



Joelma Choma

SCRUMUX: UMA ABORDAGEM PARA INTEGRAR DESIGN DE  
INTERAÇÃO DO USUÁRIO AO PROCESSO SCRUM.

Sorocaba  
2015

Universidade Federal de São Carlos  
Programa de Pós-Graduação em Ciência da Computação (PPGCCS)

Joelma Choma

SCRUMUX: UMA ABORDAGEM PARA INTEGRAR DESIGN DE INTERAÇÃO DO USUÁRIO AO  
PROCESSO SCRUM

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCCS) da Universidade Federal de São Carlos como parte dos requisitos exigidos para a obtenção do título de Mestre em Ciência da Computação área de concentração: Engenharia de Software e Gestão do Conhecimento.

Orientadora: Profa. Dra. Luciana Aparecida Martinez Zaina

Este exemplar corresponde à versão final da dissertação defendida pela aluna, e orientada pela Prof<sup>a</sup>. Dra. Luciana Aparecida Martinez Zaina

---

Sorocaba  
2015

Choma, Joelma

ScrumUX: Uma Abordagem para Integrar Design de Interação do Usuário  
ao Processo Scrum / Joelma Choma. -- 2015.  
144 f. : 30 cm.

Dissertação (mestrado)-Universidade Federal de São Carlos, campus  
Sorocaba, Sorocaba

Orientador: Luciana Aparecida Martinez Zaina

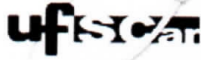
Banca examinadora: Tiago Silva da Silva, Alexandre Álvaro

Bibliografia

1. Design centrado no usuário. 2. Design de interação. 3. Scrum. I.  
Orientador. II. Sorocaba-Universidade Federal de São Carlos. III. Título.

Ficha catalográfica elaborada pelo Programa de Geração Automática da Biblioteca campus Sorocaba (B-So).

DADOS FORNECIDOS PELO(A) AUTOR(A)



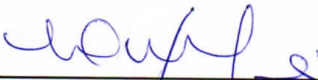
**UNIVERSIDADE FEDERAL DE SÃO CARLOS**  
Centro de Ciências em Gestão e Tecnologia  
Programa de Pós-Graduação em Ciência da Computação

---

**Folha de Aprovação**

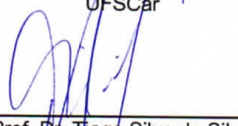
---

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado da candidata Joelma Choma, realizada em 10/11/2015:



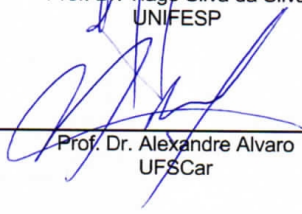
---

Prof. Dra. Luciana Aparecida Martinez Zaina  
UFSCar



---

Prof. Dr. Tiago Silva da Silva  
UNIFESP



---

Prof. Dr. Alexandre Alvaro  
UFSCar

À MINHA QUERIDA FAMÍLIA.  
AOS MEUS QUERIDOS AMIGOS.

# Agradecimentos

Agradeço,

A Deus por conduzir minha vida e conceder-me Sabedoria, Alegria e Fé.

Ao meu querido Hélio pelo carinho, incentivo e apoio em todos os momentos.

A minha família e amigos que há muito tempo acompanham a minha caminhada. Em especial, a minha querida irmã Jane e meus sobrinhos Davi e Daniel que testemunharam minha determinação e dedicação.

A minha querida orientadora Professora Dra. Luciana Zaina pela confiança, motivação, inspiração e oportunidade de trabalhar em seus projetos.

Ao Professor Dr. Alexandre Álvaro pelo apoio e pelas valiosas contribuições durante o meu projeto de pesquisa.

Ao Professor Dr. Tiago Silva da Silva por ter aceitado prontamente o convite para participar da banca examinadora, contribuindo com sua experiência para o enriquecimento deste trabalho.

Aos demais professores da UFSCar que também contribuíram para a minha formação compartilhando seus conhecimentos.

