



SARAH FIGUEIREDO CAIXETA LEITE

**INSPEÇÃO DE USABILIDADE APLICADA A
MÉTODOS ÁGEIS:
UM ESTUDO DE CASO**

LAVRAS – MG

2013

SARAH FIGUEIREDO CAIXETA LEITE

**INSPEÇÃO DE USABILIDADE APLICADA A MÉTODOS ÁGEIS:
UM ESTUDO DE CASO**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Sistemas de Informação para obtenção do título de Bacharel em Sistemas de Informação.

Orientador

Prof. DSc. André Pimenta Freire

LAVRAS – MG

2013

SARAH FIGUEIREDO CAIXETA LEITE

**INSPEÇÃO DE USABILIDADE APLICADA A MÉTODOS ÁGEIS:
UM ESTUDO DE CASO**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Sistemas de Informação para obtenção do título de Bacharel em Sistemas de Informação.

APROVADA em 22 de Agosto de 2013.

Prof. DSc. André Luiz Zambalde

Universidade Federal de Lavras

MSc. Gabriel Chaves Afonso Coutinho

Mitah Technologies



Prof. DSc. André Pimenta Freire
(Orientador)

LAVRAS – MG

2013

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus, que me deu a oportunidade de viver e concluir esta fase da minha vida.

Agradeço aos meus pais, Marcos e Monica, pelo apoio incondicional, pela compreensão, pelas conversas e sucos de laranja.

Agradeço aos amigos que me acompanharam nesta jornada, Verônica, Ana Paula, Giancarlo, Guilherme, Fred, Gustavo Monteiro, Gustavo Vale, Álvaro, João Marcos, Samara, André e Bárbara, ao quase irmão Julio e o grande amigo Netto, sem os quais tudo isto seria em vão.

Aos amigos da Mitah Technologies, Mariana, Zanzini, Leandro, Tiago, Gabriel, Nilson, Ricardo, dentre outros, que me acompanharam também no aprendizado profissional, eu agradeço por tudo.

Agradeço ao meu orientador André Pimenta por toda a atenção durante o projeto, e também à Juliana Gregghi, que me prestou grande apoio. Aos amigos Mariana e Urlan, agradeço pelos conselhos na execução do projeto.

À todos que de alguma forma participaram deste momento, meus agradecimentos.

RESUMO

Empresas que utilizam métodos ágeis de desenvolvimento têm grande preocupação com a qualidade do produto entregue. A usabilidade é um requisito importante da qualidade do software. No entanto, muitas empresas têm dificuldade para integrar processos de usabilidade às práticas ágeis utilizadas. Poucos estudos já foram feitos com este tema, e eles mostram que ainda há muito trabalho a ser feito para chegar à integração adequada dos processos ágeis e de usabilidade. Este trabalho propõe o estudo da utilização da avaliação heurística de um sistema Web no contexto de uma empresa de pequeno porte de desenvolvimento de software que utiliza o método ágil *Scrum*. Para isto foi realizado um estudo com a aplicação na empresa de duas versões da avaliação heurística: a avaliação tradicional e a colaborativa, que foram utilizados por duas equipes da empresa com níveis equivalentes de experiência com usabilidade. A avaliação heurística colaborativa mostrou tendência a encontrar problemas de usabilidade mais sérios e também teve aceitação pelos participantes do estudo, apesar de não ter encontrado um número maior de problemas do que a avaliação heurística tradicional. Conclui-se que a abordagem colaborativa da avaliação heurística é adequada para inspeção de usabilidade em ambientes ágeis de desenvolvimento, podendo trazer ganhos de produtividade, bem como de compartilhamento de conhecimento entre membros da equipe de desenvolvimento e melhor alinhamento para integração em processos com ciclos iterativos.

Palavras-Chave: Avaliação Heurística, Usabilidade, Métodos Ágeis, Interação Humano-Computador, Engenharia de Software.

ABSTRACT

Companies using agile development methods are concerned about delivering quality software products. Usability is an important software quality requirement. However, a number of companies have difficulties integrating usability processes with agile practices. Few studies have been conducted about this topic. The studies conducted so far have suggested that there is a significant amount of work to be done in order to achieve the appropriate integration between agile and usability processes. The present work proposed the study of the use of heuristic evaluation of a Web site in the context of a small software development company that uses the Scrum agile method. In order to accomplish this goal, a case study was designed and conducted to compare the use of two versions of the heuristic evaluation method: the standard and the collaborative version, which were performed by two different teams of evaluators from the company with equivalent usability expertise levels. The results showed that the collaborative evaluation tended to find more serious usability problems than the standard version of the method. It also had good acceptance from the participants, despite the fact that it did not find more usability problems than the standard evaluation. It was concluded that the collaborative version of the heuristic evaluation is suitable for usability inspection in agile development environments. It enabled achieving productivity gains, as well as knowledge sharing between members of the development team and better alignment for integration in processes with iterative cycles.

Keywords: Heuristic Evaluation, Usability, Agile Methods, Human-Computer Interaction, Software Engineering.

SUMÁRIO

1	Introdução	11
1.1	Objetivos	13
1.2	Organização da Monografia	14
2	Referencial Teórico	15
2.1	Métodos Ágeis	15
2.1.1	Princípios do Desenvolvimento Ágil	16
2.1.2	Scrum	18
2.2	Avaliação de Usabilidade	20
2.2.1	Usabilidade	20
2.2.2	Inspeção de Usabilidade	22
2.2.2.1	Avaliação Heurística	22
2.2.2.2	Avaliação Heurística Colaborativa	25
2.2.2.3	Percurso Cognitivo	27
2.2.2.4	Inspeção Semiótica	28
2.2.3	Heurísticas para sistemas Web de Petrie e Power (2012)	29
2.3	Avaliação de Usabilidade em Métodos Ágeis	29
3	Métodos	33
3.1	Tipo de Pesquisa	33
3.2	Procedimentos Metodológicos	33
3.2.1	Desenho	33
3.2.2	Participantes	35
3.2.3	Materiais	37
3.2.4	Procedimento	38
3.2.4.1	Treinamento	38
3.2.4.2	Avaliação Tradicional	39

3.2.4.3	Avaliação Colaborativa	39
3.2.5	Análise de Dados	40
4	Resultados e Discussão	42
4.1	Violações das Heurísticas nas Avaliações	42
4.1.1	Avaliação Heurística Tradicional	42
4.1.2	Avaliação Heurística Colaborativa	49
4.1.3	Comparação entre as violações das heurísticas nas duas versões de avaliação heurística	50
4.2	Número de Problemas Encontrados	51
4.3	Severidades dos problemas encontrados	52
4.3.1	Avaliação Heurística Tradicional	52
4.3.2	Avaliação Heurística Colaborativa	53
4.3.3	Comparação entre as severidades dos problemas encontrados nas duas versões do método	54
4.4	Análise do Processo de Avaliação	55
4.5	Discussão	58
4.5.1	Número de problemas encontrados nas duas versões do método . . .	58
4.5.2	Severidade dos problemas encontrados exclusivamente por um ou outro método	60
4.5.3	Produtividade da avaliação heurística tradicional e colaborativa . . .	60
4.5.4	Dificuldades para a atribuição de graus de severidade	61
4.5.5	Utilização das heurísticas nas avaliações	61
4.5.6	Adequação do método de avaliação heurística colaborativa para mé- todos de desenvolvimento ágil	62
5	Conclusões	64
A	Heurísticas utilizadas para as avaliações	71

B Planilhas utilizadas nas avaliações	73
B.1 Planilha de identificação de problemas	73
B.2 Planilha das severidades dos problemas	74
C Questionários Utilizados	75
C.1 Questionário demográfico	75
C.2 Questionário da avaliação tradicional	76
C.3 Questionário da avaliação colaborativa	77

LISTA DE TABELAS

2.1	As Dez Heurísticas de Nielsen	23
4.1	Número de violações de heurísticas na avaliação realizada pelo AT1 .	44
4.2	Número de violações de heurísticas na avaliação realizada pelo AT2 .	46
4.3	Número de violações de heurísticas na avaliação realizada pelo AT3 .	47
4.4	Número de violações por grupo de heurísticas após a sessão de consoli- dação	48
4.6	Número de violações por grupo de heurísticas na avaliação colaborativa	50

LISTA DE FIGURAS

2.1	Exemplo de gráfico <i>Burndown</i>	20
2.2	Proporção dos problemas de usabilidade encontrados em uma interface em relação ao número de avaliadores. Fonte: Nielsen (2000) .	25
4.1	Problemas encontrados nas duas abordagens	51
4.2	Severidade dos problemas encontrados na avaliação tradicional	52
4.3	Severidade dos problemas encontrados na avaliação colaborativa	53

1 INTRODUÇÃO

Atualmente muitas organizações têm optado pela adoção de processos e práticas ágeis de desenvolvimento. A escolha de uma empresa por métodos ágeis vem principalmente pela sua capacidade de responder mais facilmente às mudanças de requisitos durante o desenvolvimento do software (SOARES, 2004). Esta competência de resposta agrega valor ao produto, uma vez que os processos de uma empresa não são imutáveis, e o cliente espera que o sistema que será entregue atenda às suas reais necessidades, não as necessidades de quando o contrato foi assinado.

Alguns motivos do crescimento da adoção de práticas ágeis podem ser vistos em pesquisa realizada por Ambler (2008) sobre a adoção de desenvolvimento ágil de software. Nesta pesquisa é afirmado que, em comparação com equipes tradicionais, em 22% dos casos a produtividade de equipes ágeis é maior, em 29% dos casos a qualidade é melhor e em 31% dos casos os *stakeholders* estão mais satisfeitos.

Segundo Crespo *et al.* (2004), atualmente existe uma demanda por software de qualidade. Em um ambiente onde os requisitos mudam com grande frequência, organizações sofrem grande pressão para atender a esta demanda em curtos espaços de tempo. Neste contexto, as empresas tendem a optar por processos ágeis de desenvolvimento.

Para Nerur, Mahapatra e Mangalaraj (2005) na adoção de métodos ágeis a organização tem que repensar seus objetivos e reorganizar os componentes humanos, gerenciais e tecnológicos.

Para garantir a qualidade do produto desenvolvido, a avaliação de usabilidade é muito importante. A importância da usabilidade é relatada na ISO/IEC (2005). No entanto Seffah e Metzker (2004) ressaltam que técnicas de *design* centrado no usuário ainda são pouco conhecidas, pouco usadas e consideradas difíceis de dominar pelos desenvolvedores de software.

Objetivos de usabilidade do sistema incluem desempenho e satisfação do usuário em determinados contextos de uso (ISO, 2002). Um erro muitas vezes cometido é pensar que usabilidade é apenas a decoração do sistema, que fica por cima do sistema real. Este pensamento leva desenvolvedores a deixar requisitos de usabilidade para o final do projeto, o que acaba comprometendo o resultado final. A preocupação com usabilidade deve estar presente em todas as fases do desenvolvimento do software (SEFFAH; METZKER, 2004).

Seffah e Metzker (2004) apontam que estudos empíricos sobre engenharia de software mostram que, na maioria dos casos, mesmo quando uma empresa desenvolve boas práticas de integração entre Interface Humano-Computador e Engenharia de Software, estas práticas não são publicadas. As empresas não costumam ter a preocupação de documentar estas práticas, nem mesmo internamente.

Winter *et al.* (2011), Barbosa, Furtado e Gomes (2008), Nielsen e Madsen (2012) dentre outros, realizaram estudos sobre a integração entre processos de usabilidade e processos ágeis. Estudos mostram que as empresas ainda têm dificuldade em integrar estes processos. Nielsen e Madsen (2012) relatam que os profissionais estão empenhados para fazer os processos ágeis e de usabilidade funcionarem em conjunto.

Desta forma, mostrou-se relevante o estudo da aplicação da avaliação heurística no ambiente de uma empresa de desenvolvimento de software.

Avaliação heurística é um método de análise de usabilidade onde um número de avaliadores é apresentado a uma interface para que comentem sobre a mesma considerando um conjunto de heurísticas (NIELSEN; MOLICH, 1990). Avaliação heurística é um método de inspeção muito conhecido e aceito por especialistas, como mostra Buykx (2009). Em sua dissertação, Buykx (2009) apresenta a avaliação heurística colaborativa, que é uma variação da avaliação heurística com foco no trabalho em grupo para diminuição da duplicação e aumento da produtividade.

Poucos autores relataram experiências com a avaliação heurística colaborativa. Dentre eles, pode-se citar Buykx (2009) e Babajo (2012). O conhecimento sobre a aplicação da avaliação heurística colaborativa em pequenas empresas é limitado. Alguns estudos mostram a utilização de métodos de inspeção de usabilidade em ambientes ágeis de desenvolvimento. Estes estudos mostraram que em ambientes onde são utilizados métodos ágeis ainda há muitos problemas a serem resolvidos quanto à integração com práticas de usabilidade.

São raros os estudos de caso mostrando como a avaliação heurística é aplicada em empresas que utilizam metodologias de desenvolvimento ágeis. Desta forma, a proposta deste trabalho foi efetuar um estudo de caso comparando a utilização de métodos de inspeção baseados em avaliação heurística, nas suas versões tradicional e colaborativa, em uma empresa de pequeno porte que utiliza um método ágil.

1.1 Objetivos

O objetivo geral deste trabalho é analisar a utilização da avaliação heurística tradicional e a avaliação heurística colaborativa em uma pequena empresa de desenvolvimento de software que utiliza o método ágil *Scrum*.

Entre os objetivos específicos deste trabalho estão:

- Observar a quantidade de problemas encontrados na avaliação heurística tradicional e na avaliação heurística colaborativa.
- Analisar os graus de severidade obtidos na avaliação heurística colaborativa.
- Analisar o tempo gasto para a consolidação de resultados na avaliação heurística tradicional.
- Investigar prós e contras da utilização da avaliação heurística tradicional e a avaliação heurística colaborativa no desenvolvimento ágil.

1.2 Organização da Monografia

No Capítulo 2 é conceituado o tema do trabalho, apresentando métodos ágeis, com ênfase em *Scrum*, avaliação de usabilidade, técnicas de inspeção e avaliação de usabilidade em métodos ágeis.

O Capítulo 3 apresenta o tipo de pesquisa realizada e os procedimentos metodológicos, incluindo o desenho da pesquisa e detalhes do procedimento.

