Avaliando o Tratamento de Exceção em um Sistema Web Corporativo

Um Estudo de Caso

Dêmora Bruna Cunha de Sousa
Universidade Federal do Ceará (UFC)
Fortaleza, Ceará, Brasil
de.brunasousa@gmail.com

Windson Viana de Carvalho
Universidade Federal do Ceará (UFC)
Fortaleza, Ceará, Brasil
windson@great.ufc.br

Lincoln Souza Rocha
Universidade Federal do Ceará (UFC)
Fortaleza, Ceará, Brasil
lincoln@dc.ufc.br

ABSTRACT

The exception handling has necessary elements for the development of a fault-tolerant Web system. Although, even with its high importance, studies indicate a neglect and low quality of the exception handling development. This research aims to conduct an in-depth analysis of exception handling, exploring human and technical relevant aspects obtained from a corporate environment Web system. We have initiated the first unit of analysis. It regards the developers' perception of exception handling on an organisation. Among the obtained results, it is possible to realise that the importance of the exceptional flows is recognised, and developers suggest its normalisation would bring benefits to the developed systems. During the case study is intended to assess the quality of the exception handling on the target systems comparing to the developers' background, moreover, institutionalise rules to the implemented exception handling.

KEYWORDS

exception handling, Web system, case study

1 CARACTERIZAÇÃO DO PROBLEMA

O tratamento de exceção fornece elementos necessários para a construção de sistemas tolerantes a falha [8]. Apesar de sua importância, estudos indicam a negligência e o desenvolvimento de baixa qualidade do tratamento de exceção, mesmo em sistemas corporativos [2, 20]. Por ser inerente ao desenvolvimento de um sistema, problemas no tratamento de exceção afetam sistemas de diversos domínios e plataformas. Isso não é diferente no domínio de sistemas Web [11].

Ao longo dos anos, pesquisas foram realizadas buscando entender as perspectivas dos desenvolvedores e das organizações com relação à temática. Shah et al. [20] abordam os divergentes pontos de vista sobre o tratamento de exceção relatados por novos desenvolvedores e por desenvolvedores experientes. Ebert e Castor [5] conduzem um estudo acerca das percepções de desenvolvedores sobre *bugs* do tratamento de exceção. Outros estudos sugerem novos métodos e técnicas para facilitar a condução do tratamento de exceção. Huang et al. [10] apresentam um *framework* de tratamento de exceção para a composição de serviços Web. Filho [7] desenvolve uma solução para verificação da conformidade arquitetural do tratamento de exceção. Sawadpong e Allen [17] propõem um conjunto de métricas, obtidas através da análise estática de um gráfico de chamadas do tratamento de exceção, para auxiliar na predição de defeitos de exceção.

Além da percepção dos usuários, outro fator pode contribuir com o decaimento da qualidade dos fluxos excepcionais de um sistema é a erosão arquitetural de software. Ela ocorre quando a arquitetura planejada para o sistema torna-se diferente durante o processo de evolução do software [12, 21]. Dentre as causas da erosão, pode-se citar o aumento da complexidade do sistema, a falta de clareza para descrever decisões arquiteturais, o aumento dos custos de produção e o acúmulo de decisões de *design*, pois implica na não correspondência do sistema aos requisitos [21]. O tratamento de exceção, participante do *design* arquitetural de sistemas, torna-se propenso aos efeitos da erosão arquitetural de software, como também pode estar associado a uma de suas consequências, o aumento da suscetibilidade de defeitos.

Este trabalho de mestrado pretende examinar de forma profunda o tratamento de exceção em um sistema Web. Para alcançar esse objetivo, pretende-se analisar como os profissionais responsáveis pelo desenvolvimento do sistema, tais como desenvolvedores e analistas, compreendem e utilizam o tratamento de exceção, considerando suas visões técnicas e pessoais, além de compreender a influência dos processos estabelecidos pela instituição. Outro aspecto importante é determinar a qualidade do tratamento de exceção dos sistemas desenvolvidos, basando-se em recomendações estabelecidas na literatura e boas práticas do tratamento de exceção. Por fim, a pesquisa de mestrado irá definir e implantar normas para o tratamento de exceção para o sistema analisado por meio de técnicas e metodologias presentes na literatura. Nesse contexto, o trabalho de mestrado vislumbra responder três questões de pesquisa:

- **QP1:** Qual é a percepção dos desenvolvedores e analistas de um sistema Web sobre o tratamento de exceção?
- **QP2:** Como a visão sobre o tratamento de exceção desses profissionais impacta nos sistemas desenvolvidos?
- **QP3:** Como definir e implantar normas para o tratamento de exceção de um sistema já em uso de modo a evitar/diminuir a ocorrência de erros derivados de uma má qualidade do tratamento de exceção?