A empresa Mega Sistemas pelo apoio financeiro.

A diretoria e colaboradores da Mega Sistemas por toda receptividade e atenção dispensada.

Ao meus colegas Diego Quintale e Daniela Beraldo por suas contribuições e pelo prazer de trabalharmos juntos.

Aos alunos do curso de Ciência da Computação da UFSCar que participaram do projeto na fase de validação da proposta.

A UFSCar e seus colaboradores pela estrutura e serviços oferecidos durante o curso.

A todos que direta ou indiretamente fizeram parte dessa jornada.

Ao meu Anjo da Guarda pela companhia e por todas as aventuras!

*“A lei da mente é implacável.  
O que você pensa, você cria;  
O que você sente, você atrai;  
O que você acredita  
Torna-se realidade.”*

*(Buda)*

# Resumo

Princípios do desenvolvimento ágil asseguram que a atenção contínua à excelência técnica aliada à um bom *design* contribui para o aumento de agilidade garantindo produtos com maior qualidade. A usabilidade é um dos atributos da qualidade, e está relacionada com tornar a experiência do usuário agradável por meio de um *design* de interface eficiente e intuitivo. Métodos ágeis estão comprometidos em satisfazer os clientes entregando software de valor. Mas, por vezes, acabam entregando somente a utilidade, sem cobrir os aspectos de usabilidade. Projetar soluções com maior usabilidade e mais apropriadas ao contexto dos usuários é a principal preocupação do Design de Interação (DxI). O DxI fundamenta-se no *design* centrado no usuário, que leva em conta as necessidades e desejos dos usuários para promover a mais agradável experiência de interação. O interesse em integrar práticas de *design* centrado no usuário com metodologias ágeis tem aumentado nos últimos anos. Contudo, um número bem limitado de contribuições têm abordado a integração em um nível mais abrangente, reunindo práticas e fornecendo orientações para a organização do trabalho dentro de um roteiro de desenvolvimento de software. Esta dissertação apresenta uma abordagem para integrar o Design de Interação do usuário ao processo Scrum por meio de atividades e artefatos. Na primeira etapa do estudo, realizou-se um estudo bibliográfico para verificar as possíveis estratégias para a integração do DxI com o processo Scrum, e em paralelo uma abordagem de pesquisa-ação foi aplicada, junto à profissionais da indústria de desenvolvimento de software, para definir e avaliar os artefatos que dariam suporte à abordagem de integração ScrumUX. Na segunda etapa, um estudo experimental foi conduzido com objetivo de avaliar a proposta de integração ScrumUX em relação ao Scrum tradicional. O experimento foi realizado em contexto acadêmico, onde dezenove alunos divididos em cinco equipes atuaram em projetos de desenvolvimento de software no domínio de sistemas ERP. Os testes estatísticos confirmaram uma diferença significativa entre os grupos experimentais e de controle, apontando que as equipes que utilizaram a abordagem ScrumUX produziram softwares com melhor grau de usabilidade na perspectiva de suporte à tarefas, considerando o ponto de vista de usuários de sistemas ERP.

Palavras-chave: Engenharia de Software. IHC. Metodologia Ágil. Scrum. Design Centrado no Usuário. Experiência do Usuário. Design de Interação. ERP.