O Capítulo 4 apresenta os resultados do trabalho, e no Capítulo 5 são relatadas as conclusões da monografia.

2 REFERENCIAL TEÓRICO

Neste são apresentados conceitos fundamentais para o entendimento do trabalho desenvolvido.

Na Seção 2.1 são apresentados conceitos de métodos ágeis, com enfoque no *Scrum*, método utilizado na empresa onde o estudo de caso foi executado. A Seção 2.2 apresenta conceitos de avaliação de usabilidade, métodos de inspeção de usabilidade e uma descrição aprofundada do método avaliação heurística, em suas versões: tradicional e colaborativa. A Seção 2.3 mostra como questões de usabilidade têm sido tratadas nas empresas atualmente e apresenta estudos da utilização de processos de usabilidade no contexto de métodos ágeis.

2.1 Métodos Ágeis

Nesta seção é abordada a filosofia e as práticas dos métodos ágeis. O foco desta seção é o método *Scrum*, que foi utilizado no estudo de caso descrito.

Durante o desenvolvimento de um software podem ocorrer mudanças nos requisitos. A equipe tem que estar preparada para lidar com mudanças. Desta forma, o processo de desenvolvimento não pode estar engessado, amarrado a um contrato que não pode ser alterado. É necessário que o método de desenvolvimento seja flexível para receber bem as alterações de requisitos durante o desenvolvimento (FOWLER, 2005).

Segundo Fowler (2005) o movimento ágil teve início em 2001, quando dezessete pessoas que tinham experiência com desenvolvimento de software por alguns anos se reuniram devido ao seu interesse em uma nova abordagem sobre a forma de desenvolver software. Eles concordavam que os métodos tradicionais de desenvolvimento de software não eram tão produtivos, e tinham ideias de como melhorar esta situação.

No *workshop* organizado em 2001, os dezessete participantes escreveram o Manifesto Ágil (BECK *et al.*, 2001), que descreve clara e sucintamente os ideais do movimento ágil. As premissas descritas no manifesto são:

- Indivíduos e interações mais que processos e ferramentas.
- Software em funcionamento mais que documentação abrangente.
- Colaboração com o cliente mais que negociação de contratos.
- Responder a mudanças mais que seguir um plano.

2.1.1 Princípios do Desenvolvimento Ágil

Shore e Warden (2008), descrevem cinco princípios do *Extreme Programming*, que são válidos também para outros métodos ágeis, são eles:

- Coragem: para tomar as decisões certas e dizer a verdade aos stakeholders quando necessário.
- Comunicação: para dar às pessoas informações corretas, para serem usadas em sua vantagem.
- Simplicidade: para descartar coisas que queremos, mas não precisamos.
- *Feedback*: para tirar lições em toda oportunidade que tivermos.
- Respeito: tratar uns aos outros com dignidade, reconhecendo suas competências.

É importante entender o projeto, e como o processo de desenvolvimento o afeta. Para isso, deve-se usar o *feedback*, para entender o que funciona e o que não funciona. A integração do time é crucial para isto, quando alguém aprender algo, deve passar para os outros, assim como quando tiver dúvidas, deve procurar alguém com conhecimento para ajudá-lo.

Os indivíduos envolvidos nos processos são muito valorizados em ambientes ágeis. De acordo com Taniguchi & Correa (TANIGUCHI; CORREA, 2009), em métodos ágeis é importante que as pessoas sejam competentes e motivadas, para isto elas participam nas decisões e gestão de tarefas. Para produzir bons resultados, é necessário que os envolvidos trabalhem juntos. Ambientes ágeis incentivam comunicação e bom relacionamento entre os membros do time (SHORE; WARDEN, 2008).

Na prática, as reuniões diárias são o principal momento onde as informações são trocadas, alguns ficam sabendo o que os outros estão fazendo, as dificuldades que estão tendo e, assim, podem ajudar e aprender. Estas reuniões também são uma forma de incentivar o relacionamento entre a equipe.

De acordo com Taniguchi e Correa (2009), um dos valores principais no *Scrum* é a característica de se preocupar com a gestão do projeto e não com processos e controles. Assim com o *Scrum* é possível descobrir pontos falhos nos processos e como corrigi-los.

Quando se trabalha com métodos ágeis, não se deve ter medo de mudanças: se uma alteração é necessária, deve ser feita. O ambiente e os requisitos sofrem mudanças, assim sendo, processo tem que se adaptar (HIGHSMITH, 2002). Ocasionalmente uma mudança pode tornar o processo pior, mas o mais importante é que o time de desenvolvimento seja flexível. O time tem a coragem de experimentar, mesmo que em algum momento falhe (SHORE; WARDEN, 2008).

É necessário encontrar um balanceamento entre planejamento e adaptação (HIGHSMITH, 2002). Num ambiente com muito planejamento e pouca adaptação a empresa não responde ao ambiente externo, ao passo que na situação inversa há muita oscilação sem resultado. Este balanceamento é uma das maiores barreiras para empresas que começam a usar métodos ágeis.

O Manifesto Ágil (BECK *et al.*, 2001) tem dois princípios que dizem respeito à previsibilidade e adaptabilidade. Estes princípios dizem que mudanças

nos requisitos podem ser feitas, mesmo que num estado avançado do desenvolvimento. Processos ágeis usam das mudanças para obter vantagem competitiva. As melhores arquiteturas, requisitos e designs vêm de times de desenvolvimento auto-organizados.

Um dos objetivos dos métodos ágeis é eliminar perdas, e para isto é importante dividir o trabalho em pequenas tarefas. Dividindo o trabalho em partes menores, a execução de cada parte é mais rápida, e reduz as possibilidades de erros para as alterações mais recentes (HIGHSMITH, 2002). Tarefas pequenas são de mais fácil compreensão, desta forma o time não fará atividades que estejam fora do escopo, que são tarefas desnecessárias.

Para eliminar perdas, também importante saber quando desistir, ou mudar o escopo do projeto para algo que possa ser realizado. É necessário estar sempre a par do andamento do projeto, para que medidas corretivas, se necessárias, sejam tomadas o mais rápido possível, diminuindo ao máximo as perdas.

Os métodos ágeis mais conhecidos são: *Scrum* (HIGHSMITH, 2002), *Extreme Programming* (BECK, 2000), *Crystal Methods* (COCKBURN, 2004), *Lean Development* (POPPENDIECK; POPPENDIECK, 2003), *Dynamic Systems Development* (STAPLETON, 1997), *Feature-Driven Development* (PALMER; FELSING, 2001) e *Adaptive Software Development* (HIGHSMITH III, 2000).

2.1.2 Scrum

Highsmith (2002) diz que no desenvolvimento de software não há como definir um plano exato do que será entregue, quando, e quanto vai custar, no entanto, é possível definir um processo para monitoramento e gerenciamento, observando de perto o processo de desenvolvimento. A ideia do *Scrum* é definir um processo onde seja valorizada a comunicação, colaboração e compartilhamento de conhecimento entre os membros da equipe.

Algumas das práticas do *Scrum* são *Backlog* e *Sprint* (NASCIMENTO, 2007). *Sprints* são iterações do projeto, que duram por volta de trinta dias, e *Backlog* consiste em uma lista das atividades a ser realizadas durante o projeto. No início de um *Sprint* é feita uma reunião de planejamento, na qual são selecionados determinados itens do *Backlog* que serão executados no período.

A imprevisibilidade do ambiente de desenvolvimento de software pode dificultar o trabalho da equipe de desenvolvimento. No *Scrum* os requisitos selecionados para um *Sprint* não podem ser alterados no decorrer do mesmo. Os requisitos especificados ao início do período continuam imutáveis até que seja entregue esta parte do produto (HIGHSMITH, 2002). Para uma próxima iteração alterações podem ser feitas.

Nos ambientes de desenvolvimento que utilizam *Scrum* é realizada diariamente uma reunião em que o time relata o que foi feito, o que está fazendo, e quais as dificuldades que está encontrando, desta forma todos ficam cientes do andamento do desenvolvimento e também é incentivado o compartilhamento do conhecimento, à medida que cada membro do time tem a oportunidade de auxiliar e aprender com os outros.

Ao fim de um *Sprint* é realizada uma reunião onde são apresentados os resultados. Nela é analisado o que foi feito no período, como anda o projeto de acordo com o cronograma, e podem ser demonstradas as funcionalidades que foram implementadas (HIGHSMITH, 2002).

O gráfico *Burndown* é um artefato utilizado no *Scrum* para demonstração do andamento do projeto. Ele representa o quanto resta do *Backlog* em relação ao tempo disponível para finalizar o projeto (TANIGUCHI; CORREA, 2009). A Figura 2.1 apresenta um exemplo de gráfico *Burndown*.

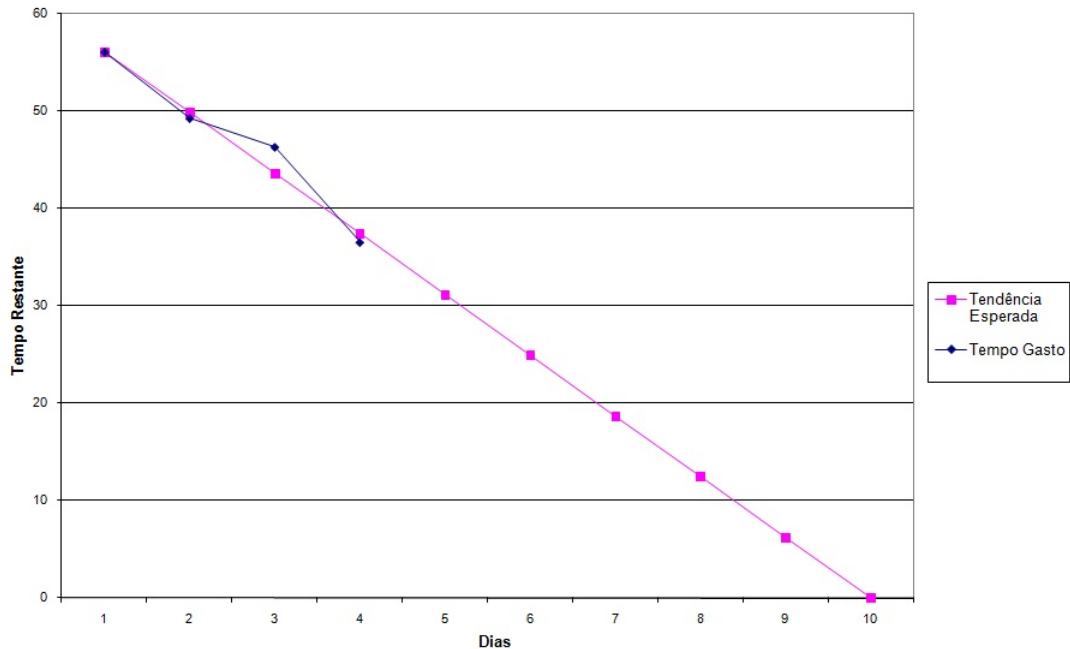


Figura 2.1: Exemplo de gráfico *Burndown*

2.2 Avaliação de Usabilidade

2.2.1 Usabilidade

A ISO 9241 (ISO, 2002) traz definições e orientações importantes para a usabilidade. Ela define usabilidade como uma medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso. Esta norma ISO ainda contém orientações para aquisição, projeto, desenvolvimento, avaliação e comunicação da informação sobre usabilidade.

Outra norma que também possui conteúdos sobre usabilidade é a ISO 25000 (ISO/IEC, 2005). Ela define usabilidade como a capacidade de um usuário alcançar determinados objetivos com efetividade, eficiência e satisfação em

situações especificadas. Segundo a referida norma ISO, usabilidade é constituída, dentre outros, de apreensibilidade, flexibilidade, acessibilidade, reconhecibilidade, facilidade de uso, proteção contra erros do usuário, estética da interface com o usuário e conformidade com convenções e regulamentações relacionadas à usabilidade.

Apreensibilidade é a capacidade de o software ser aprendido pelo usuário. Flexibilidade é a usabilidade e segurança em contextos diferentes dos identificados inicialmente. Acessibilidade é a usabilidade e segurança para usuários com determinadas deficiências. Reconhecibilidade é o grau de informação provida que possibilita ao usuário reconhecer se o software é apropriado para as suas necessidades. Facilidade de uso diz respeito à possibilidade do usuário operar e controlar o produto. Proteção contra erros é o grau no qual o sistema protege o usuário de cometer erros. Estética da interface é o quanto o produto de software é atraente ao usuário. Conformidade quanto à usabilidade é o produto estar de acordo com normas, conversões, guias de estilo ou regulamentações relacionadas à usabilidade. (ISO/IEC, 2005)

Holzinger (2005) menciona algumas outras características que, para ele, devem ser parte de qualquer projeto de software: Eficiência: possibilitar que o usuário tenha alta produtividade; Memorização: possibilitar que usuário casual retorne ao sistema depois de um longo período e não precise reaprender tudo; Baixa taxa de erro: não possibilitar erros catastróficos ocorram, e quando ocorrerem erros, que eles sejam recuperáveis; e Satisfação: fazer o sistema agradável.

Para avaliar a usabilidade de um sistema podem ser utilizados métodos de inspeção ou de observação. Barbosa e Silva (2010) citam como métodos de observação o teste com usuários e a avaliação de comunicabilidade. Já os métodos de inspeção são realizados por especialistas.

Gutwin e Greenberg (2000) descrevem que a realização de testes observacionais é feita através da análise de como as pessoas efetuam determinadas tarefas

em um sistema. De acordo com Buykx (2009), por envolverem usuários do sistema, testes com usuários tendem a exigir mais esforços e ser mais caros (BUYKX, 2009). Apesar do alto custo, Holzinger (2005) afirma que testes com usuários finais são, de certa forma, indispensáveis. Testes com usuários provêem informações diretas sobre como as pessoas utilizam o sistema e seus problemas com determinadas interfaces.

Para Buykx (2009), métodos de inspeção despertam interesse porque podem aprimorar a usabilidade antes de o sistema ser exposto ao teste com usuários.

2.2.2 Inspeção de Usabilidade

Existem vários métodos de inspeção de usabilidade. Alguns conhecidos são: Avaliação Heurística, Percurso Cognitivo e Inspeção Semiótica. Neste trabalho será utilizada a avaliação heurística, portanto os outros métodos serão descritos brevemente.

