números do gráfico após marcação
18	983	Projeto-D		Defeito	Primeiro item do check-box do filtro esta vazio.
19	969	Projeto-D		Defeito	Problema na ordenação do diâmetro no detalhamento de Inconsistências
20	1031	Projeto-D		Defeito	Filtro aplicado em uma Análise influenciando outra Análise
21	1135	Projeto-D		Defeito	Configuração - Modelo 3D - Habilitação do botão Tabelas Customizadas
22	920	Projeto-D		Design	Padronizar título dos elementos pai no grid de parâmetro fusíveis
23	203	Projeto-D		Design	Estilizar botões de configurar relatório
24	583	Projeto-D		Design	Estilizar TabWidget
25	861	Projeto-D		Design	Inserir tooltip com título do grid [Janela Parâmetros Fusíveis]
26	1018	Projeto-D		Design	Desfazer ordenação da janela de Inconsistências

Tabela B.5: Lista final de itens de DT do Projeto-D - parte 2 de 2

#	# Ticket	Projeto	Heurística Nielsen	Tipo DT	Descrição
27	1190	Projeto-D		Design	Organizar exportação para xls de inconsistências
28	1200	Projeto-D		Documentacao	Help - Atualiza as telas com as abas de análise
29	1202	Projeto-D		Documentacao	Help - Atualizar nome da análise ""Instrumentos com loop
30	1182	Projeto-D		Documentacao	Help - Atualização (filtro coluna status)
31	136	Projeto-D		Documentacao	Documentar as classes, comentar o código existente e gerar documentação de apoio ao desenvolvimento.
32	557	Projeto-D		Requisitos	Filtragem de inconsistências dos itens especiais de tubulação do SPPID x SM3D
33	561	Projeto-D		Requisitos	Filtragem de inconsistências dos pontos de tie-in de tubulação do SPPID x SM3D
34	562	Projeto-D		Requisitos	Filtragem de inconsistências dos sistemas de comissionamento do SPPID x SM3D
35	645	Projeto-D		Requisitos	Criar modelo de planilha excel (arquivo exportado)
36	882	Projeto-D		Requisitos	Link das barras do histórico para as análises
37	1199	Projeto-D		Requisitos	Abas de análise - Duplo clique para renomear aba
38	943	Projeto-D		Requisitos	Implementar exibição dos resultados agrupados no demonstrativo de inconsistências
39	1035	Projeto-D		Requisitos	Exibir demonstrativo de inconsistência mesmo sem gráfico.
40	1186	Projeto-D		Requisitos	Incluir no gráfico de inconsistências itens não-importantes
41	351	Projeto-D		Requisitos	Concatenar nome da base com nome do servidor para conexão com a configuração editada pelo usuário
42	413	Projeto-D		Requisitos	Modificações nas configurações do projeto só podem ser feitas após teste de perfil de Usuário do Controle de Acesso
43	126	Projeto-D		Teste	Criação de testes unitários para verificar todas as conexões de banco
44	860	Projeto-D	5	Usabilidade	Ajustar sensibilidade do clique de acionamento do filtro [Tela Parâmetros Fusíveis]
45	821	Projeto-D	6	Usabilidade	- TAGs ligeiramente diferentes (Adicionar identificador nos tags)
46	970	Projeto-D	3,5	Usabilidade	Restaurando a ordenação padrão do demonstrativo de inconsistências
47	1261	Projeto-D	1	Usabilidade	Exportar para excel - Cursor de espera
48	965	Projeto-D	5	Usabilidade	Desabilitar botão SALVAR apos o salvamento do projeto