# Abstract

Agile development principles ensure that continuous attention to technical excellence combined with a good design contributes to increased agility ensuring products with higher quality. Usability is one of the quality attributes, and is related to making the experience enjoyable user through an efficient and intuitive interface design. Agile methods are committed to deliver valuable software to the satisfaction of the customer. At times, they deliver only the utility, without covering aspects of usability. Design solutions providing greater usability and more appropriate to the users' context is the main concern of the Interaction Design (IxD). The IxD is based on user-centered design, which takes into account the users needs and wants, aiming to promote interaction more enjoyable experience. In recent years, the attention in integrating user-centered design practices into agile methodologies has increased. However, a very limited number of contributions have addressed the integration into a broader level, gathering practices and providing guidance to the organization of work within a software development roadmap. This dissertation presents an approach to integrate user interaction design into Scrum process through activities and artifacts. In the first stage of the study, a literature study was performed to determine possible strategies for integrating IxD into Scrum process. Furthermore, an action research approach was applied to define and evaluate the artifacts that would support the ScrumUX integration approach, closely with the professionals in the software development industry. In the second stage, an experimental study was carried out to evaluate the proposed ScrumUX integration compared to traditional Scrum approach. The experiment was carried out in an academic context, in which nineteen students divided into five teams worked in software development projects, in the ERP systems domain. Statistical tests confirmed a significant difference between the experimental and control groups, noting that the teams who used the ScrumUX approach produced software with better degree of usability in the task support perspective, considering ERP systems' users opinion.

Key-words: Software Engineering. HCI. Agile Methodology. Scrum. User-Centered Design. User Experience. Interaction Design. ERP.

# Lista de Figuras

1.1	Visão geral da metodologia utilizada . . . . .	5
2.1	Abrangência de UX e as disciplinas em torno do DxI. Adaptado de Saffer (2009)	8
2.2	Ciclo de vida de DxI em conformidade com as atividades do DCU . . . . .	9
2.3	Manifesto Ágil. Fonte: Beck et al. (2001) . . . . .	16
2.4	Metodologias ágeis utilizadas. Fonte: VersionOne (2015) . . . . .	18
2.5	Artefatos e cerimônias do Scrum. Adaptado de Schwaber (2004) . . . . .	19
2.6	User Story (COHN, 2009) . . . . .	21
3.1	Esquema <i>snowballing</i> . . . . .	25
3.2	Número de artigos sobre integração do DCU em metodologias ágeis por ano. . .	28
4.1	Ciclos do DSR Fonte: Hevner (2007) . . . . .	41
4.2	SoftCoDeR. Fonte: Choma, Zaina e Silva (2015) . . . . .	43
4.3	Modelo de Lean Persona . . . . .	49
4.4	Abordagem para conduzir teste com usuários e reportar soluções/recomendações de usabilidade. Fonte: Choma, Zaina e Beraldo (2015) . . . . .	51
4.5	Exemplo Lean Persona desenvolvida pelo <i>designer</i> de UX . . . . .	52
4.6	Exemplo de recomendação para um dos problemas identificados utilizando o Protocolo de Comunicação . . . . .	52
4.7	UserX Story - primeira versão . . . . .	54
4.8	UserX Story - versão refinada . . . . .	54
4.9	Exemplo de UserX Story desenvolvida por um PO . . . . .	56
4.10	Abordagem ScrumUX . . . . .	60
5.1	Heurísticas violadas por perspectiva de apresentação . . . . .	81
5.2	Heurísticas violadas por perspectiva de suporte à tarefas . . . . .	82
5.3	Gravidade das violações . . . . .	82
5.4	Nível de gravidade . . . . .	85
5.5	Índice de usabilidade geral (Equação 5.1) . . . . .	86
5.6	Índice de usabilidade geral (Equação 5.2) . . . . .	86
H.1	Lean Persona (Equipe 1) . . . . .	126

H.2	Persona (Equipe 3) . . . . .	127
I.1	Cenário (Equipe 1) . . . . .	128
I.2	Cenário (Equipe 3) . . . . .	128
J.1	Protótipo de baixa utilizando a ferramenta Balsamiq (Equipe 1) . . . . .	129
J.2	Protótipo de baixa - <i>sketches</i> (Equipe 2) . . . . .	130
J.3	Protótipos de baixa utilizando a ferramenta Balsamiq (Equipe 3) . . . . .	130
K.1	Itens do protocolo gerados na ferramenta UX Designer (Equipe 1) . . . . .	131
L.1	UserX Story - US expandidas na ferramenta Agile Wrap (Equipe 1) . . . . .	132