2.2.2.1 Avaliação Heurística

Segundo Barbosa e Silva (2010), é um método de avaliação de Interação Humano-Computador (IHC) criado para encontrar problemas de usabilidade. Nielsen (1994) explica que na avaliação heurística um conjunto de avaliadores inspeciona a interface com base em um pequeno conjunto de princípios, chamados de "heurísticas". Esta inspeção sistemática da interface busca problemas que prejudiquem a usabilidade, e é considerada uma alternativa mais rápida e de baixo custo comparada a métodos empíricos.

Nielsen (1994) apresenta um conjunto de heurísticas que representam uma síntese de vários outros conjuntos existentes da época do referido estudo. O objetivo de Nielsen ao elaborar esta lista foi reunir as heurísticas de forma a encontrar um conjunto que melhor explique os problemas existentes em sistemas reais. Ele

avaliou 101 heurísticas, que descreviam 249 problemas de usabilidade e chegou à uma lista de dez heurísticas, descritas na Tabela 2.1.

Tabela 2.1: As Dez Heurísticas de Nielsen

Heurística	Descrição
1. Visibilidade do <i>status</i> do sistema	O sistema deve manter o usuário informado sobre o que está acontecendo.
2. Relacionamento entre o sistema e o mundo	Devem ser seguidas conversões do mundo real, as informações devem aparecer em ordem natural e lógica.
3. Controle do usuário e liberdade	Se o usuário realizar uma ação indesejada, ele deve poder desfazê-la.
4. Consistência e padrões	As mesmas ações devem ser nomeadas da mesma forma, devem ser seguidos padrões.
5. Prevenção de erros	Deve-se eliminar situações onde possam ocorrer erros. Se isto não for possível, deve-se ter uma opção de confirmação da ação.
6. Reconhecer ao invés de relembrar	Deixar visíveis objetos, ações e opções. Instruções devem ser visíveis e fáceis de recuperar.
7. Flexibilidade e eficiência de uso	Usuários especialistas devem ter ações facilitadas.
8. Estética e <i>design</i> minimalista	Mostrar somente informações realmente necessárias.
9. Auxílio aos usuários para reconhecer, diagnosticar e recuperar ações erradas	Mensagens de erro devem ser mostradas em linguagem simples e indicar precisamente qual o erro.
10. Ajuda e documentação	O sistema deve oferecer alguma forma de documentação e a busca por informações deve ser facilitada.

Em sua concepção inicial a avaliação heurística determina que cada especialista em usabilidade avalie a interface sozinho. Depois que todos terminarem os avaliadores participam de uma reunião de consolidação de resultados, onde eles compartilham as avaliações (HOLZINGER, 2005). Este formato de avaliação é conhecido como avaliação heurística tradicional.

Durante uma seção de avaliação o avaliador percorre a interface várias vezes. Ele inspeciona vários elementos, comparando-os com princípios de usabilidade reconhecidos. Quando o avaliador encontra um possível problema de usabilidade, ele o registra. Segundo Babajo (2012) a execução individual da avaliação da interface pode contar com a participação de um facilitador, que fica responsável por registrar e reunir os problemas e tratar dos níveis de severidade.

Os princípios de usabilidade devem estar adequados ao sistema avaliado, portanto podem variar na avaliação de sistemas diferentes (HOLZINGER, 2005).

Após a realização das sessões de avaliação, as listas de problemas de cada avaliador são reunidas, com os problemas duplicados identificados e removidos. Babajo (2012) recomenda que a mesma pessoa que desempenhou o papel de facilitador durante as sessões realize este processo.

Depois que os problemas foram reunidos, todos os avaliadores se reúnem para classificação das severidades. Babajo (2012) esclarece que se não for possível que todos os participantes estejam presentes, pelo menos quatro devem estar. Nesta sessão os avaliadores classificam as severidades dos problemas de 1 a 4, onde 1 é considerado problema cosmético e quatro uma catástrofe. A severidade definida para um problema é aquela que todos os avaliadores concordarem.

Vários autores, como Nielsen e Mack (1994) e Holzinger (2005), consideram que o número de avaliadores necessário é entre três e cinco. A Figura 2.2, retirada de Nielsen (2000) mostra a relação entre o número de avaliadores e os problemas encontrados em relação ao total. De acordo com Nielsen (2000), na maioria dos casos a utilização de mais que cinco avaliadores não apresenta bom

custo-benefício. Não é necessário que os avaliadores sejam especialistas, no entanto isto interferirá nos resultados.

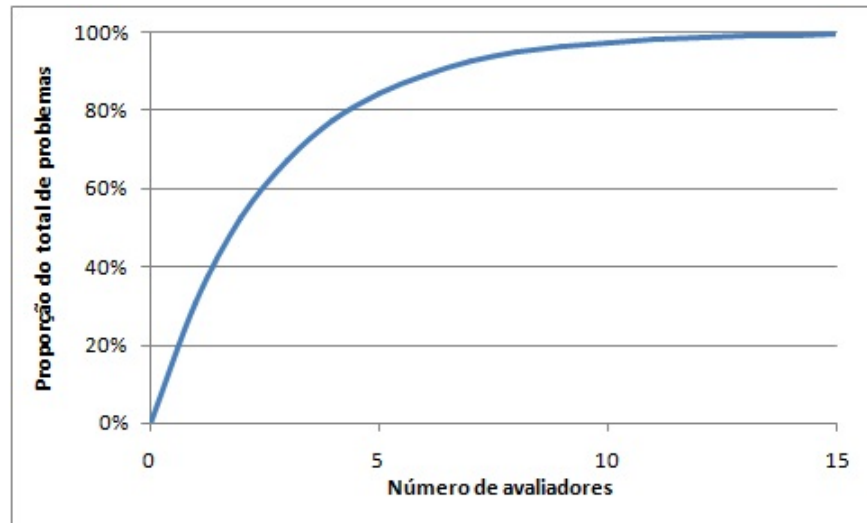


Figura 2.2: Proporção dos problemas de usabilidade encontrados em uma interface em relação ao número de avaliadores. Fonte: Nielsen (2000)

Dentre as vantagens da avaliação heurística estão a utilização de princípios reconhecidos, intuitividade, usabilidade em todo o processo de desenvolvimento e identificação efetiva de problemas. Como desvantagens pode-se citar a separação dos usuários finais e o fato de que não há garantia que todo o *design* seja avaliado (HOLZINGER, 2005).

2.2.2.2 Avaliação Heurística Colaborativa

A avaliação heurística também pode ser realizada de forma colaborativa. Nesta modalidade, os participantes realizam a avaliação como um time (DANINO, 2001).

Babajo (2012) ressalta que quando um grupo de avaliadores trabalha individualmente no mesmo sistema, há uma tendência de encontrar problemas duplicados quando uma comparação é feita. Ele cita que a realização de avaliação

em conjunto pode ser mais efetiva devido às contribuições de vários avaliadores juntos.

Na dissertação de Buykx (2009) é utilizada a avaliação heurística colaborativa. Babajo (2012) considera o método de Buykx (2009) mais efetivo, eficiente e interessante para avaliadores especialistas. Os resultados da dissertação de Buykx (2009) mostram que com a utilização da avaliação colaborativa é possível obter resultados substancialmente mais confiáveis do que com a avaliação heurística tradicional.

Buykx (2009) descreve a realização da avaliação heurística colaborativa. Nela, é utilizado um projetor para que todos os participantes avaliem juntos as interfaces. Somente um dos participantes, denominado piloto, opera o sistema, e somente um participante registra os problemas encontrados. Cada participante preenche uma planilha com as severidades dos problemas.

Todos os participantes têm acesso aos problemas que são registrados, e podem opinar sobre a forma que o item é registrado. No entanto, não é recomendado que os participantes discutam sobre a validade ou não de um problema encontrado (BUYKX, 2009).

Danino (2001) esclarece que todos os problemas devem ser registrados. Se um participante considera um item inválido, ele deve atribuir àquele item a severidade zero. Pode ainda ser definida uma regra para a definição da severidade ao final da avaliação. Por exemplo, que se determinado número de avaliadores atribuir a severidade zero, então o problema é descartado.

Alguns pontos positivos da avaliação heurística colaborativa, dentre eles os descritos por Babajo (2012), são:

- Maior concordância entre avaliadores a respeito de problemas.
- Maior proporção de problemas severos do que a avaliação tradicional.
- Melhor experiência para os avaliadores.

- Classificação de severidade mais confiável, uma vez que os avaliadores vêem os problemas ao mesmo tempo.
- Eliminação da reunião de consolidação, os avaliadores atribuem individualmente as severidades dos problemas.
- Redução da duplicação de esforços.

No entanto, alguns pontos negativos citados são risco de resultados tendenciosos causados por determinado avaliador e restrição do escopo da avaliação devido ao limite de tempo.

2.2.2.3 Percurso Cognitivo

Percurso cognitivo (*cognitive walkthrough*) é um método orientado a tarefas, onde cada analista explora o sistema passo-a-passo para cada tarefa. Ele dá ênfase às questões cognitivas, analisando os processos mentais necessários ao usuário (HOLZINGER, 2005).

Rieman, Franzke e Redmiles (1995) explicam que os pré-requisitos para o percurso são:

- Uma descrição geral de quem os usuários serão e quais os conhecimentos relevantes eles possuem.
- Uma descrição específica de uma ou mais tarefas representativas que serão desempenhadas com o sistema.
- Uma lista das ações corretas necessárias para completar cada uma das tarefas com a interface a ser avaliada.

A avaliação é limitada a considerar se o usuário selecionará quais das ações corretas pelo caminho. Este método requer que o avaliador seja experiente, uma vez que ele deve simular o processo cognitivo do usuário.

Vantagens do percurso cognitivo são o auxílio aos *designers* para entender a perspectiva dos usuários, identificação de problemas no sistema e definição dos objetivos dos usuários. Problemas com este método podem decorrer da seleção imprópria de tarefas, ênfase em detalhes de baixo nível e não envolvimento do usuário final (HOLZINGER, 2005).

2.2.2.4 Inspeção Semiótica

Barbosa e Silva (2010) definem inspeção semiótica como um método que avalia a comunicabilidade de uma solução de IHC por meio de inspeção, com o objetivo de avaliar a qualidade da emissão da metacomunicação do *designer* codificada na interface.

O primeiro passo para inspeção semiótica é definir o propósito da avaliação (BENTO; PRATES; CHAIMOWICZ, 2009). Com isso em mente, o avaliador deve estudar informalmente o sistema para identificar o foco da avaliação, confirmar os usuários do sistema e os objetivos e atividades que ele suporta. O avaliador define o escopo da avaliação através de um ou mais cenários de interação que descrevem o contexto de uso, os usuários-alvo a serem considerados e qual parte do sistema será considerada.

Na execução da inspeção, é realizada análise de signos metalinguísticos, signos estáticos e signos dinâmicos, reconstruindo as metamensagens correspondentes (BARBOSA; SILVA, 2010). Em seguida é feita a consolidação dos resultados, onde as metamensagens são contrastadas e comparadas e os problemas de comunicabilidade encontrados são julgados. Para finalizar são relatados os resultados da avaliação de comunicabilidade da solução de IHC, sob o ponto de vista do emissor da metamensagem (BARBOSA; SILVA, 2010).

A inspeção semiótica não requer mais de um avaliador. Se houverem mais avaliadores, eles devem trabalhar em conjunto, ou cada avaliador inspecionar a interface para determinado perfil de usuário (BARBOSA; SILVA, 2010).

2.2.3 Heurísticas para sistemas Web de Petrie e Power (2012)

Diferentes conjuntos de heurísticas podem ser utilizados em uma avaliação, dependendo do sistema a ser avaliado. No estudo de Petrie e Power (2012) foi desenvolvido um conjunto de heurísticas para sistemas Web. Neste estudo foram comparados problemas de usabilidade encontrados em avaliações de seis sites governamentais complexos e altamente interativos. Os sites foram avaliados por quinze usuários potenciais e três diferentes métodos de inspeção utilizando times com três especialistas.

No estudo de Petrie e Power (2012) os participantes identificaram problemas de usabilidade e os classificaram numa escala de catastrófico a cosmético. Os problemas encontrados foram categorizados, de forma a permitir um agrupamento natural dos mesmos.

Como resultado deste estudo, foi obtido um conjunto de 21 heurísticas, divididas nas categorias: Apresentação Física, Conteúdo, Arquitetura da informação e Interatividade. Estas heurísticas são apresentadas detalhadamente no Apêndice A.

2.3 Avaliação de Usabilidade em Métodos Ágeis

Para que um software atenda os requisitos de usabilidade, é necessário que profissionais de IHC e engenheiros de software trabalhem juntos. No entanto, tanto métodos tradicionais quanto métodos ágeis ainda não são integrados de forma adequada com métodos e processos de IHC (MEMMEL; GUNDELSWEILER; REITERER, 2007).

Para Barbosa e Silva (2010) a utilização de métodos ágeis é interessante para IHC, uma vez que eles buscam interação com o cliente, têm pequenos ciclos de desenvolvimento e trabalham de forma iterativa e incremental. No entanto, Barbosa e Silva (2010) ressaltam que é preciso cuidado em relação à qualidade no

uso, pois raramente a comunidade ágil cita usuários, ou faz uma distinção entre usuário e cliente.

Em IHC a distinção entre cliente e usuário é fundamental. Ao contratar o desenvolvimento do software, o cliente muitas vezes não é claro ao informar qual o conjunto de usuários do sistema e quais as atividades desempenhadas por eles.

No método de desenvolvimento ágil *Scrum*, uma funcionalidade é desenvolvida em um *Sprint*. Assim como no processo de *design* centrado no usuário, a funcionalidade é desenvolvida, testada e documentada. Nos *Sprints* seguintes o *design* é refinado com base na evolução dos requisitos. Devido às similaridades, *design* centrado no usuário pode ser incorporado no desenvolvimento ágil sem grandes impactos no processo (NAJAFI; TOYOSHIBA, 2008).

Winter *et al.* (2011) apresentam um estudo de caso da utilização de um pacote para teste de usabilidade. Este pacote é chamado de *UIQ Technology Usability Methods (UTUM)*. O objetivo do estudo de caso foi determinar como balancear as demandas de resultados ágeis e formais na execução de testes de usabilidade para garantia de qualidade. Os dados do estudo foram obtidos através de observação, entrevistas não estruturadas, participação em reuniões com *stakeholders* e documentos do projeto. Foi concluído que o pacote de testes obteve resultados satisfatórios para a empresa estudada.

