Submission date: 10/04/2019
Resubmission date: 25/05/2019
Resubmission date: 12/08/2019
Resubmission date: 29/10/2019
Camera ready submission: 19/11/2019

Section: regular

1st round notification: 29/04/2019
2nd round notification: 11/07/2019
3rd round notification: 11/09/2019
4th round notification: 03/11/2019
Available online: 19/11/2019
Publication date: 31/03/2020

Dívida Técnica de Usabilidade em Projetos de Software: Um Estudo de Casos Múltiplos
Title: Technical Debt of Usability in Software Projects: A Multi-Case Study

Luiz Carlos da Fonseca Lage[1], Daniela G. Trevisan[1,2], Marcos Kalinowski[3]

[1]Instituto de Computação
Universidade Federal Fluminense (UFF)
24.210-346 – Niterói – RJ – Brasil

[2]Laboratório de Documentação Ativa e Design Inteligente
Universidade Federal Fluminense (UFF)
24.210-346 – Niterói – RJ – Brasil

[3]Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
22.451-900 – Rio de Janeiro – RJ – Brasil

{luizlage}@id.uff.br, daniela@ic.uff.br, kalinowski@inf.puc-rio.br

Abstract. Background: *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.* ***Aims:*** *The goal of this paper is to provide an initial usability debt characterization in software projects regarding its occurrence, type, and resolution effort.* ***Method:*** *We conducted a multi-case study, analyzing TD items of five software projects from four brazilian public companies.* ***Results:*** *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.* ***Conclusions:*** *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.*

Resumo. Contexto: *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*

Cite as: Lage, L. C. F., Trevisan, D. G., Kalinowski, M. (2020). Technical Debt of Usability in Software Projects: A Multi-Case Study (Dívida Técnica de Usabilidade em Projetos de Software: Um Estudo de Casos Múltiplos). iSys: Revista Brasileira de Sistemas de Informação (Brazilian Journal of Information Systems), 13(2), 34-59.

sobre a dívida de usabilidade motivou nossa pesquisa. ***Objetivo:*** *O objetivo deste artigo é fornecer uma caracterização inicial da dívida técnica de usabilidade em projetos de software com relação à sua ocorrência, tipo e esforço de resolução.* ***Método:*** *Realizamos um estudo de casos múltiplos, analisando itens de DT de cinco projetos de software de quatro empresas publicas Brasileiras.* ***Resultados:*** *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% nos 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.* ***Conclusão:*** *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.*

1. Introdução

As equipes de desenvolvimento de software enfrentam, com frequência, o desafio de entregar produtos de software sob cronogramas apertados, enquanto tentam manter a qualidade como padrão. Para lidar com restrições de tempo e recursos, os desenvolvedores de software precisam priorizar implementações, concentrando-se em requisitos cruciais e até mesmo pegar atalhos, por exemplo, postergando a documentação do projeto [Guo et al. 2016].

Além disso, durante o desenvolvimento, a qualidade do software tende a diminuir quando considera-se aspectos como sua estrutura interna, adesão às normas, documentação e facilidade de entendimento para futuras manutenções [Parnas 1994] [Lehman 1996] [Alves et al. 2016].

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

O conceito de DT contextualiza os problemas enfrentados durante a evolução do software, considerando as tarefas que não são realizadas adequadamente durante o seu desenvolvimento [Rios et al. 2018]. 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 [Falessi et al. 2016]. Um outro ponto destacado por Tom et al. [Tom et al. 2013] é que a dívida técnica pode crescer devido à incapacidade dos desenvolvedores de produzirem aplicativos de alta qualidade.

Esse conceito de DT tem sido estendido para se referir a qualquer artefato mal elaborado durante o ciclo de vida [Alves et al. 2016]. 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 [Seaman and Guo 2011] [Kruchten et al. 2012] [Alves et al. 2016].

Para se ter um parâmetro da gravidade do problema, 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 [Tom et al. 2013].

1.1. Declaração do problema

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 conhecimento [Tom et al. 2012] [Tom et al. 2013] [Li et al. 2015] [Ampatzoglou et al. 2015] [Alves et al. 2016] [Fernández-Sánchez et al. 2017].

É importante ressaltar que há opiniões conflitantes sobre tipos de DT. Segundo Li et al. [Li et al. 2015] o conceito DT foi originalmente cunhado para descrever os atalhos técnicos na codificação, a fim de acelerar o desenvolvimento e atender a prazos de liberação urgente. Nos últimos anos, o conceito de DT tem sido estendido a outras fases do ciclo de vida de desenvolvimento de software e cada vez mais conceitos foram colocados sob a égide de DT [Kruchten et al. 2012].