As respostas para as questões de pesquisa serão buscadas durante um estudo de caso focado no tratamento de exceção aplicado em um sistema Web corporativo, utilizado há 10 anos e com mais 20 mil usuários ativos. O propósito é perceber similaridades e divergências entre os estudos já realizados sobre o tratamento de exceção e a pesquisa a ser desenvolvida, além de buscar a melhoria na qualidade

In:XVII Workshop de Teses de Dissertações (WTD 2017), Gramado, Brasil. Anais do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web: Workshops e Pôsteres. Porto Alegre: Sociedade Brasileira de Computação, 2017.

ISBN 978-85-7669-380-2.

do sistema existente por meio da normatização de regras, de boas práticas de programação do tratamento de exceção e da adoção de um suporte ferramental adequado.

Um estudo de caso é um método científico destinado a investigar fenômenos no seu ambiente de origem, sendo caracterizado por sua flexibilidade para lidar com a dinamicidade existente em ambientes reais, além de se basear em uma clara cadeia de evidências e proporcionar fortes e relevantes conclusões [23]. Tais características são necessárias para alcançar os objetivos apresentados nesta pesquisa e contribuem para a condução de uma investigação profunda do tratamento de exceção de um sistema. O detalhamento do *design* de pesquisa e de todos os procedimentos adotados durante o estudo de caso tornará possível a sua reprodução em outras entidades que lidam com o desenvolvimento de sistemas, especialmente aquelas especializadas em sistemas Web corporativos. Tornando possível qualificar o tratamento de exceção implementado e determinar normas vislumbrando o seu bom funcionamento nessas instituições.

O restante deste documento está organizado em seis seções que estão descritas a seguir. A Seção 2 trata da fundamentação teórica e dos trabalhos relacionados, abordando o tratamento de exceção e os problemas relacionados a ele, além de uma revisão da bibliografia dos trabalhos correlatos a este trabalho de mestrado. A proposta desta pesquisa é apresentada na Seção 3. A Seção 4 aborda o estado atual do trabalho. A Seção 5 trata das dos resultados obtidos até o momento. Por fim, a Seção 6 apresenta as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

2.1 Tratamento de Exceção

Durante a execução de um sistema, nem sempre é possível alcançar um comportamento satisfatório. Às vezes, a chamada de uma rotina deixa o sistema em um estado anormal ou mesmo viola o procedimento determinado na especificação, o que é caracterizado como uma falha.

Uma exceção é um evento anormal em tempo de execução que interrompe a atividade de um sistema durante a chamada de um procedimento, podendo resultar em uma falha [13]. Tais eventos devem ser tratados para deixar o sistema em um estado no qual seja possível prosseguir.

Mesmo que não seja de maneira padrão, as linguagens de programação de alto nível apresentam um sistema de tratamento de exceção. Por meio desse sistema ocorre a comunicação entre o procedimento que detecta a exceção, sinalizador da exceção, e entidade que requisitou a operação, além de permitir a definição de tratadores para exceções específicas [4]. Segundo Dony [4], para sinalizar uma exceção é necessário: identificar a situação causadora da exceção; interromper a sequência de execução; procurar um tratamento para exceção; e passar as informações necessárias para que o tratamento ocorra.

Tratadores de exceção (*handlers*) são parte do comportamento excepcional do sistema, sendo responsáveis por alternar entre o fluxo de controle normal e o fluxo de controle excepcional [9]. Normalmente, componentes tolerantes a falha apresentam essa divisão bem definida.

Garcia et al. [9] ilustram o funcionamento das chamadas regiões protegidas ou contexto de tratamento, que constituem a parte do código cujo os tratadores de exceção estão anexados. Quando uma exceção é lançada em uma região protegida, o fluxo normal é alternado para o fluxo excepcional. Uma região protegida pode ter mais de um tratador, sendo ele escolhido pelo sistema de tratamento de exceção de acordo com a exceção lançada.