Memmel, Gundelsweiler e Reiterer (2007) mostram a utilização do *CRUISER*, um ciclo de vida de interface de usuário multidisciplinar e de Engenharia de Software. O estudo mostra que o *CRUISER* é apropriado para unir IHC e engenharia de software sob a perspectiva ágil.

Barbosa, Furtado e Gomes (2008) propõem uma estratégia para institucionalização da usabilidade com base em conceitos de desenvolvimento e gestão ágil. Foi realizado um *workshop* para reflexão sobre a importância em aplicar novas práticas. Para obtenção de informações sobre a organização, processos e experiências anteriores foram utilizados questionários. Os objetivos de negócio e o

processo integrado com as práticas de usabilidade foram definidos em três encontros com a alta direção. Algumas das contribuições obtidas pelos autores foram: a integração de práticas de usabilidade em um processo de desenvolvimento ágil e estreitamento da relação entre IHC e o negócio da organização.

Nielsen e Madsen (2012) relatam um estudo envolvendo doze profissionais em usabilidade de doze diferentes países. Foram realizadas entrevistas com os profissionais sobre inovação em negócios e novas práticas. Posteriormente, as entrevistas foram analisadas em busca de influências de métodos ágeis de desenvolvimento nos testes de usabilidade. Foram informados como benefícios as decisões mais rápidas, flexibilidade e relatos verbais. O estudo concluiu que os profissionais estão se esforçando para conseguir soluções para integrar práticas de usabilidade e métodos ágeis, e que eles veem benefícios nas práticas que os métodos ágeis os forçam a aplicar.

A utilização da abordagem de usabilidade ágil *eXtreme Scenario-based Design (XSBD)* é resumida e relatada no estudo de caso de Lee, Judge e McCrickard (2011). Neste trabalho foi concluído que vários dos desafios que o time encontrava eram relacionados à comunicação, colaboração e compartilhamento de informação. Como lições tiradas do estudo, foi recomendado ter foco nos aspectos importantes da interface, definir objetivos de alto nível e definir mecanismos de comunicação e compartilhamento de informação para incluir os membros distantes.

Wolkerstorfer *et al.* (2008) mostra adaptações ao processo clássico do *Extreme Programming (XP)*. Cinco instrumentos de IHC foram integrados com o processo do XP. Estes instrumentos são: testes de unidade estendidos para avaliação de usabilidade automatizada, *Extreme Personas* (variação do método clássico de *personas*), estudos de usuário para tomar conhecimento sobre o usuário final, avaliações feitas por especialistas em usabilidade e testes de usabilidade com usuários.

Os trabalhos apresentados mostram avanços da integração das práticas de usabilidade nos ambientes ágeis. No entanto ainda existem poucos estudos sobre este tema, e ainda há dificuldades a serem estudadas. A integração entre métodos de inspeção de usabilidade e métodos ágeis de desenvolvimento é importante. Princípios ágeis podem beneficiar o processo de inspeção de usabilidade, assim como a inspeção de usabilidade é benéfica para os resultados do desenvolvimento.

3 MÉTODOS

3.1 Tipo de Pesquisa

O método de pesquisa utilizado neste trabalho é a pesquisa observacional, caracterizada segundo a perspectiva filosófica como positivista e quanto ao estilo da pesquisa, estudo de caso. Estudos de caso são mais adequados para responder à questões explicativas, onde o pesquisador não interfere diretamente no objeto de estudo. Este trabalho se trata da análise da utilização de dois métodos de avaliação no contexto de uma empresa, portanto caracteriza um estudo de caso.

De acordo com Rampazzo (2002), estudo de caso é uma pesquisa descritiva que trabalha sobre dados ou fatos colhidos da própria realidade. Utiliza-se para isto a observação, entrevista, questionário e outras técnicas.

A pesquisa realizada combinou aspectos qualitativos e quantitativos. No estudo foram coletados dados qualitativos referentes às impressões dos avaliadores sobre os métodos. Como resultado da aplicação do estudo de caso foram obtidas as quantidades de problemas em cada método de avaliação, os graus de severidade atribuídos aos problemas, dentre outros dados quantitativos.

3.2 Procedimentos Metodológicos

3.2.1 Desenho

De acordo com Yin (2009), desenho da pesquisa pode ser definido como um plano lógico para que, a partir de um conjunto de perguntas, possa se obter um conjunto de conclusões. Este plano lógico inclui vários passos, como coleta e análise de dados relevantes.

O primeiro passo para este trabalho foi a revisão de literatura. O foco da pesquisa é a avaliação de usabilidade em ambientes que utilizam métodos ágeis, portanto a revisão de literatura envolveu conceitos e práticas de métodos ágeis,

com foco em *Scrum* e também avaliação de usabilidade, com foco na avaliação heurística, que é o método de avaliação utilizado no estudo de caso.

Foi realizado um plano de acordo com o perfil da *startup* a aplicar o estudo de caso. A empresa é de pequeno porte, e a maior parte dos colaboradores não tinha contato com questões de usabilidade.

Para viabilizar o estudo de caso, foi realizado um treinamento com os colaboradores. Era necessário que todos os participantes tivessem um nível mínimo de conhecimento para que pudessem realizar as avaliações. O objetivo principal do treinamento foi explicar os conceitos e práticas necessários para a realização da avaliação heurística.

Depois que todos tiveram o treinamento as avaliações foram realizadas. Para isto os colaboradores foram divididos em duas equipes. As equipes foram sorteadas, no entanto houve preocupação para que o nível das duas equipes fosse o mesmo. Para isto, os dois participantes que já tinham experiência com avaliação de usabilidade foram atribuídos a equipes diferentes. Os demais participantes, classificados como iniciantes, foram atribuídos aleatoriamente a uma das equipes.

As duas equipes analisaram o mesmo produto, um sistema Web desenvolvido pela empresa. Um dos grupos realizou a avaliação heurística tradicional, enquanto a outra equipe realizou a avaliação heurística colaborativa. Foi determinado o tempo de uma hora e trinta minutos para as avaliações do sistema, e depois das avaliações, a equipe da avaliação tradicional realizou uma sessão de consolidação, onde não foi determinado limite de tempo. Foram utilizadas para as avaliações as heurísticas de Petrie e Power (2012), que são mais adequadas para sistemas Web. O conjunto de heurísticas está disponível no Apêndice A.

Após as avaliações, os participantes responderam dois questionários. Um questionário, comum para todos os participantes, tinha finalidade de obter informações demográficas, como idade, gênero, formação acadêmica e informações profissionais, tais como cargos desempenhados, tempo no mercado de trabalho e

experiência. Ainda, os participantes responderam outro questionário, específico para o tipo da avaliação executada, com objetivo de tomar conhecimento da opinião do participante sobre o método utilizado.

O projeto desenvolvido não foi avaliado pelo Comitê de Ética em Pesquisas com Seres Humanos (COEP), pois, de acordo com a resolução N° 196, de 10 de outubro de 1996, as pesquisas a ser submetidas à COEP são aquelas cujo objetivo é contribuir para o conhecimento generalizável. Como o objetivo deste trabalho é observar o uso de métodos de avaliação de usabilidade especificamente em uma empresa, o conhecimento gerado não é generalizável. Portanto entendeu-se não ser necessária a submissão no mesmo, de acordo com esclarecimento realizado pela COEP-UFLA.

Entretanto, o estudo prezou pela observação de todos os princípios éticos durante sua realização. Todos os envolvidos participaram voluntariamente do estudo, e os dados coletados foram tratados de maneira confidencial e anônima. Os participantes não foram expostos a nenhum risco excessivo, e foram tomadas todas as providências para a mitigação de quaisquer riscos identificados.

3.2.2 Participantes

Participaram do estudo de caso seis colaboradores da empresa, que foram divididos em dois grupos. Todos os participantes frequentaram ou frequentam a Universidade Federal de Lavras. Ao separar os participantes houve atenção para que os grupos tivessem o mesmo nível de experiência com avaliação de usabilidade.

Na empresa há dois colaboradores que conheciam e já tinham realizado avaliação heurística, ambos fazem parte da recém criada equipe de usabilidade da empresa. Para as avaliações, eles foram separados para evitar que uma equipe fosse mais experiente que outra.

Um deles é mestre em Ciência da Computação, e tem experiência com desenvolvimento de software, gerência de projetos e elicitação de requisitos. Já tinha realizado avaliações heurísticas quando acompanhou um projeto de aplicativo móvel e auxiliou na criação de somente algumas interfaces mais complexas para outros projetos. Não tem experiência com teste, no entanto participa da implantação de sistemas e treinamento de usuários, onde observa as dificuldades dos usuários e reporta as dificuldades para a equipe de usabilidade e desenvolvedores.

A outra integrante da equipe de usabilidade que participou deste estudo é bacharel em Sistemas de Informação. Ela tem experiência como desenvolvedora e atuou em projetos de ERP e aplicativos para dispositivos móveis Android. Como membro da equipe de usabilidade participou de avaliações heurísticas nos projetos da empresa e também já participou de uma avaliação heurística para a monografia de uma aluna do curso de Sistemas de Informação em 2010. Sua experiência com testes se limita a testes unitários, e não possui experiência profissional com elicitação de requisitos.

Os outros participantes tinham conhecimento básico sobre usabilidade e avaliação heurística adquirido na graduação, na disciplina chamada "Interação Homem-Máquina". Estes participantes foram sorteados para os grupos, visto que tinham mesmo nível de conhecimento.

Um deles é bacharel em Ciência da Computação e tem experiência em desenvolvimento de sistemas na linguagem Java. Já realizou elicitação de requisitos não profissionalmente. Tem pouca experiência com testes com usuários, e nenhuma formação específica para testes. Já fez um curso com o tema de usabilidade, assistiu e participou de simulação de avaliação heurística, mas nunca aplicou os conceitos de maneira formal.

Um dos participantes declarou não possuir formação acadêmica. Ele tem experiência com levantamento de requisitos de sistemas junto ao cliente, desenvolvimento de sistemas usando diferentes linguagens e plataformas, como Ruby on

Rails, .Net e Java, elaboração e implantação de ambientes de produção de software e consultoria em redes wireless. Já participou de palestras sobre usabilidade em eventos de desenvolvimento. Tem experiência com *Test Driven Development* e *Behavior Driven Development* e participou de testes com usuários no desenvolvimento de alguns sistemas quando solicitado pelo cliente.

Dois dos participantes estão cursando bacharelado em Sistemas de Informação. Um deles tem experiência em suporte em educação a distância com moodle, desenvolvimento para a plataforma wordpress, desenvolvimento na linguagem Java e Ruby utilizando o framework Rails. Declarou já ter participado de duas ou três avaliações heurísticas, e tem conhecimento sobre usabilidade adquirido em artigos na internet. Nunca realizou teste com usuários, somente observou a realização.

O último participante tem experiência de três anos e meio de estágio com desenvolvimento na linguagem Java, gestão de processos e inteligência de negócios. No estágio, trabalha com desenvolvimento de interfaces, mas sempre seguindo um padrão definido e já participou de duas avaliações heurísticas. Tem experiência com teste de funcionalidade e não possui experiência com testes com usuários. Declarou ter experiência razoável com elicitación de requisitos, proveniente de discussões com usuários e clientes.

Dois dos participantes têm 28 anos, e os outros quatro têm 24 anos. Desta forma, a média de idades das equipes foi a mesma, de 25,3 anos.

3.2.3 Materiais

O objeto da avaliação foi um sistema Web para gerenciamento de concursos chamado Lactus Contest. O sistema abrange desde o cadastro de empresas participantes de um concurso até a emissão de relatórios sobre o desempenho das amostras fornecidas, passando pela avaliação das mesmas em suas diversas cate-

gorias e critérios. A interface do sistema foi criada a partir do Twitter Bootstrap (<http://getbootstrap.com/>).

Para a realização das avaliações foram utilizadas as heurísticas para sistemas Web de Petrie e Power (2012). Foi disponibilizada para cada participante uma lista com as heurísticas, em anexo no Apêndice A.

Foi disponibilizada uma planilha para registro dos problemas de usabilidade e as heurísticas afetadas. O grupo que realizou a avaliação colaborativa preencheu somente uma planilha, onde todos os problemas, apontados por qualquer membro do grupo, eram registrados, enquanto cada participante da avaliação tradicional recebeu uma, e ao final ainda foi gerada outra planilha com o resultado de todos. Esta planilha é mostrada no Apêndice B.

Cada participante da avaliação tradicional e colaborativa recebeu ainda uma planilha para registro da severidade dos problemas, apresentada no Apêndice B.

Foram utilizados questionários para coletar informações demográficas, profissionais e também a opinião dos participantes sobre o método executado. Estes questionários encontram-se no Apêndice C.

3.2.4 Procedimento

Esta subseção apresenta os procedimentos realizados para o estudo de caso.

3.2.4.1 Treinamento

A equipe passou por um treinamento envolvendo interação humano-computador, experiência do usuário e avaliação de usabilidade.

O treinamento teve duas partes. Na primeira parte foram apresentados conceitos de usabilidade e interação humano-computador. Foram mostradas aos

participantes as heurísticas de Nielsen (1994) e os conceitos de avaliação heurística.

Na segunda parte do treinamento, os participantes realizaram uma avaliação heurística colaborativa do site de uma companhia aérea. Desta forma eles puderam consolidar os conhecimentos adquiridos na parte teórica.

3.2.4.2 Avaliação Tradicional

A avaliação heurística tradicional é composta de duas fases: primeiro o sistema é avaliado, e depois os avaliadores se reúnem e compartilham suas avaliações, originando um documento com a avaliação de todos a partir das discussões em uma sessão de consolidação dos resultados.

Os participantes da avaliação tradicional fizeram a avaliação do sistema individualmente. Depois das avaliações, a equipe realizou uma reunião de consolidação, onde os problemas encontrados por cada avaliador foram apresentados aos outros. Os participantes avaliaram a validade e severidade de cada problema, e foi gerada uma planilha final com todos os problemas. Para a sessão de consolidação não houve limitação de tempo, uma vez que era interessante medir o tempo gasto.