# Lista de Tabelas

2.1	Técnicas para o <i>design</i> centrado no usuário . . . . .	10
3.1	Artigos por tipo de estratégias . . . . .	29
4.1	Perfil dos entrevistados . . . . .	47
4.2	Visão geral dos comentários dos participante por aspecto observado . . . . .	48
4.3	Perfil dos participantes . . . . .	55
4.4	Principais papeis da abordagem ScrumUX . . . . .	59
4.5	Artefatos de <i>design</i> recomendados na abordagem ScrumUX . . . . .	61
5.1	Heurísticas por perspectivas de sistemas ERP . . . . .	67
5.2	<i>Scores</i> para a escala <i>Likert</i> . . . . .	67
5.3	Grau de importância por Perspectiva de Apresentação. . . . .	68
5.4	Grau de importância por Perspectiva de Suporte à Tarefas . . . . .	68
5.5	Desenho experimental . . . . .	71
5.6	Materiais usados pelos grupos experimental e de controle . . . . .	73
5.7	Cronograma do projeto integrado (PDS-IHC) . . . . .	74
5.8	Pesquisa do domínio e público alvo . . . . .	75
5.9	Análise comparativa de produtos similares . . . . .	75
5.10	Personas e cenários de interação . . . . .	77
5.11	User Stories e protótipos . . . . .	77
5.12	Atividades de implementação . . . . .	78
5.13	Atividades de verificação . . . . .	79
5.14	Gravidade das violações heurísticas . . . . .	83
5.15	Índice de Usabilidade (Equação 5.1) . . . . .	84
5.16	Índice de Usabilidade (Equação 5.2) . . . . .	84
5.17	Avaliação qualitativa . . . . .	87

# Lista de Abreviaturas e Siglas

AR	Action Research
BDUF	Big Design Up Front
CMD	Cooperative Method Development
DAS	Desenvolvimento Ágil de Software
DCU	Design Centrado no Usuário
DS	Design Science
DSR	Design Science Research
DxI	Design de Interação
ED	Equipe de Desenvolvimento
ERP	Enterprise Resource Planning
ES	Engenharia de Software
HP	Hipótese de Persona
IHC	Interação Humano-Computador
LDUF	Little Design Up Front
MSL	Mapeamento Sistemática de Literatura
PDS	Projeto de Desenvolvimento de Software
PO	Product Owner
RSL	Revisão Sistemática de Literatura
SI	Sistema de Informação
SM	Scrum Master

SoftCoDeR Software Cooperative Design Research

UI<sub>ERP</sub> ERP Usability Index

US User Story

UX Design de Interação

UXD User eXperience Design

# Sumário

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Motivação e problema . . . . .	2
1.2	Objetivos . . . . .	3
1.2.1	Objetivo geral . . . . .	3
1.2.2	Objetivos específicos . . . . .	3
1.3	Metodologia e organização . . . . .	4
1.4	Contribuições e resultados obtidos . . . . .	4
1.5	Organização do trabalho . . . . .	5
<b>2</b>	<b>Fundamentos e técnicas</b>	<b>7</b>
2.1	Considerações iniciais . . . . .	7
2.2	Design de Interação (DxI) . . . . .	7
2.2.1	Ciclo de vida do DxI . . . . .	9
2.2.2	Técnicas para o design centrado no usuário . . . . .	10
2.2.3	Usabilidade . . . . .	13
2.3	Desenvolvimento Ágil de Software . . . . .	15
2.3.1	Scrum . . . . .	17
2.3.1.1	Papéis, cerimônias e artefatos do Scrum . . . . .	19
2.3.2	User Story . . . . .	20
2.4	Considerações finais . . . . .	21
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>22</b>
3.1	Considerações iniciais . . . . .	22
3.2	Estudos sistemáticos de literatura . . . . .	22
3.3	Terminologias . . . . .	22
3.4	Revisão Sistemática de Literatura . . . . .	23
3.4.1	Atualização da revisão de literatura . . . . .	24
3.4.2	Análise dos artigos selecionados . . . . .	28
3.4.2.1	Práticas . . . . .	29
3.4.2.2	Profissionais . . . . .	31
3.4.2.3	Processos . . . . .	33