Em seu estudo de mapeamento Li et al. [Li et al. 2015], dedicam um tópico a "Opiniões conflitantes sobre DT", onde apresentam alguns entendimentos divergentes sobre o escopo de DT. Por exemplo, existem autores que argumentam que funcionalidade adiada é DT, enquanto outros explicitamente apontam que recursos e funcionalidades não se referem a DT. Além disto, neste estudo, Li et al. [Li et al. 2015] contabilizam onze estudos que consideram defeitos que tiveram sua correção deliberadamente postergada como DT, enquanto quatro estudos enfatizam explicitamente que defeitos não devem ser incluídos como DT. Apesar de entendermos este conflito de opiniões, neste trabalho adotamos a taxonomia proposta por Alves et al. [Alves et al. 2016], apresentada em detalhes na Seção 2.

Num recente trabalho de Mapeamento Sistemático, Alves et al. [Alves et al. 2016] identificaram diversos tipos de DT, alguns mais fáceis de identificar através de mecanismos automatizados outros altamente dependentes da percepção humana.

Segundo Alves et al. [Alves et al. 2016], pode-se observar uma alta concentração de estudos sobre tipos de dívida mais relacionados ao código-fonte (design, arquitetura, código e defeito). Uma possível explicação para isso é o grande número de ferramentas que executam análise de código-fonte e podem ser usadas para suportar a detecção de DT a partir do código-fonte.

Ainda segundo Alves et al. [Alves et al. 2016], pode-se perceber que, ao longo do tempo, os tipos DT se expandiram e novos foram incluídos, como serviço, processo, usabilidade e controle de versão. Porém, constata-se a ausência de trabalhos relevantes sobre estes tipos de DT.

Apesar de haver um esforço de equipes de desenvolvimento de software em criar

aplicativos cada vez mais fáceis de usar e agradáveis ao usuário final, não há estudos contundentes sobre os impactos da DT de usabilidade no contexto de desenvolvimento de projetos de software [Alves et al. 2016]. O objetivo deste artigo é investigar este fenômeno e entender a natureza da DT de usabilidade e suas implicações.

1.2. Metodologia

A metodologia empregada para realizar a investigação pretendida envolve a condução de um estudo de múltiplos casos a respeito da manifestação de dívida técnica de usabilidade em projetos de software 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 [Runeson 2009]. De fato, o problema tratado neste artigo envolve o estudo de um fenômeno contemporâneo (a manifestação de DT de usabilidade) em seu contexto natural (em projetos de desenvolvimento de software).

Seaman [Seaman 1999], corrobora com a visão de Runeson [Runeson 2009], quando diz que a comunidade que pesquisa o desenvolvimento de sistemas de software tem uma visão pragmática e orientada para resultados sobre metodologia de pesquisa, ao invés de uma posição filosófica.

O problema é a falta de evidências empíricas sobre o gerenciamento de dívida técnica em um ambiente de desenvolvimento de software da vida real [Li et al. 2015].

É importante reunir evidências sobre o DT e o seu comportamento em situações reais de desenvolvimento de software para entender como o gerenciamento da dívida técnica é atualmente percebido por equipes reais de desenvolvimento de software e usar esse conhecimento para aprimorar os processos e ferramentas existentes [Yli-Huumo et al. 2016].

O estudo de casos múltiplos descrito neste artigo é exploratório e descritivo e tem caráter quantitativo [Runeson 2009] [Runeson 2012]. Foi conduzido em cinco projetos de quatro empresas públicas brasileiras. Cada estudo de caso envolveu o acesso autorizado às bases de dados corporativas e reuniões com as equipes dos projetos, fazendo uso da triangulação de fontes de dados. Foram seguidas as diretrizes para conduzir e relatar estudos de caso propostas por Runeson et al. [Runeson 2009] [Runeson 2012].

1.3. Organização do artigo

O restante deste artigo está organizado da seguinte forma: A Seção 2 apresenta a fundamentação teórica a respeito dos tipos de DT em projetos de software e motiva a investigação da DT de usabilidade. A Seção 3 descreve o estudo de caso, apresenta os projetos selecionados, as questões de pesquisa e os procedimentos para coleta de dados. A Seção 4 apresenta a análise dos dados e os resultados obtidos. A Seção 5 contém as considerações finais, com o resumo das constatações, ameaças à validade, seus impactos e implicações e trabalhos futuros.