3.2.4.3 Avaliação Colaborativa

Na avaliação heurística colaborativa, os participantes se reuniram e avaliaram o sistema de forma conjunta. Como a equipe era pequena, o mesmo avaliador foi piloto, responsável por operar o sistema, e escrivão, responsável por registrar os problemas apontados pelo grupo na planilha de registro de problemas. Todos os problemas levantados pelos avaliadores foram considerados. Quando um avaliador considerava um problema inválido, ele simplesmente atribuía severidade nula na sua planilha individual de atribuição de severidade, para evitar perder tempo com discussões durante a avaliação.

Ao final da avaliação, foi entregue uma planilha com todos os problemas encontrados e três planilhas de severidade, uma de cada avaliador.

3.2.5 Análise de Dados

De acordo com Yin (2003) tanto dados qualitativos quanto quantitativos são relevantes na aplicação de um estudo de caso. Para ele, esta dualidade reforça o posicionamento do estudo de caso como um método que não é limitado ao tipo de dados. O método de estudo de caso possui *design*, coleta de dados e procedimentos analíticos próprios.

Neste estudo de caso foram utilizados critérios qualitativos e quantitativos para análise de dados. Os critérios são descritos a seguir.

A análise quantitativa do número de problemas mostrou o número de problemas encontrados em cada abordagem de avaliação heurística. Na avaliação tradicional, foram relatados os problemas encontrados em cada avaliação e também o resultado final após a sessão de consolidação, e na avaliação colaborativa, foram relatados os problemas encontrados pelo grupo.

Foi também efetuada uma análise quantitativa do número de violações às heurísticas de usabilidade. Como os problemas podem violar mais que uma heurística, esta análise se mostrou útil para mostrar de forma mais específica os pontos falhos do sistema e a cobertura e utilização das heurísticas nos diferentes tipos de avaliação.

A análise quantitativa da severidade dos problemas foi utilizada para comparar os níveis de severidade dos problemas encontrados em cada abordagem. Para determinar a severidade dos problemas encontrados na avaliação colaborativa foi utilizada a média aritmética das severidades atribuídas por cada avaliador. Nos casos onde a média das severidades não foi um número inteiro, foi utilizado arredondamento natural.

Por fim, a análise qualitativa das respostas dos questionários relacionados às impressões dos participantes sobre cada tipo de avaliação foi utilizada para identificar a adequação de cada método ao ambiente da empresa-alvo do estudo. Para isto, foi feita uma análise de conteúdo sobre as respostas dadas pelos participantes, visando identificar os principais temas relacionados à forma de aplicação e à integração do método no desenvolvimento.

4 RESULTADOS E DISCUSSÃO

Este capítulo descreve os dados obtidos com a realização do estudo de caso.

Atribuiu-se a nomenclatura de problema de usabilidade aos problemas encontrados na interface pelos avaliadores. Cada problema poderia violar mais de uma heurística. Para cada heurística afetada por um problema foi utilizada a nomenclatura de violação de heurística. Também foi utilizada uma variável para identificar o número de vezes que cada heurística foi violada.

Os avaliadores que participaram da avaliação tradicional serão denominados AT1, AT2 e AT3, enquanto os participantes da avaliação colaborativa serão denominados AC1, AC2 e AC3.

4.1 Violações das Heurísticas nas Avaliações

Esta seção apresenta o número de problemas e as heurísticas violadas pelos mesmos para cada uma das avaliações. São descritos os grupos de heurísticas afetados pelos problemas encontrados, de acordo com a divisão realizada por Petrie e Power (2012).

A quantidade de violações por grupo de heurísticas pode exceder a quantidade de problemas encontrados, uma vez que cada grupo pode englobar várias heurísticas, e cada problema de usabilidade pode violar mais que uma heurística. O conjunto completo das heurísticas para sistemas Web de Petrie e Power (2012) encontra-se no Apêndice A.

4.1.1 Avaliação Heurística Tradicional

Cada participante realizou uma avaliação do sistema individualmente. A seguir são mostrados os resultados destas avaliações.

O avaliador AT1 encontrou dezenove problemas. A Tabela 4.1 mostra o número de pontos em que cada heurística foi violada, separando por grupos de heurísticas, de acordo com a classificação de heurísticas feita por Petrie e Power (2012).

Foram encontradas dez violações no grupo de heurísticas de interatividade do sistema. Os violações encontradas foram referentes a instruções e rótulos, repetição de ações, *feedback* para as ações do usuário, sequência de interação e convenções, apresentação de funcionalidades interativas que o usuário precisa e espera, distinção entre elementos interativos e não-interativos e apresentação de mensagens de erros informativas.

Nove violações afetam o grupo de heurísticas referentes ao conteúdo. São violações às heurísticas que determinam a apresentação de conteúdo relevante e apropriado, quantidade adequada de informação e utilização de termos e abreviações claras.

Foram encontradas cinco violações às heurísticas de apresentação do sistema. Estas violações interferem na clareza dos elementos interativos e *layout* da página e destaque dos principais conteúdos e mudanças no conteúdo.

A arquitetura da informação apresentou quatro violações. A heurística de arquitetura da informação diz respeito à apresentação de estruturas de informação claras e bem organizadas.

A heurística que apresentou mais pontos falhos no sistema foi a heurística de número sete. Ela faz parte do grupo de heurísticas do conteúdo do sistema e define que deve-se utilizar termos e abreviações claras e evitar termos de difícil entendimento.

A heurística de número oito, de arquitetura da informação, também apresentou alto número de violações. Ela orienta o fornecimento de estruturas de organização de informação claras e bem organizadas.

Tabela 4.1: Número de violações de heurísticas na avaliação realizada pelo AT1

Heurísticas	Número de violações
Apresentação Física	5
H1 - Textos e elementos interativos devem ser grandes e claros	1
H2 - O <i>layout</i> da página deve ser claro	2
H4 - Os principais conteúdos e possíveis mudanças de conteúdo devem ser destacados	2
Conteúdo	9
H5 - Forneça conteúdo relevante e apropriado para o site	3
H6 - Forneça conteúdo em quantidade suficiente, mas não excessiva	1
H7 - Utilize termos e abreviações claras	5
Arquitetura da Informação	4
H8 - Forneça estruturas de organização da informação claras e bem organizadas.	4
Interatividade	10
H10 - Instruções e rótulos claros	1
H11 - Evite esforço excessivo e repetição de ações	1
H13 - Forneça <i>feedback</i> para as ações do usuário e para mostrar o estado do sistema	1
H14 - Faça com que a sequência de interação seja lógica	2
H16 - Siga convenções para interação	1
H17 - Forneça as funcionalidades interativas que os usuários precisam e esperam do sistema	1
H19 - Elementos interativos e não-interativos devem ser fáceis de distinguir	2
H21 - Forneça mensagens de erros informativas e que auxiliem os usuários a se recuperar de erros	1

O avaliador AT2 encontrou vinte e nove problemas de usabilidade no sistema. Como apresentado na Tabela 4.2, em trinta e sete pontos, heurísticas do grupo referente à interatividade foram violadas. A heurística que define padrões para o conteúdo do sistema foi violada em seis pontos, quatro violações foram encontradas quanto à apresentação física e uma violação na arquitetura da informação.

Nesta avaliação, a heurística que apresentou mais violações foi a de número dez, que integra o grupo de interatividade do sistema. Foram encontradas nesta avaliação quatorze ocorrências de violações a esta heurística. Nela é definido que as instruções e rótulos devem ser claros, e que devem ser utilizados padrões já conhecidos.

O avaliador AT3 encontrou quatro problemas no sistema. A Tabela 4.3 mostra resultados desta avaliação.

Foram encontrados sete pontos onde heurísticas referentes à interatividade foram violadas. Foram encontrados três violações à arquitetura da informação, uma violação no conteúdo e uma na apresentação física do sistema.

A heurística que obteve maior número de violações foi a heurística oito, com três ocorrências. A heurística oito se trata do fornecimento de estruturas de organização de informação claras e bem organizadas. Foram encontradas duas violações que afetam as heurísticas dez e doze, integrantes das heurísticas de interatividade do sistema. A heurística dez diz respeito a instruções e rótulos claros, e a heurística doze orienta que os formatos de entrada de dados sejam claros e fáceis.

Após as avaliações, foi realizada uma sessão de consolidação dos resultados. Um dos avaliadores havia saído da empresa, portanto foi decidido que somente dois deles participariam da sessão, uma vez que este avaliador é o que havia identificado o menor número de problemas entre os três. Os dois participantes da reunião incluíram nos resultados finais a avaliação do terceiro participante. A reunião de consolidação durou uma hora e vinte e dois minutos.

Tabela 4.2: Número de violações de heurísticas na avaliação realizada pelo AT2

Heurísticas	Número de violações
Apresentação Física	4
H2 - O <i>layout</i> da página deve ser claro	4
Conteúdo	6
H5 - Forneça conteúdo relevante e apropriado para o site	2
H6 - Forneça conteúdo em quantidade suficiente, mas não excessiva	4
Arquitetura da Informação	1
H8 - Forneça estruturas de organização da informação claras e bem organizadas.	1
Interatividade	37
H10 - Instruções e rótulos claros	14
H11 - Evite esforço excessivo e repetição de ações	1
H12 - Faça com que os formatos de entrada de dados sejam claros e fáceis	1
H13 - Forneça <i>feedback</i> para as ações do usuário e para mostrar o estado do sistema	7
H14 - Faça com que a sequência de interação seja lógica	3
H16 - Siga convenções para interação	3
H17 - Forneça as funcionalidades interativas que os usuários precisam e esperam do sistema	1
H19 - Elementos interativos e não-interativos devem ser fáceis de distinguir	6
H21 - Forneça mensagens de erros informativas e que auxiliem os usuários a se recuperar de erros	1

Na sessão de consolidação foram identificados trinta e oito problemas. A Tabela 4.4 mostra a quantidade de vezes em que cada heurística foi violada.

Tabela 4.3: Número de violações de heurísticas na avaliação realizada pelo AT3

Heurísticas	Número de violações
Apresentação Física	1
H1 - Textos e elementos interativos devem ser grandes e claros	1
Conteúdo	1
H5 - Forneça conteúdo relevante e apropriado para o site	1
Arquitetura da Informação	3
H8 - Forneça estruturas de organização da informação claras e bem organizadas.	3
Interatividade	7
H9 - Forneça informações claras sobre a interação com o sistema ("como e porque")	1
H10 - Instruções e rótulos claros	2
H11 - Evite esforço excessivo e repetição de ações	1
H12 - Faça com que os formatos de entrada de dados sejam claros e fáceis	2
H15 - Forneça um conjunto de opções lógico e completo	1

Dos resultados consolidados, a heurística dez, que define a utilização de instruções e rótulos claros, teve quatorze violações. As heurísticas oito e treze também obtiveram alto número de violações, respectivamente, oito e sete pontos. A heurística oito constitui o grupo da arquitetura da informação, e define a organização das estruturas de informação. A heurística treze, integrante do grupo de interatividade do sistema, define o fornecimento de *feedback* para as ações realizadas pelo usuário.

Assim como nas avaliações individuais, o grupo de heurísticas mais afetado foi o de interatividade, onde foram encontrados quarenta e uma violações.

Tabela 4.4: Número de violações por grupo de heurísticas após a sessão de consolidação

Heurísticas	Nº de violações
Apresentação Física	10
H1 - Textos e elementos interativos devem ser grandes e claros	2
H2 - O <i>layout</i> da página deve ser claro	6
H4 - Os principais conteúdos devem ser destacados	2
Conteúdo	11
H5 - Forneça conteúdo relevante e apropriado para o site	5
H6 - Forneça conteúdo em quantidade suficiente, mas não excessiva	1
H7 - Utilize termos e abreviações claras	5
Arquitetura da Informação	8
H8 - Forneça estruturas de organização claras e bem organizadas.	8
Interatividade	41
H9 - Forneça informações claras sobre a interação com o sistema	1
H10 - Instruções e rótulos claros	14
H11 - Evite esforço excessivo e repetição de ações	2
H12 - Faça com que os formatos de entrada de dados sejam claros e fáceis	3
H13 - Forneça <i>feedback</i> para as ações do usuário e estado do sistema	7
H14 - Faça com que a sequência de interação seja lógica	3
H15 - Forneça um conjunto de opções lógico e completo	1
H16 - Siga convenções para interação	2
H17 - Forneça as funcionalidades que os usuários precisam e esperam do sistema	2
H19 - Elementos interativos e não-interativos devem ser fáceis de distinguir	5
H20 - Agrupe elementos interativos de forma lógica e clara	1
H21 - Forneça mensagens de erros informativas	1

Quanto ao conteúdo, foram encontrados onze violações. A apresentação física apresentou dez violações e a arquitetura da informação, oito.

4.1.2 Avaliação Heurística Colaborativa

Como resultado da avaliação heurística colaborativa foram encontrados vinte e dois problemas de usabilidade no sistema. A Tabela 4.6 mostra a quantidade de violações para cada grupo de heurísticas.

As heurísticas que mais apresentaram violações no sistema foram as referentes à interatividade. Foram encontradas treze violações relacionadas a este grupo. As violações encontradas para este grupo de heurísticas foram relacionadas com formatos de entrada de dados, *feedback* para as ações do usuário, sequência de interação, apresentação de um conjunto de opções lógico, apresentação de funcionalidades interativas que o usuário precisa e espera e fornecimento de mensagens de erro informativas.

Foram encontradas quatro violações à apresentação física do sistema. A apresentação física do sistema teve violações quanto ao *layout* da página e visibilidade dos elementos interativos.

Três violações foram identificadas quanto ao conteúdo do sistema. As violações são relacionadas à relevância e quantidade do conteúdo da página.

Outras três violações foram encontradas na arquitetura da informação. Estas violações dizem respeito à estrutura de organização de informação adequadas para auxiliar os usuários a completar tarefas.

A heurística que apresentou mais problemas foi a de número treze, que define que deve ser fornecido retorno para as ações realizadas pelo usuário e demonstrado o estado do sistema. A heurística dezessete apresentou três problemas. De acordo com ela, devem ser fornecidas as funcionalidades interativas que o usuário precisa e espera do sistema.