3.4.2.4	<i>Frameworks</i> . . . . .	35
3.5	Considerações Finais . . . . .	38
<b>4</b>	<b>ScrumUX: da pesquisa-ação para a abordagem proposta</b>	<b>39</b>
4.1	Considerações iniciais . . . . .	39
4.2	Metodologias de Pesquisa . . . . .	39
4.2.1	Pesquisa-Ação e o Método Cooperativo de Desenvolvimento . . . . .	40
4.2.2	<i>Design Science Research</i> . . . . .	40
4.2.3	Trabalhos Relacionados . . . . .	41
4.3	SoftCoDeR: a abordagem de pesquisa . . . . .	42
4.4	Construção e avaliação dos artefatos de apoio . . . . .	44
4.4.1	Primeiro ciclo . . . . .	45
4.4.1.1	Entendendo a prática . . . . .	45
4.4.1.2	Deliberando melhorias e construindo artefatos . . . . .	47
4.4.1.3	Implementando, observando as melhorias e avaliando artefatos . . . . .	49
4.4.2	Segundo ciclo . . . . .	53
4.4.2.1	Entendendo a prática . . . . .	53
4.4.2.2	Deliberando melhorias e construindo artefatos . . . . .	53
4.4.2.3	Implementando, observando as melhorias e avaliando artefatos . . . . .	55
4.4.3	Resultados . . . . .	56
4.5	ScrumUX . . . . .	57
4.6	Análise comparativa . . . . .	63
4.7	Considerações finais . . . . .	64
<b>5</b>	<b>Validação da abordagem ScrumUX: um estudo experimental</b>	<b>65</b>
5.1	Considerações iniciais . . . . .	65
5.2	Experimentação em Engenharia de Software . . . . .	65
5.3	Escopo do estudo experimental . . . . .	66
5.4	Métrica de Usabilidade para Sistemas ERP . . . . .	66
5.5	Desenvolvimento da experimentação . . . . .	70
5.5.1	Planejamento do experimento . . . . .	70
5.5.1.1	Execução do experimento . . . . .	72
5.5.1.2	Análise e interpretação dos resultados . . . . .	80
5.5.1.3	Ameaças à validade do estudo . . . . .	87
5.6	Considerações finais . . . . .	89
<b>6</b>	<b>Conclusões e trabalhos futuros</b>	<b>90</b>
6.1	Limitações e trabalhos futuros . . . . .	92
6.2	Publicação e submissões . . . . .	92
	<b>Referências</b>	<b>93</b>
	<b>Apêndices</b>	<b>103</b>



<b>A</b>	<b>Revisão de Literatura</b>	<b>104</b>
A.1	Lista de artigos . . . . .	104
<b>B</b>	<b>Estudo experimental: Cenário</b>	<b>110</b>
B.1	Cenário . . . . .	110
B.1.1	Descrição do sistema . . . . .	110
B.1.2	Requisitos do sistema . . . . .	111
B.1.3	Glossário . . . . .	111
<b>C</b>	<b>Estudo experimental: planilhas de inspeção</b>	<b>112</b>
C.1	Grupo Experimental . . . . .	112
C.2	Grupo de Controle . . . . .	113
<b>D</b>	<b>Estudo experimental: heurísticas PERP</b>	<b>114</b>
<b>E</b>	<b>Tutorial: Agile Wrap</b>	<b>115</b>
E.1	Cadastrar equipe . . . . .	115
E.2	Cadastrar e planejar projeto . . . . .	116
E.3	Criar user stories . . . . .	117
E.4	Criar tarefas . . . . .	117
E.5	Acompanhar projeto: quadro de tarefas . . . . .	118
<b>F</b>	<b>Tutorial: UX Designer</b>	<b>119</b>
F.1	Criar projeto . . . . .	119
F.2	Inserir protótipos e criar cenários . . . . .	120
F.3	Criar personas . . . . .	121
F.4	Gerar Protocolo de Comunicação . . . . .	122
<b>G</b>	<b>Planilhas de acompanhamento do grupos</b>	<b>123</b>
G.1	Projeto E1 - Experimental . . . . .	123
G.2	Projeto E2 - Experimental . . . . .	123
G.3	Projeto E3 - Controle . . . . .	124
G.4	Projeto E4 - Controle . . . . .	124
G.5	Projeto E5 - Controle . . . . .	125
<b>H</b>	<b>Estudo experimental: personas</b>	<b>126</b>
<b>I</b>	<b>Estudo experimental: cenários</b>	<b>128</b>
<b>J</b>	<b>Estudo experimental: protótipos</b>	<b>129</b>
<b>K</b>	<b>Estudo experimental: Protocolo de Comunicação</b>	<b>131</b>
<b>L</b>	<b>Estudo experimental: UserX Story</b>	<b>132</b>
<b>M</b>	<b>Estudo experimental: histórias de usuário</b>	<b>133</b>