2. Dívida Técnica em Projetos de Software

Esta seção apresenta uma fundamentação teórica a respeito de DT para permitir o melhor entendimento do trabalho, incluindo os conceitos básicos e a taxonomia de tipos de DT

que utilizaremos como referência neste trabalho.

DT é reconhecida como um problema crítico no setor de desenvolvimento de software: a conta de dívida técnica global foi estimada pela Gartner em US$ 500 bilhões em 2010 [Tom et al. 2013].

A partir da publicação desta estimativa verificou-se um aumento do volume de pesquisas, fato relatado por Tom et al.[Tom et al. 2013] e confirmado por Li et al. [Li et al. 2015] afirmam que "na última década intensificou-se o interesse pelo metáfora da dívida técnica entre os profissionais e a academia".

Diversas pesquisas nessa área tem investigado propostas de solução envolvendo ferramentas e práticas para uma efetiva identificação e gerenciamento da DT [Alves et al. 2016]. No entanto, de acordo com um recente mapeamento, há uma falta de evidências empíricas sobre o gerenciamento da DT em ambientes de desenvolvimento de software reais [Li et al. 2015]. 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 [Yli-Huumo et al. 2016].

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 estudo 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 [Alves et al. 2016]. Dentre estes trabalhos, destacam-se cronologicamente:

- Em 2012 Tom et al.[Tom et al. 2012] 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. [Tom et al. 2013] ampliaram o trabalho de 2012.
- Em 2015 Li et al.[Li et al. 2015] publicaram um novo MS ampliando o mapeamento de Tom et al. [Tom et al. 2012] que visa identificar atividades a serem realizadas na gestão da DT.
- 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 [Alves et al. 2016]. 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 1).

Além destes, outros MS foram desenvolvidos nos últimos anos, com focos bem específicos, dentre eles: o de Fernández-Sánchez et al. [Fernández-Sánchez et al. 2017], interessados especificamente nos elementos necessários para gerenciar a DT e Ampatzoglou et al. [Ampatzoglou et al. 2015] 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 Alves et al. [Alves et al. 2016]:

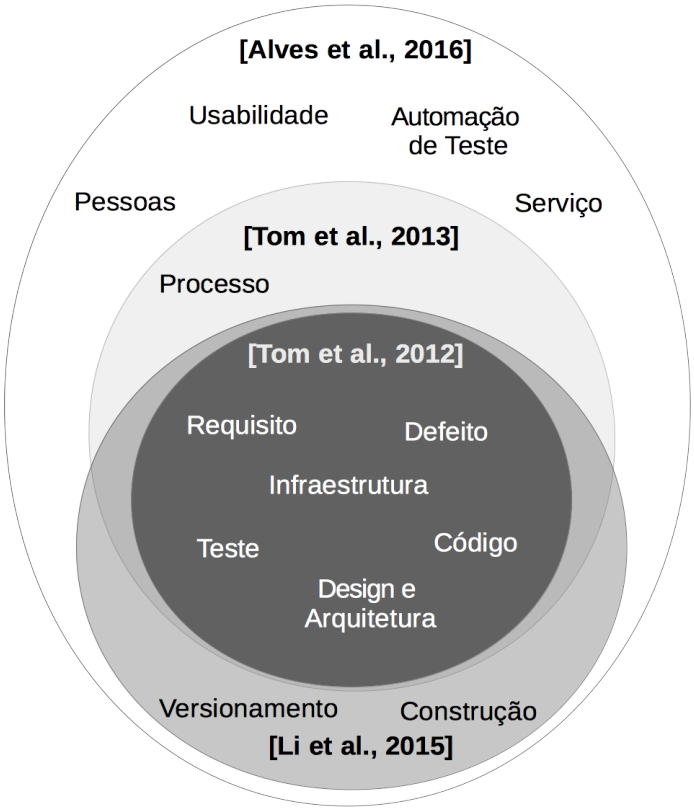

Figura 1. Interseção entre os estudos para a questão de pesquisa sobre tipos de DT. Adaptado de Alves et al. [Alves et al. 2016]

- **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 acompanhamento de *bugs*, que a equipe de desenvolvimento concorda que devem ser reparados, mas devido a prioridades concorrentes e recursos limitados, tem que ser adiado para um momento posterior. Tais decisões podem dificultar a correção posterior". O ponto é, a presença de defeitos e a decisão deliberada de adiar a sua reparação pode fazer o defeito ser mais difícil de ser mantida no futuro e, como consequência, o próprio software.
- **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 deste artigo, diversos autores reconhecem sua importância na prática. Zazworka et al. [Zazworka et al. 2013], 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 artigo o primeiro a explorar o fenômeno da dívida técnica de usabilidade na prática.