Tabela 4.6: Número de violações por grupo de heurísticas na avaliação colaborativa

Heurísticas	Número de violações
Apresentação Física	4
H2 - O <i>layout</i> da página deve ser claro	2
H4 - Os principais conteúdos e possíveis mudanças de conteúdo devem ser destacados	2
Conteúdo	3
H5 - Forneça conteúdo relevante e apropriado para o site	2
H6 - Forneça conteúdo em quantidade suficiente, mas não excessiva	1
Arquitetura da Informação	3
H8 - Forneça estruturas de organização da informação claras e bem organizadas.	3
Interatividade	13
H12 - Faça com que os formatos de entrada de dados sejam claros e fáceis	2
H13 - Forneça <i>feedback</i> para as ações do usuário e para mostrar o estado do sistema	5
H14 - Faça com que a sequência de interação seja lógica	1
H15 - Forneça um conjunto de opções lógico e completo	1
H17 - Forneça as funcionalidades interativas que os usuários precisam e esperam do sistema	3
H21 - Forneça mensagens de erros informativas e que auxiliem os usuários a se recuperar de erros	1

4.1.3 Comparação entre as violações das heurísticas nas duas versões de avaliação heurística

Foi feita uma comparação entre o número de violações das heurísticas utilizadas nas duas versões do método de avaliação heurística. Um teste de ranking

com sinais de Wilcoxon mostrou uma diferença significativa entre o número de violações de heurísticas entre a avaliação heurística tradicional e colaborativa ($W+ = -3,024$, $N=21$, $p < 0,01$).

Foi observado que houve mais violações no método tradicional. Um número maior de heurísticas foi mencionado na avaliação tradicional do que na avaliação colaborativa.

4.2 Número de Problemas Encontrados

A avaliação tradicional encontrou 38 problemas, enquanto a avaliação colaborativa encontrou 22. As duas avaliações encontraram um total de 60 problemas. Considerando que alguns problemas foram encontrados nos dois métodos, foram encontrados 55 problemas únicos. A Figura 4.1 mostra o número de problemas em comum encontrados nas avaliações heurísticas tradicional e colaborativa.

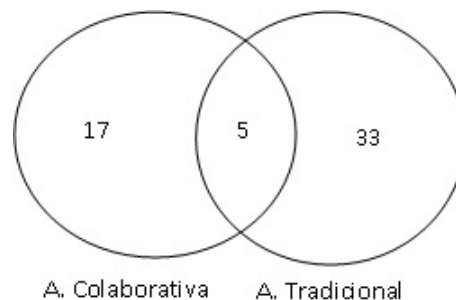


Figura 4.1: Problemas encontrados nas duas abordagens

Cinco problemas foram encontrados nas duas modalidades de avaliação. Outros problemas encontrados foram exclusivamente da avaliação tradicional ou da avaliação colaborativa.

4.3 Severidades dos problemas encontrados

Esta seção apresenta as severidades dos problemas encontrados nas duas modalidades de avaliação heurística.

4.3.1 Avaliação Heurística Tradicional

Na avaliação tradicional foram encontrados trinta e oito problemas de usabilidade. A Figura 4.2 mostra a quantidade de problemas para cada severidade atribuída.

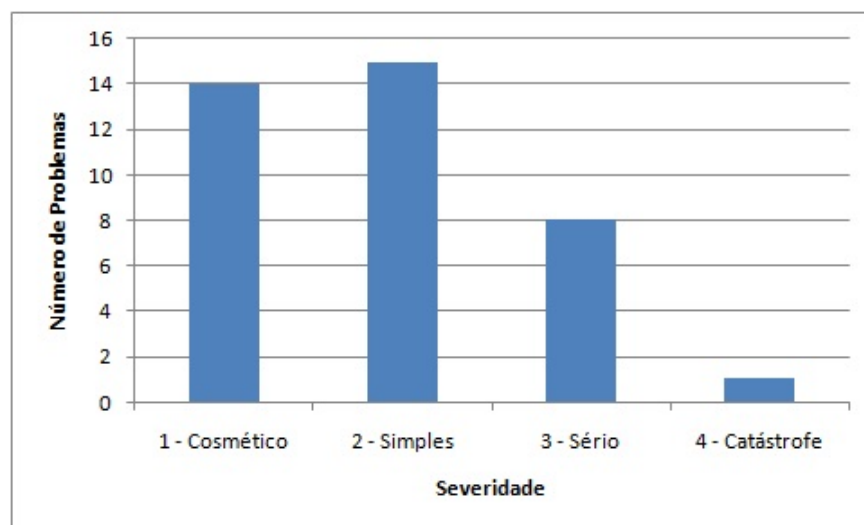


Figura 4.2: Severidade dos problemas encontrados na avaliação tradicional

A maior parte dos problemas foi de severidade um e dois, que são problemas cosméticos e simples, respectivamente. Quatorze problemas foram considerados cosméticos, e quinze problemas considerados simples. Problemas simples devem ter prioridade menor no desenvolvimento do sistema e problemas cosméticos só devem ser resolvidos caso haja tempo disponível no projeto.

Um problema recebeu a severidade quatro, que é definido como catástrofe. Considerou-se que o sistema não pode ser lançado antes de este problema ser re-

solvido. A severidade três, que são problemas sérios, teve oito ocorrências. Estes problemas são os de maior prioridade depois dos catastróficos.

4.3.2 Avaliação Heurística Colaborativa

Foram encontrados vinte e dois problemas de usabilidade no sistema com a utilização da avaliação heurística colaborativa. A severidade de cada problema foi obtida a partir da média entre as severidades atribuídas pelos três avaliadores e foi utilizado arredondamento natural. A Figura 4.3 apresenta a severidade dos problemas encontrados.

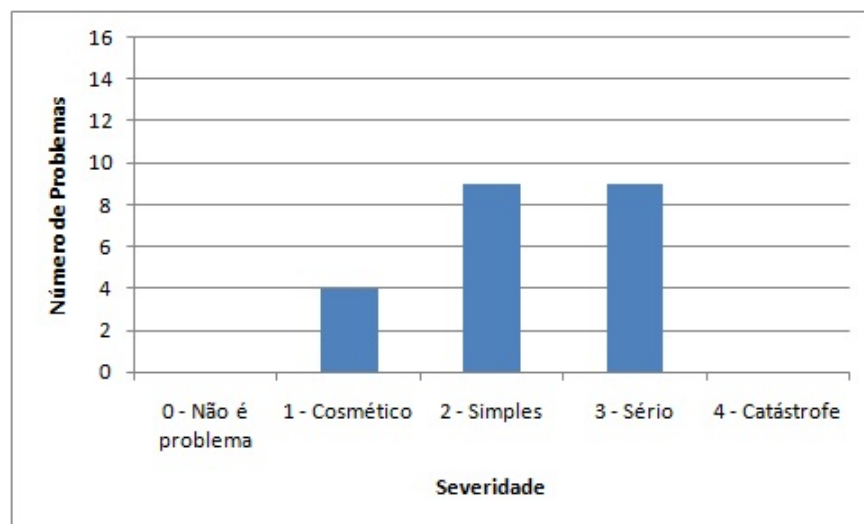


Figura 4.3: Severidade dos problemas encontrados na avaliação colaborativa

A avaliação heurística colaborativa possui a opção de severidade zero, não existente na avaliação tradicional. Isto ocorre devido à avaliação colaborativa não realizar sessão de consolidação de resultados, portanto um avaliador pode considerar que um problema identificado por outro avaliador não é realmente um problema. Na avaliação tradicional os avaliadores discutem e entram em consenso sobre os problemas, descartando problemas quando necessário.

As severidades dois e três, que representam problemas simples e sérios, tiveram o mesmo número de ocorrências. É importante que problemas sérios sejam corrigidos antes do lançamento do produto. Já problemas simples têm prioridade menor no desenvolvimento de um sistema. Foram registrados nove problemas de cada um destes níveis de severidade.

Quatro problemas foram identificados com a severidade um, que indica problemas cosméticos. A correção destes problemas só deve ser realizada se houver tempo disponível.

As severidades zero e quatro, que representam, respectivamente, pontos que não foram considerados problemas e problemas catastróficos, não foram registradas.

A severidade mínima atribuída por AC1 foi 1 e a máxima foi 4. A média das severidades atribuídas por ele foi de 3,14 e o desvio padrão 0,834. Os outros avaliadores atribuíram severidades entre 0 e 3. A média das severidades atribuídas por AC2 foi de 1,95 e o desvio padrão foi 1,046. A média de AC3 foi 1,59 e o desvio padrão foi 0,959.

Um teste de Friedman de duas vias com análise de variância por rankings para amostras relacionadas mostrou evidências de diferenças significativas entre as severidades atribuídas pelos avaliadores ($\chi^2 = 28,261$, $N=22$, $df=2$, $p < 0,001$). Análises post-hoc com testes de ranking de sinais de Wilcoxon mostraram diferenças significativas entre as severidades dos avaliadores AC1 e AC2 ($p < 0,001$) e entre os avaliadores AC1 e AC3 ($p < 0,001$), não houve diferença significativa entre os avaliadores AC2 e AC3 ($p = 0,123$). O avaliador AC1 era o mais experiente.

4.3.3 Comparação entre as severidades dos problemas encontrados nas duas versões do método

Somente cinco problemas foram encontrados em ambas, portanto não foi possível realizar uma análise aprofundada sobre o nível de concordância entre as

severidades atribuídas nos dois métodos. A análise dos cinco problemas encontrados nas duas avaliações verificou que três deles tinham o mesmo nível de severidade. Um deles teve severidade 1 na avaliação tradicional e 3 na avaliação colaborativa e um problema teve severidade 3 na avaliação tradicional e 2 na colaborativa.

Foi feita uma comparação entre os níveis de severidade dos problemas que foram encontrados exclusivamente na avaliação heurística tradicional e na avaliação heurística colaborativa. Um teste de Mann-Whitney para amostras independentes mostrou uma diferença significativa entre a severidade média dos problemas encontrados na avaliação colaborativa e o nível de severidade da avaliação tradicional ($MW = -2.072$, $N=50$, $p < 0,05$). A média das severidades atribuídas na avaliação tradicional foi de 1,89, e a média encontrada na avaliação colaborativa foi de 2,22.

4.4 Análise do Processo de Avaliação

Os participantes responderam individualmente questionários sobre utilização de cada versão do método de avaliação. Apenas dois dos participantes da avaliação tradicional responderam o questionário sobre a avaliação realizada.

Os participantes da avaliação heurística tradicional declaram que não tiveram dificuldades para concordar sobre a validade de problemas encontrados. Segundo eles, a reunião foi tranquila e eles chegaram rapidamente a um consenso. Também não foi relatada dificuldade para identificar problemas iguais descritos de forma diferente. Para os avaliadores, como o sistema era simples, os problemas eram fáceis de identificar e quando um deles reportava um problema existente em sua lista, os outros eram capazes de dizer que haviam encontrado o mesmo problema. Em alguns casos foi necessário reescrever problemas devido aos comentários dos dois avaliadores serem complementares.

Os avaliadores declaram que tiveram dificuldade para entrar em acordo quanto ao grau de severidade. Os participantes discutiram bastante sobre o assunto. Foi necessário estabelecer critérios sobre o significado de cada grau no contexto do sistema e considerando o perfil do usuário. Um deles descreveu que houve momentos em que cada um sugeria uma severidade diferente e, mesmo não concordando completamente, uma pessoa teve que ceder. Para ele, este tipo de situação pode onerar bastante o processo, dependendo do tamanho do sistema.

Quanto às heurísticas, não foi relatada dificuldade. Os participantes fizeram observações sobre os casos de problemas iguais encontrados por diferentes avaliadores. Um deles relatou que algumas vezes as heurísticas não eram as mesmas, no entanto as heurísticas atribuídas por ambos avaliadores faziam sentido no contexto do problema. O outro avaliador observou que muitas vezes heurísticas se repetiam nestes problemas.

Já quanto ao esquema de severidade eles mostraram que tiveram algumas dificuldades. Para um deles o esquema de severidade deixou muito espaço para subjetividade. Ele declarou que a maior dificuldade foi parametrizar o que caracterizava severidade dois ou três. Para a realização da consolidação foi necessário que o grupo definisse melhor o significado de cada grau de severidade.

Um dos avaliadores declarou que considerou o método válido para ajudar na identificação de problemas de interface e usabilidade. No entanto, ele declarou que ficou preocupado com o tempo que as discussões podem tomar. Ele considerou que, com a possibilidade de um grupo maior, a sessão de consolidação ficaria cada vez mais complicada e demorada.

O outro participante, que é membro da equipe de usabilidade da empresa, relatou que um ponto desfavorável do método que ele somente apresenta problemas, e não soluções. Ele baseou esta conclusão no funcionamento atual da empresa. No processo de criação de uma interface são elaborados diferentes *mockups* para a mesma funcionalidade. Estes *mockups* são avaliados por pelo menos três

membros da equipe de usabilidade, que são notas para cada opção. Ao final, o *mockup* que obter maior nota é utilizado no desenvolvimento. O mesmo processo acontece quando um problema é encontrado na interface. Neste caso, são elaborados *mockups* alternativos, que são avaliados juntamente com a interface atual. Se a interface atual obtiver melhor nota, ela é mantida, caso contrário a interface é alterada. Ele alega que desta forma é garantido que a equipe desenvolveu a melhor interface que conseguiu projetar. Por fim, ele considerou que uma boa solução seria a fusão entre os dois métodos. Desta forma, primeiro seria realizada a avaliação heurística para apontar os principais problemas que precisam ser corrigidos. Em seguida, propostas de interface seriam montadas e avaliadas segundo o métodos já utilizados na empresa.

Na avaliação heurística colaborativa nenhum participante declarou ter sentido dificuldade em expressar sua opinião, relatar problemas em voz alta ou entender de qual problema o grupo estava falando. Os participantes declararam não ter se sentido influenciados em nenhum momento da avaliação. Um deles declarou que o grupo decidiu os problemas em conjunto e todos respeitaram as opiniões uns dos outros sobre as falhas encontradas no sistema. Também foi afirmado que com o uso do projetor todos podiam ver a tela e era fácil apontar onde estavam os problemas.

Quanto às heurísticas e esquema de severidade, o grupo que realizou a avaliação colaborativa declarou não ter tido dificuldades. Afirmaram que as heurísticas foram bem explicadas no treinamento e que estavam descritas de forma clara no documento disponibilizado para consulta. Um dos participantes manifestou, entretanto, que os graus do esquema de severidade poderiam ser explicados mantendo um padrão.