Tabela B.6: Lista final de itens de DT do Projeto-E

#	# Ticket	Projeto	Heurísticas de Nielsen	Tipo de DT	Título
1	1-SI	Projeto-E		Arquitetura	Constar no histórico do projeto, programa e relatórios as mudanças
2	2-SI	Projeto-E		Código	Permitir administração de Câmara enviar email pelo sistema interno
3	3-SI	Projeto-E		Código	Visualizar coordenadores inativos de um projeto
4	4-SI	Projeto-E		Defeito	Corrigir mensagem quando o projeto é retomado e submetido novamente
5	5-SI	Projeto-E		Defeito	Opção para o coordenador do programa desvincular projetos do programa
6	6-SI	Projeto-E		Defeito	Retirar quantitativo de participantes quando a ação não for realizada
7	1-SG	Projeto-E		Design	Baixa abstração
8	1-SIG	Projeto-E		Design	Baixa abstração
9	7-SI	Projeto-E		Documentação	Adicionar caixas de texto sobre cada página e/ou botão do sistema expediente
10	8-SI	Projeto-E		Processo	Abrir relatório completo ao Inserir Relatório Final
11	2-SG	Projeto-E		Requisitos	Disponibilizar dados abertos
12	3-SG	Projeto-E		Requisitos	Avaliação de relatórios de iniciação científica
13	2-SIG	Projeto-E		Requisitos	Melhorias no sistema
14	1-MN	Projeto-E		Requisitos	Alteração de cadastro
15	1-AL	Projeto-E		Requisitos	Controle de acesso
16	9-SI	Projeto-E		Requisitos	Enviar notificação ao responsável da avaliação quando nova proposta e relatório estiver disponível
17	10-SI	Projeto-E		Requisitos	Opção das Instâncias Superiores enviar para Avaliador (Parecerista) a proposta e relatório
18	11-SI	Projeto-E	3	Usabilidade	Listar projetos por unidade/ Responsável por unidade visualizar todas as propostas
19	12-SI	Projeto-E	7	Usabilidade	Cadastrar função a partir da tela de membro de equipe
20	13-SI	Projeto-E	1,6	Usabilidade	Mostrar quando a ação não foi realizada sem clicar em detalhes

APÊNDICE C - ESTIMATIVA DE ESFORÇO (HOMEM-HORA) PARA RESOLUÇÃO DA DÍVIDA TÉCNICA

C.1 Apuração das estimativas de esforço de dívida técnica, por projeto.

Nessa seção serão exibida as planilhas com os cálculos das estimativas de esforço (homem-hora) dos projetos A, C e D, apurados na fase 5 (ver subseção 3.3.5) dos procedimentos para coleta de dados:

- Tabela C.1: Estimativa de esforço de DT do Projeto-A.
- Tabela C.2: Estimativa de esforço de DT do Projeto-C.
- Tabela C.3: Estimativa de esforço de DT do Projeto-D - parte 1 de 2.
- Tabela C.4: Estimativa de esforço de DT do Projeto-D - parte 2 de 2.

Tabela C.1: Estimativa de esforço do Projeto-A

[illegible]

Tabela C.3: Estimativa de esforço do Projeto-D - parte 1 de 2

	# Ticket	Projeto	Tipo DT	Ciclos de Manutenção – Atividades Envolvidas												Esforço (Homem / Hora)
				Análise e Projeto		Arquitetura/ Banco de Dados		Desenvolvimento		Teste		Implantação / Treinamento		Gerência de Projetos		
				NP	H/H	NP	H/H	NP	H/H	NP	H/H	NP	H/H	NP	H/H	
1	997	Projeto-D	Arquitetura					1	4,0							4,0
2	865	Projeto-D	Arquitetura	1	2,0			1	4,0							6,0
3	1154	Projeto-D	Arquitetura	1	8,0	1	2,0	1	8,0							18,0
4	1155	Projeto-D	Arquitetura					1	28,0							28,0
5	62	Projeto-D	Codigo	1	4,0			1	16,0					1	2,0	22,0
6	173	Projeto-D	Codigo	1	4,0			1	16,0					1	2,0	22,0
7	205	Projeto-D	Codigo	1	4,0			1	16,0					1	2,0	22,0
8	263	Projeto-D	Codigo	1	2,0			1	20,0							22,0
9	455	Projeto-D	Codigo					1	4,0							4,0
10	1153	Projeto-D	Codigo	1	4,0	1	2,0	1	8,0							14,0
11	854	Projeto-D	Defeito					1	10,0							10,0
12	1120	Projeto-D	Defeito					1	2,0							2,0
13	620	Projeto-D	Defeito					1	4,0							4,0
14	723	Projeto-D	Defeito					1	4,0							4,0
15	1104	Projeto-D	Defeito					1	5,0							5,0
16	1121	Projeto-D	Defeito					1	8,0							8,0
17	1134	Projeto-D	Defeito													
18	983	Projeto-D	Defeito	1	2,0			1	8,0							10,0
19	969	Projeto-D	Defeito					1	1,0							1,0
20	1031	Projeto-D	Defeito					1	1,0							1,0
21	1135	Projeto-D	Defeito					1	1,0							1,0
22	920	Projeto-D	Design					1	3,0							3,0
23	203	Projeto-D	Design					1	30,0							30,0
24	583	Projeto-D	Design					1	20,0							20,0
25	861	Projeto-D	Design					1	20,0							20,0
26	1018	Projeto-D	Design					1	2,0							2,0