3. Estudo de caso

Nesta seção, o estudo de caso é descrito. Seguindo as diretrizes propostas por Runeson [Runeson 2012], 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.

3.1. Questões de pesquisa

Este trabalho se propõe a responder às seguintes questões de pesquisa:

RQ1: Itens de dívida técnica de usabilidade são frequentes em projetos de desenvolvimento de software?

Esta questão tem o propósito de mostrar a relação entre a quantidade total de itens de DT apuradas e a quantidade de itens de DT de usabilidade em cada projeto. Esta relação estabelece a frequência de itens de DT de usabilidade nos projetos.

RQ2: Quais tipos de dívida técnica de usabilidade costumam ocorrer em projetos de desenvolvimento de software?

Após serem apurados todos os itens de DT de usabilidade, estes serão classificados com base nas Heurísticas de Usabilidade de Nielsen [Nielsen 1994]. Esta classificação pode ajudar a determinar as possíveis origens da existência deste tipo de DT nos projetos.

RQ3: Qual o esforço para a resolução da dívida técnica de usabilidade em projetos de desenvolvimento de software?

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 [Runeson 2012].

Tendo isto em vista, conduzimos um estudo de múltiplos casos. Foram selecionadas quatro empresas públicas, 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 indícios de DT, registrados com data de cadastramento anterior a 01/01/2017 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 e servidor de aplicação JBoss EAP. A metodologia de desenvolvimento empregada foi baseada no RUP.

Há aproximadamente 29 profissionais de diversas especialidades envolvidos nas equipes dos projetos A e B, destes, doze 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 x atividade desenvolvida, neste contexto, alguns profissionais exercem mais de uma atividade nos projetos, atualmente sete 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. Este sistema permite registrar, controlar e acompanhar todas as demandas encaminhadas ao projeto, mantendo os históricos das evoluções. O primeiro autor teve acesso autorizado ao RedMine e sua 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 fortes indícios de DT 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 e OpenProject.

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 e banco de dados MySQL, utilizando a metodologia ágil Scrum 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.

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 evolutivas, 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.

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. 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 equipe da Coordenação de Engenharia de Software (CES), responsável pelo desenvolvimento de novos sistemas e manutenção dos existentes, possui 12 servidores

que atuam em mais de 50 projetos, entre novos e manutenção de legados.

O sistema de gestão de tarefas utilizado é o RedMine. 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 a base de dados SQL Server 2008 e servidor de aplicação IIS. A metodologia utilizada para desenvolvimento de sistemas foi desenvolvida pela própria empresa denominada MDS a partir dos conceitos do RUP.

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. 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, 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 Alves et al. [Alves et al. 2016]. 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 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 a 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 que estejam com status "em aberto" no momento da seleção, ou seja encontram-se pendentes de resolução.

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 1).

Nos projetos C e E, a seleção, preliminar, ficou a cargo dos seus respectivos gerentes de projeto, pois o primeiro autor não foi autorizado a acessar diretamente os 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, preliminarmente 255 itens que se enquadravam nos conceitos de DT.

Tabela 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

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 preliminarmente. Fazem uma nova rodada de avaliação e executam 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 observando as características dos itens no contexto de DT e na taxonomia adotada.

No Projeto-A, essa fase foi 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 sequencia 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 (problema de viabilidade técnica e ou financeira);
- itens fora do escopo do projeto (obsoletos).

Em relação aos itens suspensos, apesar de serem considerados DT, estes fogem de um dos filtros da pesquisa que é status em aberto. Os itens incluídos neste critério de exclusão tiveram seus status modificados para "suspensos"e foram excluídos da pesquisa. De fato, não havia interesse por parte das empresas em analisar estes itens. É importante destacar que este critério afetou todos os tipos de dívida técnica da mesma forma. Assim, a exclusão não implica em viés que prejudique os resultados da pesquisa.

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.

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 considerada finalizada.

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.

A proposta inicial era que o gerente do projeto, amparado em sua expertise, valorasse as estimativas em homem-hora de forma direta.