Os participantes também afirmaram que não tiveram problemas devido ao fato de apenas uma pessoa operar o sistema, pois o piloto também executava o que era instruído pelos outros avaliadores.

Sobre a adequação do método ao esquema de trabalho da empresa, um dos participantes opinou que considera a avaliação importante, pois os desenvolvedores não conseguem encontrar todos os problemas da interface. Também foi declarado que seria interessante a aplicação da avaliação de usabilidade a cada iteração do desenvolvimento, pois assim seriam avaliadas partes menores do sistema e aumentaria a visão crítica da equipe para o desenvolvimento de novas funcionalidades.

Um dos participantes considerou o método válido por abrir possibilidades para identificação dos problemas da aplicação, que muitas vezes é mais oneroso e menos produtivo na avaliação individual. Foram consideradas válidas as discussões entre o grupo, pois ao ouvir a opinião do outro é possível aprender a olhar diferentes aspectos do sistema, de forma a aperfeiçoar o próprio desempenho em avaliações futuras. No entanto também foi julgado que, devido à experiência neste projeto, a avaliação deveria ser executada por profissionais mais experientes na área de usabilidade. De acordo com este participante, as discussões levaram tempo acima do esperado, pois muitas vezes os desenvolvedores não tinham maturidade para aceitar um problema que foi encontrado, mesmo que fosse considerado um problema cosmético. Para ele, esta postura dificulta o andamento da avaliação e torna discussões menos produtivas.

4.5 Discussão

4.5.1 Número de problemas encontrados nas duas versões do método

Ao analisar os resultados das duas modalidades de avaliação heurística, observou-se que na avaliação tradicional foram encontrados mais problemas de usabilidade do que na avaliação colaborativa. É possível identificar alguns fatores que podem ter levado à estes resultados.

Um fator é o perfil dos avaliadores. Foi dada atenção à divisão das equipes quanto à experiência dos participantes, no entanto existe a possibilidade de um avaliador ser mais atencioso ou exigente que os outros, independente da sua experiência. Outro fator, que foi citado por um dos participantes da avaliação colaborativa, foi o tempo gasto para discussões durante a avaliação. Considerou-se que o tempo gasto foi maior que o esperado devido ao nível de maturidade dos participantes. O fato de alguns dos participantes das avaliações atuarem como desenvolvedores no sistema avaliado também pode ter influenciado os resultados.

Hornbæk e Frøkjær (2008) mostram que o chamado "efeito avaliador" é um fator que interfere significativamente nos resultados das avaliações. O efeito avaliador pode interferir na comparação de problemas, pois dois problemas podem parecer iguais para um avaliador e diferentes para outro. Hornbæk e Frøkjær (2008) também afirmam que este efeito existe tanto para a combinação de problemas quanto da identificação dos mesmos.

A existência do efeito avaliador neste estudo de caso compromete a generalização dos resultados para outros casos. Mais estudos com diferentes grupos de avaliadores seriam necessários para confirmar tendências sobre o número de problemas encontrados pelas duas versões da avaliação heurística.

Observou-se que o número de problemas em comum entre as avaliações foi pequeno. Este resultado pode ter sido obtido devido às áreas da aplicação exploradas por cada avaliação. Os participantes da avaliação colaborativa ficaram focados em uma funcionalidade, enquanto a tradicional mostrou problemas de usabilidade em diferentes lugares do sistema. Mais estudos são necessários para determinar se as áreas exploradas em sistemas na avaliação heurística colaborativa podem ser diferentes da cobertura da avaliação heurística tradicional.

4.5.2 Severidade dos problemas encontrados exclusivamente por um ou outro método

Foi mostrado que a severidade média dos problemas foi maior na avaliação colaborativa, assim como na dissertação de Buykx (2009). Análises realizadas concluíram que a severidade média da avaliação colaborativa foi de 2,22, enquanto a média da avaliação tradicional foi de 1,89. A partir destes dados é possível concluir que a avaliação heurística colaborativa tende a revelar problemas de usabilidade mais sérios. Isto se mostra como fator favorável, considerando-se a necessidade de descobrir os problemas mais graves o mais cedo possível durante o desenvolvimento, uma vez que estes podem causar prejuízos maiores no futuro.

Identificou-se que houve diferença entre as médias das severidades atribuídas pelos participantes da avaliação colaborativa. O avaliador AC1 era o mais experiente da equipe, e a média das severidades atribuídas por ele foi maior do que as médias dos outros avaliadores. A partir destes dados, e levando em conta que todos os participantes são desenvolvedores, é possível inferir que avaliadores menos experientes tendem a considerar problemas de usabilidade menos sérios do que um especialista considera.

Estes dados reforçam a necessidade de observar cuidadosamente a experiência dos profissionais que efetuam as avaliações na organização. É de grande importância que os avaliadores tenham experiência e conhecimento sobre como diferentes usuários encontram diferentes tipos de problemas em sistemas e como podem ser afetados por problemas de usabilidade na realização de suas tarefas.

4.5.3 Produtividade da avaliação heurística tradicional e colaborativa

Apesar do número de problemas encontrados na avaliação tradicional ter sido maior, ela consumiu mais tempo, devido à necessidade da reunião de consolidação de resultados. Ambas as modalidades tiveram sessões de avaliação de

mesma duração: uma hora e trinta minutos. No entanto, a sessão de consolidação da avaliação tradicional durou uma hora e vinte e dois minutos. Desta forma, a avaliação tradicional tomou tempo significativamente maior. Se for comparado o tempo total gasto com as avaliações juntamente com o tempo para consolidar as listas individuais de problemas em uma lista única, a relação entre problemas e tempo gasto da avaliação tradicional não se mostra tão grande em relação à avaliação colaborativa.

4.5.4 Dificuldades para a atribuição de graus de severidade

Os participantes da avaliação tradicional tiveram mais dificuldades para estabelecer as severidades dos problemas. Já na avaliação colaborativa não foi identificada dificuldade. Considera-se favorável à avaliação colaborativa o fato de os participantes avaliarem as interfaces ao mesmo tempo. Isto torna mais fácil a argumentação entre eles sobre os problemas existentes e sobre o impacto de cada problema na usabilidade do sistema.

4.5.5 Utilização das heurísticas nas avaliações

Quanto às heurísticas violadas, observou-se que a avaliação heurística tradicional teve uma maior cobertura. Mais heurísticas foram atribuídas para os problemas encontrados. A possível causa deste resultado é que os participantes da avaliação tradicional tiveram mais preocupação em procurar em todo o grupo de heurísticas as que se aplicavam ao problema encontrado, enquanto na avaliação colaborativa quando um avaliador encontrava uma heurística que se aplicava, ela já era registrada e os avaliadores passavam para o próximo problema. Outra possível causa deste resultado é que os avaliadores da avaliação tradicional podem ter associado problemas a diferentes heurísticas, e na sessão de consolidação eles atribuíram todas elas ao problema. Este resultado da execução do estudo de caso levanta questões que devem ser investigadas em estudos futuros, com objetivo de

investigar de forma mais aprofundada a relação entre a cobertura das heurísticas utilizadas nos métodos tradicional e colaborativo.

4.5.6 Adequação do método de avaliação heurística colaborativa para métodos de desenvolvimento ágil

A avaliação colaborativa apresenta fatores em comum com os princípios ágeis, dentre eles, o compartilhamento de informações. Shore e Warden (2008) relatam que a integração do time é crucial, e que o conhecimento deve ser buscado e compartilhado no ambiente ágil. Participantes da avaliação colaborativa consideraram o método válido por permitir que isso aconteça. A participação de membros com diferentes níveis de conhecimento contribui para a formação dos participantes à medida que os menos experientes têm contato com o método juntamente com os mais experientes.

A excelência técnica e o bom *design* também são focos da filosofia ágil. Estes objetivos são alcançados com mais facilidade quando há integração entre o processo de desenvolvimento e as práticas de usabilidade. A avaliação heurística colaborativa auxilia na melhoria dos conhecimentos da equipe, uma vez que os participantes trabalham conjuntamente e acabam trocando experiências.

Empresas que utilizam métodos ágeis se preocupam em minimizar a quantidade de trabalho que não precisou ser feito. Com a realização da avaliação heurística são identificados os problemas de usabilidade, e eles podem ser corrigidos antes de o sistema chegar ao usuário. A versão colaborativa do método ainda contribui para encontrar problemas mais sérios.

Um dos participantes da avaliação heurística colaborativa relatou que considera adequado que sejam realizadas avaliações a cada iteração do desenvolvimento. Desta forma o escopo da avaliação seria menor. Esta ideia é compartilhada com métodos ágeis. Highsmith (2002) destaca que utilizando métodos ágeis é importante dividir o trabalho em pequenas tarefas, para que sejam executadas mais

rápido e com menor possibilidade de erros. Este princípio também auxilia a evitar a realização de atividades fora do escopo.

A avaliação heurística colaborativa pode ser integrada com desenvolvimento com ciclos iterativos. A cada *Sprint* pode ser realizada uma avaliação das funcionalidades implementadas. Com esta prática, o escopo das avaliações fica menor, assim como a duração da sessão de avaliação. Os problemas de usabilidade podem ser corrigidos mais cedo, diminuindo o custo das correções e aumentando a satisfação do usuário.

5 CONCLUSÕES

Neste trabalho foram abordadas duas modalidades de avaliação heurística aplicadas a uma pequena empresa de desenvolvimento de software que utiliza o método ágil *Scrum*. O objetivo do trabalho foi analisar a utilização da avaliação heurística tradicional e a avaliação heurística colaborativa no contexto da empresa e as implicações do uso das duas versões da avaliação heurística em uma organização que utiliza métodos ágeis.

Para atingir os objetivos especificados foi efetuado um estudo de caso com duas etapas: na primeira etapa foi realizado treinamento com a equipe da empresa para que todos os participantes do estudo tivessem um nível mínimo de conhecimento sobre usabilidade e avaliação heurística. Na segunda etapa, foi feita uma análise da aplicação das duas versões da avaliação heurística, dividindo a equipe em duas. As duas equipes avaliaram o mesmo produto: um sistema Web desenvolvido pela empresa. Uma das equipes realizou a avaliação heurística tradicional e a outra realizou a avaliação heurística colaborativa. Após a realização das avaliações foram analisados os resultados dos procedimentos.

Este estudo de caso contribuiu para avaliar a utilização de métodos de inspeção de usabilidade no contexto de uma *startup*. Os resultados obtidos podem ser utilizados como ponto de partida para a escolha de um método de inspeção de usabilidade adequado para empresas similares, e em particular para aquelas que também utilizem métodos ágeis de desenvolvimento de software.

Os resultados do estudo mostraram que a avaliação tradicional encontrou mais problemas que a avaliação colaborativa. Também foi observado que a avaliação tradicional teve maior cobertura em relação aos tipos de heurísticas apontados pelos avaliadores do que a avaliação colaborativa, os problemas encontrados violavam maior número de heurísticas do que na avaliação colaborativa.

A média das severidades atribuídas pelos avaliadores na avaliação heurística colaborativa foi maior que as severidades atribuídas na avaliação tradicional.

Isto sugere que a avaliação colaborativa tende a revelar mais problemas sérios do que a avaliação tradicional.

A sessão de consolidação de resultados da avaliação heurística tradicional durou uma hora e vinte e dois minutos. Considerando que as sessões de avaliação tiveram tempo de uma hora e trinta minutos, a avaliação tradicional levou tempo consideravelmente maior do que a avaliação colaborativa.

Foi observado que a avaliação heurística colaborativa tem pontos em comum com os princípios ágeis de desenvolvimento. A avaliação colaborativa auxilia o compartilhamento de informação entre os participantes, contribuindo para o aumento dos conhecimentos dos iniciantes. Os participantes da avaliação heurística colaborativa consideraram o método compatível com as práticas da empresa.

Dentre os principais fatos observados nesse estudo de caso, destacam-se os seguintes: a avaliação heurística tradicional pode encontrar mais problemas de usabilidade que a avaliação colaborativa, dependendo das pessoas alocadas para realizar cada tipo de avaliação; a avaliação colaborativa tende à encontrar problemas de usabilidade mais sérios do que a avaliação tradicional; a avaliação tradicional toma tempo significativamente maior do que a avaliação colaborativa e a avaliação colaborativa apresentou características que permitem que se encaixe melhor em ambientes onde são utilizados métodos ágeis.

Para explorar em mais profundidade as questões tratadas nessa monografia, trabalhos futuros deverão investigar a produtividade das duas modalidades de avaliação heurística com a participação de avaliadores experientes, analisar se realmente a avaliação tradicional tem maior cobertura de heurísticas e a relação entre a modalidade de avaliação e a dificuldade encontrada para atribuir graus de severidade à problemas de usabilidade. O levantamento de questões de pesquisa referentes aos fatos observados nesse estudo de caso também constituem uma importante contribuição deste trabalho. Serão necessários estudos empíricos mais detalhados com vistas a investigar cada uma dessas questões de pesquisa individualmente.

REFERÊNCIAS BIBLIOGRÁFICAS

AMBLER, S. W. *Agile Adoption Rate Survey Results: February 2008*. February 2008. Disponível em: <<http://www.ambysoft.com/surveys/agileFebruary2008-.html>>.

BABAJO, A. *The effectiveness of Collaborative Heuristic Evaluation*. Dissertação (Mestrado) — University of York, 2012.

BARBOSA, D. F.; FURTADO, E. S.; GOMES, A. S. Uma estratégia de apoio á institucionalização da usabilidade em ambientes de desenvolvimento ágil. Sociedade Brasileira de Computação, Porto Alegre, Brazil, Brazil, p. 214–223, 2008. Disponível em: <<http://dl.acm.org/citation.cfm?id=1497470-.1497494>>.

BARBOSA, S. D. J.; SILVA, B. S. da. *Interação Humano-computador*. Elsevier Brasil, 2010. ISBN 9788535211207. Disponível em: <<http://books.google.com-.br/books?id=qk0skwr\cewC>>.

BECK, K. *Extreme Programming Explained*. [S.l.]: Addison Wesley, 2000.

BECK, K.; BEEDLE, M.; BENNEKUM, A. van; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R. C.; MELLOR, S.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. *Manifesto for Agile Development*. 2001. Disponível em: <<http://agilemanifesto.org/>>.