2.2 Problemas Relacionados ao Tratamento de Exceção

2.2.1 Bugs do Tratamento de Exceção. Barbosa et al. [1] adotam o termo falhas excepcionais ou falhas do tratamento de exceção para referenciar aquelas relacionadas à definição, lançamento, propagação e documentação de exceções. Ebert et al. [6] se referem ao mesmo problema, mas empregam o termo *bugs* do tratamento de exceção, que por definição são *bugs* que ocorrem quando uma exceção é definida, lançada, propagada, tratada ou documentada; na ação de limpeza de uma região protegida onde é lançada; quando ela deveria ter sido lançada ou tratada, mas não foi.

Ebert et al. [6] categorizam *bugs* para o tratamento de exceção e suas causas baseando-se em informações extraídas de repositórios de *bug* e na opinião de desenvolvedores. Segundo os dados apresentados no estudo, dentre as categorias criadas, as exceções não tratadas e as não levantadas são as maiores geradoras de erro no contexto dos projetos analisados.

2.2.2 Erosão do Tratamento de Exceção. Perry e Wolf [15] caracterizam, de forma geral, partes de um produto de software: requisitos, arquitetura, projeto e implementação. Dentre elas, a arquitetura preocupa-se com o conjunto de elementos arquiteturais, as interações existentes entre eles e as regras para esses elementos e interações. As entidades que constituem a arquitetura são definidas para satisfazer os requisitos e utilizadas como base para as próximas etapas inerentes a construção de um software. Os autores também apresentam um modelo de arquitetura que é composto por três elementos arquiteturais, sendo eles: elementos, organização e decisões. A arquitetura é de suma importância para reger o bom funcionamento de um software, além de servir como guia para a adição de novas funcionalidades ou durante a alteração das existentes. No entanto, nem sempre os conceitos definidos pela arquitetura são obedecidos. Van Gurp e Bosch [21] destacam alguns fatores que facilitam a ocorrência dessas violações:

- **Rastreabilidade das decisões de projeto:** As notações utilizadas para descrever as decisões arquiteturais não são expressivas o suficiente para um entendimento posterior;
- **Aumento do custo de produção:** Devido à evolução do sistema, que produz um aumento de sua complexidade, a realização de tarefas torna-se custosa, o que facilita a tomada de decisões que ferem a integridade arquitetural;
- **Acúmulo de decisões de projeto:** O acúmulo de decisões de *design* implica na não correspondência do sistema aos requisitos;
- **Metodologias iterativas:** Metodologias iterativas incorporam modificações que impactam na arquitetura durante o desenvolvimento, no entanto, tais decisões deveriam ser integradas durante a fase de projeto.

O acúmulo de violações causa um fenômeno conhecido como erosão arquitetural. Esse fenômeno ocorre quando a arquitetura implementada de um software, definida como o modelo da arquitetura construída a partir de elementos de baixo nível ou código-fonte, diverge da arquitetura planejada [3].

O tratamento de exceção faz parte do *design* arquitetural do sistema. Por isso, também está sujeito a erosão de software caso o acúmulo de violações referentes a sua especificação aconteça. A verificação de conformidade é uma solução para evitar esse problema, pois permite a identificação do código violador, facilitando o providenciamento de uma correção. Dentre as técnicas utilizadas para o tratamento de erosão, a checagem de relações de dependência torna possível identificar relações que deveriam, ou não, existir entre exceções e os módulos de um sistema. Sendo possível derivar regras para definir essas relações. Essas normas podem ser usadas por arquitetos e desenvolvedores para expressar o comportamento excepcional de um sistema. Como também servem de parâmetro para ferramentas que verificam a conformidade arquitetural do tratamento de exceção [7].

2.3 Trabalhos Relacionados

Nesta Seção, são apresentados os trabalhos correlatos a esta pesquisa. Esses trabalhos foram selecionados utilizando a estratégia de pesquisa sistemática *snowballing* [22]. O conjunto inicial de artigos, primeiro passo exigido pela estratégia, foram obtidos a partir de palavras-chave que remetem ao tratamento de exceção (*handling exception*) e estudos de empíricos (*case study*, *empirical study*, *exploratory study*). Necessariamente, os trabalhos selecionados deveriam considerar aspectos de ambientes reais de desenvolvimento, sendo estes os desenvolvedores ou o próprio sistema desenvolvido. Essa determinação foi usada como critério de exclusão. Além disso, foram considerados somente artigos publicados a partir de 2012.