Durante as discussões, surgiu a ideia de dividir a execução da tarefa em atividades menores e valorar em homem-hora para estas atividades. Esta ideia visou obter valores finais com maior precisão de estimativa. Algo similar já era rotina para a equipe técnica em outros projetos de estimativa de esforço.

Foram adotadas 6 atividades utilizadas no pagamento do item de DT. São atividades genéricas da maioria dos ciclo de vida de manutenção dos sistemas, ou seja, cada atividade representam ações a serem desenvolvidas na realização do item, são elas:

- 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.

4. Análise dos dados coletados e resultados obtidos

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 2 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 Alves et al. [Alves et al. 2016].

4.1. Análise dos tipos de Dívida Técnica

Foram selecionados 145 itens de DT em 5 projetos pesquisados (Tabela 2).

A quantidade de itens selecionados, por tipo de DT, conforme taxonomia adotada, ficou assim distribuída: Arquitetura (8), Código (28), Defeito (23), Design (8), Documentação (8), Infraestrutura (11), Processo (3), Requisitos (34), Serviço (1), Teste (1), e Usabilidade (20).

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).

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.

Tabela 2. 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%**

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.

4.1.1. RQ1: Dívidas Técnicas de usabilidade são frequentes em projetos de desenvolvimento de software?

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 2).

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.

4.2. 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 [Molich 1990] [Nielsen 1994]. 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 de qual estado estamos no 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 [Nielsen 1994].

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 3.

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 apenas 1 item (ticket #970 - Projeto D) violou mais de uma heurística conforme apresentado na Tabela 3.

Tabela 3. Heurísticas violadas pelos itens de dívida técnica de usabilidade.

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

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 (33,33%), seguida de Prevenção de erro com 4 itens (19,0%) e Reconhecimento ao invés de lembrar com 3 itens (14,3%) (Tabela 4).

Tabela 4. Totalização os itens de DT de usabilidade - Heurísticas de Nielsen.

Heurística de Nielsen	Qtd	%
1-Visibilidade Status Sistema	1	4,8%
2-Correspondência entre o sistema e o mundo real	1	4,8%
3-Liberdade de controle fácil para o usuário	2	9,5%
5-Prevenção de Erro	4	19,0%
6-Reconhecimento ao invés de lembrar	3	14,3%
7-Flexibilidade e eficiência de uso	7	33,3%
8-Design estético e minimalista	1	4,8%
9-Ajude o usuário a reconhecer, diagnosticar e recuperar erros	2	9,5%
Totalização	**21**	**100,0%**

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

tendência:

- **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.
- **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 ao invés de lembrar:** 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.

4.2.1. RQ2: Quais tipos de Dívida Técnica de usabilidade costumam ocorrer em projetos de desenvolvimento de software?

Os 20 itens de dívida técnica representam a violação de 8 das 10 heurísticas de usabilidade (Tabela 3). Percebe-se que não há concentração em uma determinada heurística, porém 3 heurísticas concentram aproximadamente 67% dos itens selecionados:

- **7-Flexibilidade e eficiência de uso:** Foi apontada em 33,33% 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 [Nielsen 1994].
  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.
- **5-Prevenção de erro:** Apontada em 19% 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 [Nielsen 1994].
  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.
- **6-Reconhecimento ao invés de lembrar:** Apontada em 14,3% 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 [Nielsen 1994].
As demandas selecionadas nesta categoria tinham em comum uma dificuldadedo usuário no entendimento de uma das ações de um dos sistemas.

Talvez a melhor resposta para esta questão seja: "Depende do contexto". Como as Heurísticas de Nielsen são abrangentes e cada projeto tem seu contexto único, fica difícil definir quais os tipos de dívida técnica de usabilidade que ocorrem em projetos de desenvolvimento de software.

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 maiseficiente.

4.3. 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 5).

Tabela 5. Análise do esforço homem-hora.