---

<b>N</b>	<b>Cálculo do Índice de Usabilidade</b>	<b>139</b>
N.1	Índice de usabilidade: Perspectiva Apresentação . . . . .	139
N.2	Índice de usabilidade: Perspectiva Suporte à Tarefas . . . . .	142

# Introdução

Metodologias ágeis seguem fortes princípios que almejam equilibrar custos e entregar software de alta qualidade rapidamente (HOUSTON, 2014). A agilidade dentro deste contexto está relacionada com a capacidade agir rápido para responder questões que demandam mudança. Isto pode significar aumento na velocidade de desenvolvimento, mas o seu forte foco na melhoria de processo de engenharia negligenciam, por vezes, questões importantes relacionadas à experiência do usuário sobre a usabilidade do produto (PATTON, 2002) (BEYER; HOLTZBLATT; BAKER, 2004). O resultado disso são produtos que atendem as necessidades, mas que, na maioria das vezes, não conseguem encantar seus usuários finais. Usabilidade é um importante atributo para se avaliar a qualidade do software, contudo deve ser considerada ao longo de todo processo de desenvolvimento (JOKELA, 2005).

Design Centrado no Usuário (DCU) é um termo amplo utilizado para descrever os processos de *design* em que os usuários finais influenciam a concepção e o desenvolvimento dos produtos (ABRAS; MALONEY-KRICHMAR; PREECE, 2004). DCU é uma filosofia que defende fortemente o envolvimento dos usuários durante o processo de *design* e desenvolvimento de software, seja durante a coleta dos requisitos e em testes de usabilidade, ou de maneira mais ativa como parceiros dos designers durante todo processo de *design* do produto. O pressuposto subjacente do DCU é que o tempo e custo de desenvolvimento incluindo sua aplicação serão diluídos pela qualidade da aplicação e a quantidade reduzida de retrabalho, uma vez que o envolvimento dos usuários assegura que o produto seja adequado para a finalidade pretendida no ambiente em que vai ser utilizado (MAO et al., 2005). O Design de Interação (DxI) emerge da perspectiva do DCU focando-se na concepção de produtos que venham a proporcionar uma boa Experiência do Usuário (UX - User Experience) por meio da simplicidade, elegância, eficiência, eficácia e satisfação emocional de uso (ROGERS; SHARP; PREECE, 2013).

## 1.1 Motivação e problema

As metodologias de desenvolvimento ágil de software possuem características que favorecem a incorporação do DxI, tais como processo iterativo, foco na qualidade, e colaboração com diferentes *stakeholders* (MEMMEL; GUNDELSWEILER; REITERER, 2007). Entretanto, assim como os métodos tradicionais, as metodologias DAS não costumam incorporar práticas de DxI

no processo de desenvolvimento, tão pouco fornecem orientações ou regras para como os papéis, processos de trabalho e práticas relacionadas à experiência do usuário devem ser integradas ou conduzidas (SEFFAH; GULLIKSEN; DESMARAIS, 2005).