Um total de 14 trabalhos foram obtidos durante a primeira iteração do processo de revisão da literatura, que ainda está em progresso. Alguns desses estudos de casos são brevemente descritos nas próximas subseções.

2.3.1 Estudos centrados no desenvolvedor. Ebert e Castor [5] abordam a percepção de desenvolvedores e organizações sobre *bugs* no tratamento de exceção. A coleta de dados ocorre mediante questionários online respondidos por 154 desenvolvedores e pesquisadores. A partir das respostas fornecidas, os autores apresentam algumas conclusões: os códigos relativos ao tratamento de exceção não são testados e documentados com frequência, profissionais mais experientes tendem a ser mais críticos com a qualidade do código destinado ao comportamento excepcional, desenvolvedores usam o tratamento de exceção para criar mecanismos de tolerância a falhas e para melhorar a qualidade do código desenvolvido.

Shah et al. [20] realizaram dois estudos para captar as visões de desenvolvedores novos e experientes sobre o tratamento de exceção. Os dados são coletados por meio de entrevistas semiestruturadas com os participantes do estudo. De acordo com os resultados alcançados, desenvolvedores menos experientes tentam evitar o tratamento de exceção ou somente imitar práticas existentes no código. Já desenvolvedores experientes veem o tratamento como parte integrante e inseparável do desenvolvimento de software, também o usam para fornecer melhores *feedbacks* sobre o erro corrido, evitar a corrupção de dados e controlar o fluxo de execução.

2.3.2 Estudos centrados no software. Sana et al. [19] conduzem um estudo para entender as abordagens empregadas no tratamento de exceção em bibliotecas Java. As bibliotecas selecionadas tiveram seus fluxos de exceção analisados e relacionados com *bugs* documentados em sistemas de reportação. Os autores destacam entre suas descobertas que a maioria das bibliotecas averiguadas não documentam *Runtime Exceptions*, que *anti-patterns* do tratamento de exceção foram detectados em 25% dos fluxos analisados e que mais de 20% dos problemas relatados eram relacionados ao tratamento de exceção.

Sawadpong et al. [18] tem como principal objetivo verificar se o tratamento de exceção pode ser considerado arriscado. Para isso, é analisada a densidade de defeitos em classes do código da IDE (*Integrated Development Environment*) Eclipse e também um conjunto de métricas sobre o tratamento de exceção extraídas do seu código-fonte. Os resultados extraídos revelam um declínio na quantidade de defeitos durante a evolução do software. No entanto, os defeitos associados ao tratamento de exceção continuaram crescendo ao longo dos 6 anos considerados na pesquisa. O contrário acontece com os defeitos não relacionados ao tratamento de exceção.

Osman et al. [14] apresentam um estudo sobre a evolução do tratamento de exceção em sistemas que possuem, pelo menos, 5 anos de existência, constituindo um total de 90 projetos Java. Os autores buscaram identificar as diferenças entre a evolução do tratamento de exceção em aplicações e bibliotecas, como também entre projetos de diferentes domínios.

Diferentemente dos trabalhos previamente relatados, esta pesquisa pretende utilizar os aspectos humanos e técnicos pertinentes ao tratamento de exceção, obtidos em um mesmo ambiente de desenvolvimento, com a finalidade de realizar uma análise profunda e ampla sobre o tema. Assim como abordar o tratamento de exceção em nível arquitetural, realizando a verificação da conformidade arquitetural no sistema alvo e elaborar normas para a implementação adequada do tratamento de exceção.

3 PROPOSTA E CONTRIBUIÇÕES ESPERADAS

3.1 Proposta

Esta pesquisa tem como proposta analisar o tratamento de exceção em um ambiente real de desenvolvimento de sistemas Web, considerando aspectos humanos e técnicos. Espera-se obter uma variedade rica de informações que possibilitem entender a percepção que os integrantes de uma equipe de desenvolvimento possuem sobre o tratamento de exceção e as consequências dessa visão nos sistemas desenvolvidos. Além disso, deseja-se buscar métodos ou técnicas capazes de auxiliar no estabelecimento de regras do tratamento de exceção, como também definir e implantar a normatização produzida.