	Tipos de DT	Apuração do Custo Hora/Homem										Acumulado	
		Projeto-A		Projeto-B		Projeto C		Projeto-D		Projeto-E			
		H/H	%	H/H	%	H/H	%	H/H	%	H/H	%	H/H	%
Grupo-1	Arquitetura	1.680	58,8%					108	17,7%			1.788	28,59%
	Código	198	6,9%			382	13,7%	74	12,1%			654	10,46%
	Defeito	32	1,1%			373	13,4%	175	28,7%			580	9,27%
	Design							27	4,4%			27	0,43%
	Subtotal Grupo-1											3.049	48,75%
Grupo-2	Documentação	292	10,2%					20	3,3%			312	4,99%
	Infraestrutura					885	31,8%					885	14,15%
	Processo					884	31,7%					884	14,13%
	Requisitos	464	16,2%			90	3,2%	170	27,9%			724	11,58%
	Teste							5	0,8%			5	0,08%
	Usabilidade	191	6,7%			173	6,2%	31	5,1%			395	6,32%
	Subtotal Grupo-2											3.205	51,25%
Total		**2.857**	**100,0%**			**2.787**	**100,0%**	**610**	**100,0%**			**6.254**	100,00%

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 6. 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 6), 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.

4.3.1. RQ3: Qual o esforço para a resolução da Dívida Técnica de usabilidade em projetos de desenvolvimento de software?

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 4.3.

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 5).

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 5).

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 5).

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 6).

Como descrito na seção 4.3, 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 6).

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.

5. Considerações finais

O objetivo deste artigo 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.

5.1. 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 [Nielsen 1994]. 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.

5.2. Implicações

Os resultados deste trabalho de pesquisa 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.

5.3. 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 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.

5.4. Trabalhos futuros

Segundo Alves et al. [Alves et al. 2016] 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 [Alves et al. 2016]. 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

Alves, N. S. R., Mendes, T. S., de Mendonça, M. G., Spínola, R. O., Shull, F., and Seaman, C. (2016). Identification and management of technical debt: A systematic mapping study. 70:100–121.

Ampatzoglou, A., Ampatzoglou, A., Chatzigeorgiou, A., and Avgeriou, P. (2015). The financial aspect of managing technical debt: A systematic literature review. 64:52–73.

Cunningham, W. (1992). The WyCash portfolio management system. In *Addendum to the Proceedings on Object-oriented Programming Systems, Languages, and Applications (Addendum)*, OOPSLA '92, pages 29–30. ACM.

Falessi, D., Kruchten, P., and Avgeriou, P. (2016). Introduction to the special issue on technical debt in software systems. 120:154–155.

Fernández-Sánchez, C., Garbajosa, J., Yagüe, A., and Perez, J. (2017). Identification and analysis of the elements required to manage technical debt by means of a systematic mapping study. 124:22–38.

Guo, Y., Seaman, C., and Q.B. da Silva, F. (2016). Costs and obstacles encountered in technical debt management – a case study. 120:156–169.

Kruchten, P., Nord, R., and Ozkaya, I. (2012). Technical debt: From metaphor to theory and practice. 29(6):18–21.

Lehman, M. M. (1996). Feedback in the software evolution process. 38(11):681–686.

Li, Z., Avgeriou, P., and Liang, P. (2015). A systematic mapping study on technical debt and its management. 101:193–220.

Molich, Rolf; Nielsen, J. (1990). Improving a human-computer dialogue. *Commun. ACM*, 33(3):338–348.

Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 152–158. ACM.

Parnas, D. (1994). Software aging. In *IEEE Proceedings of 16th International Conference on Software Engineering*, pages 279–287, Sorrento, Italy, Italy.

Rios, N., Mendonça, M., and Spínola, R. (2018). A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. *Information and Software Technology*, 102:117–145.

Runeson, P. Host, M. (2009). Empir software eng (2009) 14: 131. https://doi.org/10.1007/s10664-008-9102-8.

Runeson, Per; Host, M. R. A. R. B. (2012). *Case Study Research in Software Engineering - Guidelines and Examples*. John Wiley Sons Ltd.

Seaman, C. and Guo, Y. (2011). Chapter 2 - measuring and monitoring technical debt. In Zelkowitz, M. V., editor, *Advances in Computers*, volume 82, pages 25–46. Elsevier.

Seaman, C. B. (1999). Qualitative methods in empirical studies of software engineering. *IEEE Transactions on Software Engineering*, 25:557–572.

Tom, E., Aurum, A., and Vidgen, R. (2012). A consolidated understanding of technical debt.

Tom, E., Aurum, A., and Vidgen, R. (2013). An exploration of technical debt. 86(6):1498–1516.

Yli-Huumo, J., Maglyas, A., and Smolander, K. (2016). How do software development teams manage technical debt? – an empirical study. 120:195–218.

Zazworka, N., Spínola, R. O., Vetro', A., Shull, F., and Seaman, C. (2013). A case study on effectively identifying technical debt. In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, EASE '13, pages 42–47. ACM.