BENTO, L. F. H.; PRATES, R. O.; CHAIMOWICZ, L. Using semiotic inspection method to evaluate a human-robot interface. *2009 Latin American Web Congress*, p. 77–84, 2009.

BUYKX, L. *Improving heuristic evaluation through collaborative working*. Dissertação (Mestrado) — University of York, 2009.

COCKBURN, A. *Crystal clear a human-powered methodology for small teams*. First. [S.l.]: Addison-Wesley Professional, 2004. ISBN 0201699478.

CRESPO, A. N.; SILVA, O. J. da; BORGES, C. A.; SALVIANO, C. F.; JUNIOR, M. de T. A.; JINO, M. Uma metodologia para teste de software no contexto da melhoria de processo. *III Simpósio Brasileiro de Qualidade de Software*, p. 1–15, 2004.

DANINO, N. *Heuristic Evaluation - a Step By Step Guide Article*. September 2001. Disponível em: <<http://www.sitepoint.com/heuristic-evaluation-guide/>>.

FOWLER, M. *The New Methodology*. 2005. Disponível em: <<http://martinfowler.com/articles/newMethodology.html>>.

GUTWIN, C.; GREENBERG, S. The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. *IEEE Computer Society*, Washington, DC, USA, p. 98–103, 2000. Disponível em: <<http://dl.acm.org/citation.cfm?id=647068.715651>>.

HIGHSMITH III, J. A. *Adaptive software development: a collaborative approach to managing complex systems*. New York, NY, USA: Dorset House Publishing Co., Inc., 2000. ISBN 0-932633-40-4.

HIGHSMITH, J. *Agile Software Development Ecosystems*. [S.l.]: Addison Wesley, 2002.

HOLZINGER, A. Usability engineering methods for software developers. *Commun. ACM*, ACM, New York, NY, USA, v. 48, n. 1, p. 71–74, jan. 2005. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/1039539.1039541>>.

HORNBAEK, K.; FRØKJÆR, E. A study of the evaluator effect in usability testing. *Human-Computer Interaction*, Taylor & Francis, v. 23, n. 3, p. 251–277, 2008. Disponível em: <<http://dx.doi.org/10.1080/07370020802278205>>.

ISO. *ISO 9241-11: Ergonomic requirements for office work with visual display terminals*. [S.l.: s.n.], 2002.

ISO/IEC. *ISO/IEC 25000 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*. [S.l.], 2005.

LEE, J. C.; JUDGE, T. K.; MCCRICKARD, D. S. Evaluating extreme scenario-based design in a distributed agile team. ACM, New York, NY, USA, p. 863–877, 2011. Disponível em: <<http://doi.acm.org/10.1145/1979742.1979681>>.

MEMMEL, T.; GUNDELSWEILER, F.; REITERER, H. Agile human-centered software engineering. British Computer Society, Swinton, UK, UK, p. 167–175, 2007. Disponível em: <<http://dl.acm.org/citation.cfm?id=1531294.1531317>>.

NAJAFI, M.; TOYOSHIBA, L. Two case studies of user experience design and agile development. IEEE Computer Society, Washington, DC, USA, p. 531–536, 2008. Disponível em: <<http://dx.doi.org/10.1109/Agile.2008.67>>.

NASCIMENTO, G. V. *Um modelo de referência para o desenvolvimento ágil de software*. Dissertação (Mestrado) — Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, São Carlos, 2007. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-07052008-170413>>.

NERUR, S.; MAHAPATRA, R.; MANGALARAJ, G. Challenges of migrating to agile methodologies. *Communications of the ACM*, v. 48, p. 72–78, 2005.

NIELSEN, J. Enhancing the explanatory power of usability heuristics. ACM, New York, NY, USA, p. 210–, 1994. Disponível em: <<http://doi.acm.org/10.1145/259963.260333>>.

NIELSEN, J. *Why You Only Need to Test With 5 Users*. March 2000. Disponível em: <<http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>>.

NIELSEN, J.; MACK, R. L. *Usability inspection methods*. [S.l.]: Wiley & Sons, 1994.

NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. ACM, New York, NY, USA, p. 249–256, 1990. Disponível em: <<http://doi.acm.org/10.1145/97243.97281>>.

NIELSEN, L.; MADSEN, S. The usability expert's fear of agility: an empirical study of global trends and emerging practices. ACM, New York, NY, USA, p. 261–264, 2012. Disponível em: <<http://doi.acm.org/10.1145/2399016.2399057>>.

PALMER, S. R.; FELSING, M. *A Practical Guide to Feature-Driven Development*. 1st. ed. [S.l.]: Pearson Education, 2001. ISBN 0130676152.

PETRIE, H.; POWER, C. What do users really care about?: a comparison of usability problems found by users and experts on highly interactive websites. ACM, New York, NY, USA, p. 2107–2116, 2012. Disponível em: <<http://doi.acm.org/10.1145/2208276.2208363>>.

POPPENDIECK, M.; POPPENDIECK, T. *Lean Software Development: An Agile Toolkit*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003. ISBN 0321150783.

RAMPAZZO, L. *Metodologia científica para alunos dos cursos de graduação e pós graduação*. [S.l.]: Edições Loyola, 2002.

RIEMAN, J.; FRANZKE, M.; REDMILES, D. Usability evaluation with the cognitive walkthrough. ACM, New York, NY, USA, p. 387–388, 1995. Disponível em: <<http://doi.acm.org/10.1145/223355.223735>>.

SEFFAH, A.; METZKER, E. The obstacles and myths of usability and software engineering. *Commun. ACM*, ACM, New York, NY, USA, v. 47, n. 12, p. 71–76,

dez. 2004. ISSN 0001-0782. Disponível em: <[http://doi.acm.org/10.1145-1035134.1035136](http://doi.acm.org/10.1145/1035134.1035136)>.

SHORE, J.; WARDEN, S. *The Art of Agile Development*. [S.l.]: O'Reilly Media, 2008.

SOARES, M. S. Comparação entre metodologias ágeis e tradicionais de desenvolvimento de software. *Infocomp Journal of Computer Science*, p. 8–13, 2004.

STAPLETON, J. *Dynamic Systems Development Method: The Method in Practice*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997. ISBN 0201178893.

TANIGUCHI, K.; CORREA, F. E. Metodologias ágeis e motivação de pessoas em projetos de desenvolvimento de software. *Revista de Ciências Exatas e Tecnologia*, p. 163–179, 2009.

WINTER, J.; RÖNKKÖ, K.; AHLBERG, M.; HOTCHKISS, J. Meeting organisational needs and quality assurance through balancing agile and formal usability testing results. Springer-Verlag, Berlin, Heidelberg, p. 275–289, 2011. Disponível em: <<http://dl.acm.org/citation.cfm?id=2040660.2040689>>.

WOLKERSTORFER, P.; TSCHELIGI, M.; SEFELIN, R.; MILCHRAHM, H.; HUSSAIN, Z.; LECHNER, M.; SHAHZAD, S. Probing an agile usability process. ACM, New York, NY, USA, p. 2151–2158, 2008. Disponível em: <<http://doi.acm.org/10.1145/1358628.1358648>>.

YIN, R. K. *Applications of Case Study Research*. [S.l.]: SAGE Publications, Inc, 2003.

YIN, R. K. *Case Study Research: Design and Methods, 4th Edition*. 4th. ed. [S.l.]: SAGE Publications, Inc, 2009. ISBN 0761925538.

A HEURÍSTICAS UTILIZADAS PARA AS AVALIAÇÕES

Heurísticas para sistemas Web de Petrie e Power

Helen Petrie and Christopher Power. 2012. What do users really care about?: a comparison of usability problems found by users and experts on highly interactive websites. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2107-2116.

Apresentação física

Heurística 1. Faça com que texto e elementos interativos sejam suficientemente grandes e claros
O tamanho em que o texto e elementos interativos são mostrados por padrão (sem modificações pelo usuário) devem ter tamanho grande o suficiente para que usuários possam ler e manipular com facilidade.

Heurística 2. Torne o leiaute da página claro

Faça com que o leiaute das informações em cada página seja claro, fácil de ler e que reflita a organização do material disponibilizado.

Heurística 3. Evite que o tempo de exibição de informações ou para completar tarefas seja curto demais

Permita que o usuário tenha tempo suficiente para completar suas tarefas confortavelmente, e se alguma informação for exibida por tempo limitado, que seja mostrada por tempo suficientemente longo.

Heurística 4. Faça com que os principais conteúdos da página e possíveis mudanças no conteúdo estejam destacados

Faça com que os principais conteúdos e elementos interativos estejam claramente visíveis na página, e que quaisquer mudanças que ocorram na página sejam claramente indicadas.

Conteúdo

Heurística 5. Forneça conteúdo relevante e apropriado para o site

Garanta que o conteúdo mostrado na página seja relevante para as tarefas dos usuários e que seja escrito de maneira apropriada.

Heurística 6. Forneça conteúdo em quantidade suficiente, mas não excessiva

Forneça conteúdo (incluindo de ajuda) que auxilie os usuários a completarem suas tarefas, mas não em quantidades excessivas que possam sobrecarregar o usuário.

Heurística 7. Utilize termos e abreviações claras, e evite o uso de jargões e palavras de difícil entendimento

Defina todos os termos complexos, jargões e explique todas as abreviaturas.

Arquitetura da Informação

Heurística 8. Forneça estruturas de organização de informação claras e bem organizadas

Forneça estruturas de informação que organizem o conteúdo da página de maneira adequada e que auxiliem os usuários a completarem suas tarefas.

Interatividade

Heurística 9. “Como e porque”

Forneça para os usuários explicações claras sobre como a interação do sistema funciona e o motivo porque as coisas acontecem no sistema.

Heurística 10. Instruções e rótulos claros

Forneça instruções e rótulos de campos claros para todos os elementos interativos. Utilize convenções comumente utilizadas na Web (como o uso de asterisco para elementos de preenchimento obrigatório)

Heurística 11. Evite esforço excessivo e repetição de ações pelos usuários

Não peça para o usuário entrar com a mesma informação mais de uma vez, e não exija que o usuário precise de esforço excessivo para atingir objetivos que poderiam ser desempenhados de maneira mais eficiente pelo sistema.

Heurística 12. Faça com que os formatos de entrada de dados sejam claros e fáceis

Torne claro de antemão qual é o formato exigido para entrada de dados pelos usuários. Utilize formatos que sejam fáceis para os usuários.

Heurística 13. Forneça feedback para ações realizadas pelo usuário e para demonstrar o estado do sistema

Forneça feedback para as ações feitas pelo usuário, e indicações indicando se alguma ação do sistema está em curso ou se vai levar algum tempo para ser processada.

Heurística 14. Faça com que a sequência de interação seja lógica

Faça com que a sequência da interação seja lógica para os usuários (por exemplo, usuários que utilizam idiomas indo-europeus, tais como o Português, tendem a ler uma página do canto esquerdo-superior para o canto inferior-direito, de forma que um botão "Próximo" seria mais lógico se estiver no canto inferior-direito").

Heurística 15. Forneça um conjunto de opções lógico e completo

Garanta que qualquer conjunto de opções dado em operações no sistema inclua todas as opções que os usuários necessitem e que o conjunto seja organizado de forma lógica para os usuários.

Heurística 16. Siga convenções para interação

A não ser que haja algum motivo especial, siga convenções utilizadas na Web para interação (por exemplo, uma sequência lógica entre a navegação por elementos quando se pressiona o TAB)

Heurística 17. Forneça as funcionalidades interativas que os usuários precisam e esperam do sistema

Forneça todas as funcionalidades interativas que os usuários necessitem para efetuar suas tarefas e que eles esperam encontrar em determinadas situações (por exemplo, o sistema fornece uma busca?)

Heurística 18. Indique se os links vão para outra página ou para outro website externo

Se um link aponta para outro website ou para outro tipo de recurso (como um arquivo PDF), indique isso claramente na interface.

Heurística 19. Elementos interativos e não-interativos devem ser fáceis de distinguir

Elementos que fornecem algum tipo de interatividade devem ser claramente indicados como tal, e elementos que não fornecem interatividade não devem utilizar leiaute que possam sugerir que tem alguma funcionalidade.

Heurística 20. Agrupe elementos interativos de forma lógica e clara

Agrupe elementos interativos e seus rótulos e textos associados de forma que seja claro identificar qual é a sua funcionalidade.

Heurística 21. Força mensagens de erros informativas e que auxiliem os usuários a se recuperar dos erros

Forneça mensagens de erro que explique os problemas utilizando a linguagem do usuário e que forneça informações que auxiliem os usuários a se recuperar dos erros.

Graus de severidade:

0 = **Não é um problema:** não concordo que esse seja um problema de usabilidade

1 = **Cosmético:** não precisa ser consertado a não ser que haja tempo extra disponível no projeto

2 = **Simples:** a correção deste tipo de problema deve ter prioridade menor

3 = **Sério:** importante corrigir o problema, por isso dar prioridade maior a ele

4 = **Catástrofe:** o problema deve ser resolvido impreterivelmente antes do lançamento do produto

B.2 Planilha das severidades dos problemas

Lista individual de atribuição de grau de severidade

Aplicação:	
Tarefa:	
Avaliador:	

Nº do problema	Severidade
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	

C QUESTIONÁRIOS UTILIZADOS

C.1 Questionário demográfico

Questionário Demográfico

1. Cargo na empresa:
2. Idade:
3. Sexo: () Masculino
() Feminino
4. Qual a sua formação acadêmica?
5. Qual a sua experiência na indústria?
6. Qual a sua experiência com usabilidade?
7. Qual a sua experiência com teste?
8. Você já teve contato com avaliação heurística? Se sim, como foi este contato e de quantas avaliações já participou?
9. Qual a sua experiência com teste com usuários?
10. Qual a sua experiência com elicitação de requisitos?

C.2 Questionário da avaliação tradicional

Avaliação Tradicional

1. Você encontrou dificuldades para entrar em acordo com o grupo sobre quais problemas encontrados realmente caracterizavam problemas?
2. Você encontrou dificuldade para identificar problemas iguais descritos de forma diferente por diferentes avaliadores?
3. Você encontrou dificuldade de entrar em acordo sobre os graus de severidade?
4. Você teve dificuldade para entender as heurísticas?
5. E o esquema de severidade?
6. Qual a sua opinião sobre a adequação do método no esquema de trabalho da empresa?