Planeja-se conduzir este estudo nos moldes da engenharia de software experimental, mediante a condução de um estudo de caso. A aplicação do estudo de caso é possível devido a multidisciplinaridade da engenharia de software, que implica no desenvolvimento de diversas atividades, como também envolve uma variada gama de

profissionais. A riqueza desse meio favorece a execução de experimentos, já que cada ambiente possui fatores próprios que impactam nos resultados obtidos durante a execução de uma atividade [23].

O sistema selecionado como alvo deste estudo é Web, possui mais de 20 mil usuários ativos, e foi desenvolvido em J2EE (*Java Enterprise Edition*). Adquirido há 10 anos, atende a diversos segmentos da organização que o mantém. Devido sua grande importância, e por fins de segurança e confiabilidade, assegurados pelos condutores desta pesquisa, seu nome permanecerá anônimo, sendo referenciado no decorrer deste documento como Sistema S. A instituição que mantém e evolui esse sistema tem um setor composto por 42 profissionais que atuam direta ou indiretamente na codificação e análise desse sistema, sendo esta uma instituição pública.

3.2 Contribuições Esperadas

Com a proposta apresentada, pretende-se alcançar as seguintes contribuições:

- Obter informações quantitativas e qualitativas relevantes para a elaboração de processos, ferramentas e técnicas cuja aplicação seja direcionada a resolução de problemas do tratamento de exceção em instituições atuantes na produção e manutenção de sistemas Web com características similares (i.e., instituições públicas ou corporações públicas com setores de desenvolvimento de sistemas de médio a grande porte);
- Contribuir com as pesquisas na área de erosão arquitetural de software e *bugs* no tratamento de exceção oferecendo percepções obtidas em um ambiente real de desenvolvimento;
- Aplicar métodos e técnicas já propostos pela academia para a medição da qualidade de tratamento de exceção, para a definição de regras do tratamento de exceção e para a checagem da conformidade arquitetural, utilizando de forma prática as contribuições de pesquisadores da área de engenharia de software;
- Auxiliar a instituição mantenedora do Sistema S na melhoria do tratamento de exceção de seus sistemas, visto que a mesma ainda não implementa métricas de qualidade para tal, assim como não possui regras bem definidas para o tratamento de exceção.

4 ESTADO ATUAL DO TRABALHO

Tratando-se de um estudo de caso, um dos elementos chave para o prosseguimento deste trabalho de mestrado é o *design* da pesquisa. O projeto deve ser planejado para tratar dos estágios essenciais de um estudo de caso, como coleta de dados, análise e relatórios, assim como da preparação para realização dessas atividades, que podem constituir treinamentos e reuniões na corporação de aplicação [16].

A elaboração do protocolo está em processo finalização, sendo construído em torno das questões de pesquisa previamente apresentadas neste documento.

4.1 O Caso e o Seu Contexto

O estudo de caso está sendo conduzido com a intenção de medir a qualidade do tratamento de exceção em uma corporação de desenvolvimento de sistemas Web e, a partir das informações obtidas, produzir e implantar um conjunto de normas para a utilização do tratamento de exceção nos sistemas da organização. Uma premissa que valida a execução desse estudo é o fato da corporação selecionada não apresentar uma documentação extensa sobre o tratamento de exceção, o que pode gerar dúvidas durante as atividades de desenvolvimento e, consequentemente, interferir na capacidade do sistema se recuperar de faltas. No entanto, para afirmar a ocorrência dessa situação é preciso perceber como os desenvolvedores Web entendem o tratamento de exceção dos sistemas que desenvolvem e mensurar o estado atual desse tratamento. Essas informações serão buscadas durante a execução das unidades de análise que compõem o estudo de caso.

4.2 Unidades de Análise

As unidades de análise foram definidas de acordo com as questões de pesquisa estabelecidas, elas são também ilustradas na Figura 1. A primeira unidade é destinada ao aspecto humano do tratamento de exceção, que são as pessoas que lidam diretamente com o planejamento e desenvolvimento dos fluxos excepcionais do sistema. Durante a execução dessa unidade deseja-se verificar a percepção dos desenvolvedores/analistas a respeito do:

- Grau de importância dado ao tratamento de exceção;
- Estado atual do sistema Web estudado com relação ao tratamento de exceção;
- Conhecimento adquirido sobre o tratamento de exceção empregado nos sistemas desenvolvidos;
- Contentamento sobre suas próprias habilidades e conhecimento do tratamento de exceção.

A segunda unidade de análise é destinada ao aspecto técnico do tratamento de exceção, que diz respeito ao Sistema S e ao ferramental que oferece suporte ao monitoramento e normatização tratamento de exceção. Nessa unidade, um conjunto de ferramentas será utilizado para determinar a qualidade do tratamento de exceção. A seleção dessas ferramentas será feita por meio de uma pesquisa bibliográfica a fim de utilizar contribuições acadêmicas existentes.

A terceira unidade é dedicada à normatização do tratamento de exceção, à divulgação e à checagem periódica dessas normas. A normatização será obtida na própria organização, entendendo os requisitos desejáveis para o tratamento de exceção, e na literatura, através do atendimento às boas práticas, às recomendações comprovadamente estabelecidas, metodologias e técnicas para a definição de regras. Pretende-se englobar o tratamento de exceção internamente nas entidades do software (classes, componentes, etc), como também nos relacionamentos entre os componentes do sistema, criando uma regulamentação arquitetural. A divulgação dessas normas se efetuará via documentos e reuniões com a equipe de desenvolvimento, mas somente após a sua aprovação pelos líderes de equipe. Posteriormente, o cumprimento das normas será mensurado por intermédio da verificação periódica da qualidade do tratamento de exceção e da conformidade com a normatização estabelecida.

4.3 Métodos de Coleta

Vários métodos de coleta estão sendo utilizados neste estudo devido à natureza diversa da origem dos dados (pessoas, código e documentação). Alguns métodos já foram selecionados, exibidos na Figura 1.

No entanto, mudanças podem ocorrer em virtude da flexibilidade necessária ao estudo de caso.

Figure 1: Estrutura do estudo de caso.

4.4 Coleta de Dados: Unidade de Análise 1

Atualmente, a primeira unidade de análise está em andamento. O primeiro método de coleta, formulário online, foi elaborado para obter a percepção dos desenvolvedores e analistas sobre o tratamento de exceção. Os trabalhos de Ebert et al. [6] e Shah et al. [20] foram utilizados como base para a construção do formulário, assim como alguns aspectos inerentes ao desenvolvimento de software e ao tratamento de exceção, sendo eles:

- Experiência no desenvolvimento de sistemas Web;
- Experiência com os sistemas desenvolvidos na corporação em que o profissional atua;
- Documentação;
- Importância do tratamento de exceção;
- Conhecimento técnico sobre o tratamento de exceção;
- *Bugs* do tratamento de exceção;
- Utilização do tratamento de exceção durante as atividades desenvolvidas;
- Satisfação com o próprio conhecimento e com o tratamento de exceção praticado na corporação;

O formulário online foi avaliado por três especialistas em qualidade e arquitetura de software Web. Após o término das avaliações, foi divulgado por uma mensagem enviada para a lista de *e-mails* dos profissionais da corporação. Nessa mensagem, o propósito da pesquisa foi apresentado, como também garantida a anonimidade das informações enviadas pelo formulário e o uso exclusivo para fins acadêmicos.

5 RESULTADOS OBTIDOS

5.1 Unidade de Análise 1: Percepção dos Desenvolvedores e Analistas

O formulário online recebeu 17 respostas até o momento da escrita deste documento (40,5% dos desenvolvedores). Essa pequena quantidade era esperada devido ao pouco tempo em que está disponibilizado. Apesar disso, algumas considerações já puderam ser realizadas sobre as respostas e são destacadas nas subseções a seguir.

5.1.1 Experiência. A maioria dos respondentes trabalha há menos de 6 anos com o desenvolvimento Web, cerca de 52,9%. O restante está nesse ramo há mais de 7 anos. Grande parte trabalha na instituição em que atua há apenas 3 anos. Consideram-se experientes quando se trata dos sistemas desenvolvidos, mas declaram ainda não possuir domínio completo do ferramental e da linguagem de programação utilizados na corporação.

5.1.2 Importância do Tratamento de Exceção. Como pode ser visto na Figura 2, a maioria dos respondentes considera o tratamento de exceção crucial para o desenvolvimento de sistemas Web. Também concordam com a obrigação imposta por algumas linguagens para que o tratamento de exceção seja realizado.

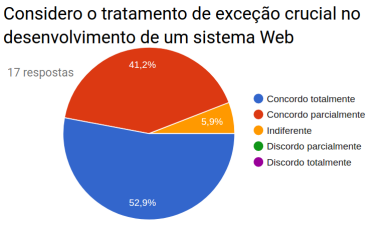

Figure 2: Importância do tratamento de exceção em sistemas Web.

5.1.3 Documentação. Cerca de 29,4% dos respondentes revelam que costumam não documentar elementos de código relativos ao tratamento de exceção. E quase a totalidade dos desenvolvedores concordam que a instituição não possui especificações ou padrões para o tratamento de exceção de seus sistemas (Figura 3).

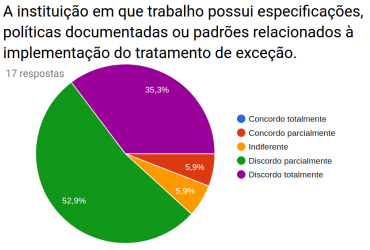

Figure 3: Documentação do tratamento de exceção nos sistemas desenvolvidos.

5.1.4 Bugs do Tratamento de Exceção. O impacto de erros relativos ao tratamento de exceção nos usuários finais do sistema é confirmado por 52,9% dos candidatos. Afirmam também que erros do tratamento de exceção já foram reportados à equipe de desenvolvimento. Esses erros podem ser consequência da pouca adesão dos desenvolvedores, apenas 17,6%, ao uso de ferramentas de manutenção da qualidade do tratamento de exceção (Figura 4).

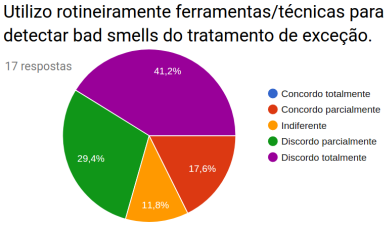

Figure 4: Utilização de ferramentas para auxiliar no desenvolvimento do tratamento de exceção.

5.1.5 Satisfação. 100% dos respondentes acreditam que o tratamento de exceção dos sistemas desenvolvidos precisa ser melhorado, mesmo que parcialmente (Figura 5). Situação semelhante também é vista com relação ao estabelecimento de normas institucionais para o tratamento de exceção. Por fim, as respostas indicam um não contentamento dos profissionais com a forma que lidam com o tratamento de exceção na condução de suas atividades de desenvolvimento e análise.

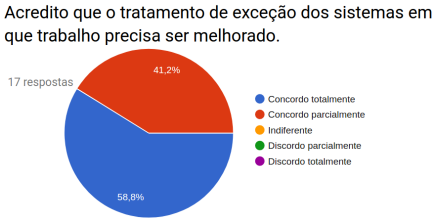

Figure 5: Opinião dos respondentes sobre o tratamento de exceção no sistemas desenvolvidos.

6 CONCLUSÕES

Este trabalho de mestrado apresenta como proposta um estudo de caso sobre o tratamento de exceção em um ambiente de desenvolvimento Web corporativo. A importância da temática foi abordada durante a explanação dos problemas relacionados ao tratamento de exceção e aos trabalhos correlatos a esta pesquisa. Detalhes sobre a condução do estudo de caso foram destacados, como também o início da primeira unidade de análise, que trata da percepção dos desenvolvedores sobre o tratamento de exceção.

A partir das respostas adquiridas no questionário online, destaca-se que os desenvolvedores reconhecem que o tratamento de exceção é crucial para sistemas Web e deve ser melhorado nos sistemas desenvolvidos por eles. Outras informações necessárias a este trabalho serão obtidas com o término da unidade de análise 1 e durante a execução das unidades 2 e 3.

Com a condução desse estudo não espera-se obter dados estatisticamente relevantes, já que é um estudo de caso e abrange um número pequeno de desenvolvedores em um contexto específico. Mas pretende-se fornecer um caso de natureza qualitativa sobre do tratamento de exceção e sua formalização em um sistema Web real.

REFERENCES

[1] Eiji Adachi Barbosa, Alessandro Garcia, and Simone Diniz Junqueira Barbosa. 2014. Categorizing faults in exception handling: A study of open source projects. In *Software Engineering (SBES), 2014 Brazilian Symposium on*. IEEE, 11–20.

[2] Bruno Cabral and Paulo Marques. 2007. Exception handling: A field study in Java and. Net. In *European Conference on Object-Oriented Programming*. Springer, 151–175.

[3] Lakshitha De Silva and Dharini Balasubramaniam. 2012. Controlling software architecture erosion: A survey. *Journal of Systems and Software* 85, 1 (2012), 132–151.

[4] Christophe Dony. 1990. Exception handling and object-oriented programming: towards a synthesis. *ACM Sigplan Notices* 25, 10 (1990), 322–330.

[5] Felipe Ebert and Fernando Castor. 2013. A study on developers' perceptions about exception handling bugs. In *Software Maintenance (ICSM), 2013 29th IEEE International Conference on*. IEEE, 448–451.

[6] Felipe Ebert, Fernando Castor, and Alexander Serebrenik. 2015. An exploratory study on exception handling bugs in Java programs. *Journal of Systems and Software* 106 (2015), 82–101.

[7] Juarez Filho. 2016. ArCatch: Uma Solução para Verificação Estática de Conformidade Arquitetural do Tratamento de Exceção. (2016).

[8] Alessandro F Garcia, Cecília MF Rubira, Alexander Romanovsky, and Jie Xu. 2001. A comparative study of exception handling mechanisms for building dependable object-oriented software. *Journal of systems and software* 59, 2 (2001), 197–222.

[9] Alessandro F Garcia, Cecília M.F Rubira, Alexander Romanovsky, and Jie Xu. 2001. A comparative study of exception handling mechanisms for building dependable object-oriented software. *Journal of Systems and Software* 59, 2 (2001), 197 – 222. https://doi.org/10.1016/S0164-1212(01)00062-0

[10] Jian Huang, Wei Zhu, Farokh Bastani, I-Ling Yen, and Jicheng Fu. 2012. Automated exception handling in service composition using holistic planning. In *Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on*. IEEE, 251–258.

[11] Junguo Li, Gang Huang, Jian Zou, and Hong Mei. 2007. Failure analysis of open source J2EE application servers. In *Quality Software, 2007. QSIC'07. Seventh International Conference On*. IEEE, 198–208.

[12] Mikael Lindvall and Dirk Muthig. 2008. Bridging the software architecture gap. *Computer* 41, 6 (2008).

[13] Bertrand Meyer. 1988. *Object-oriented software construction*. Vol. 2. Prentice hall New York. 411–438 pages.

[14] Haidar Osman, Andrei Chiş, Claudio Corrodi, Mohammad Ghafari, and Oscar Nierstrasz. 2017. Exception evolution in long-lived Java systems. In *Proceedings of the 14th International Conference on Mining Software Repositories*. IEEE Press, 302–311.

[15] Dewayne E Perry and Alexander L Wolf. 1992. Foundations for the study of software architecture. *ACM SIGSOFT Software Engineering Notes* 17, 4 (1992), 40–52.

[16] Per Runeson, Martin Host, Austen Rainer, and Bjorn Regnell. 2012. *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons.

[17] Puntitra Sawadpong and Edward B Allen. 2016. Software Defect Prediction Using Exception Handling Call Graphs: A Case Study. In *High Assurance Systems Engineering (HASE), 2016 IEEE 17th International Symposium on*. IEEE, 55–62.

[18] Puntitra Sawadpong, Edward B Allen, and Byron J Williams. 2012. Exception handling defects: An empirical study. In *High-Assurance Systems Engineering (HASE), 2012 IEEE 14th International Symposium on*. IEEE, 90–97.

[19] Demóstenes Sena, Roberta Coelho, Uirá Kulesza, and Rodrigo Bonifácio. 2016. Understanding the exception handling strategies of Java libraries: An empirical study. In *Mining Software Repositories (MSR), 2016 IEEE/ACM 13th Working Conference on*. IEEE, 212–222.

[20] Hina Shah, Carsten Gorg, and Mary Jean Harrold. 2010. Understanding exception handling: Viewpoints of novices and experts. *IEEE Transactions on Software Engineering* 36, 2 (2010), 150–161.

[21] Jilles Van Gurp and Jan Bosch. 2002. Design erosion: problems and causes. *Journal of systems and software* 61, 2 (2002), 105–119.

[22] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. ACM, 38.

[23] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media. 55–72 pages.