O principal desafio consiste em saber qual é a melhor forma de incorporar práticas de DxD no desenvolvimento de software ágil sem contrariar princípios que são fundamentais para ambas as abordagens. Além disso, muitos outros fatores que podem dificultar a integração têm sido identificados, dentre os quais pode-se destacar: a falta de tempo para realizar atividades de pesquisa com usuário, a necessidade de adequar a granularidade do projeto de *design* conforme características do projeto ágil e a importância da definição dos papéis dos profissionais de UX junto à equipe ágil (FERREIRA; NOBLE; BIDDLE, 2007) (SY, 2007) (LIEVESLEY; YEE, 2006). Dentre os fatores que contribuem para a integração estão a necessidade de promover a sinergia e estabelecer uma boa comunicação entre as equipes ágil e de UX (BROWN; LINDGAARD; BIDDLE, 2011) (SALAH; PAIGE; CAIRNS, 2014b). Em particular, esta última questão envolve lidar com a diferença de mentalidade entre os desenvolvedores ágeis e *designers* de UX, que utilizam práticas diferentes e expressam em sua maneira particular de suas áreas de conhecimento.

## 1.2 Objetivos

### 1.2.1 Objetivo geral

O objetivo geral desse trabalho é propor uma abordagem para integrar o Design de Interação do usuário ao processo Scrum por meio de atividades e artefatos.

Para alcançar o objetivo as seguintes questões de pesquisa precisam ser respondidas:

- Q1. Quais estratégias são comumente empregadas para integrar DxD com metodologias ágeis?
- Q2. Uma abordagem integrando atividades e artefatos de DxD ao processo Scrum pode apoiar a equipe ágil na implementação dos aspectos de usabilidade durante as atividades de desenvolvimento de software?

### 1.2.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Revisão da literatura para identificar estratégias comuns de integração do DxD do usuário com metodologias ágeis;
- Definição da metodologia (SoftCoDeR) para conduzir pesquisa em cooperação com profissionais da indústria;
- Definição e elaboração dos artefatos para apoiar a integração do DxD no processo ágil de desenvolvimento de software;
- Proposta da abordagem para integrar o DxD do usuário ao processo Scrum;

- Planejamento do estudo experimental para validar a abordagem proposta em ambiente acadêmico a partir de um cenário real fornecido por profissionais da indústria de software;
- Definição das métricas para avaliar a usabilidade dos produtos de software desenvolvidos na etapa de validação da proposta;
- Desenvolvimento da aplicação web para acompanhar as atividades do DxI do usuário em paralelo ao desenvolvimento de software;
- Execução do estudo experimental, análise e interpretação dos resultados obtidos.

### 1.3 Metodologia e organização

A fim de atingir o objetivo geral e o específico este trabalho foi organizado a partir de técnicas de revisão bibliográfica, uma abordagem de pesquisa derivada da pesquisa-ação e um estudo experimental. A Figura 1.1 apresenta a visão geral da metodologia utilizada.

Na primeira etapa, um estudo bibliográfico foi conduzido para verificar as possíveis estratégias para a integração do DxI com o processo Scrum, incluindo práticas, papéis dos profissionais, processos e *frameworks* (a). Além de embasar a proposta da abordagem de integração ScrumUX, o estudo bibliográfico forneceu fundamentos para conduzir o trabalho de pesquisa-ação junto com os profissionais da indústria de desenvolvimento de software para definir e validar os artefatos que dariam suporte à abordagem proposta (b). Ao final da primeira etapa a abordagem ScrumUX foi proposta (c). Na etapa 2, um estudo experimental foi preparado (d) e executado em ambiente acadêmico (e), a fim de validar a abordagem de integração ScrumUX. O último passo consistiu na análise e interpretação dos resultados, discussão sobre as ameaças à validade do estudo e as conclusões (f).

### 1.4 Contribuições e resultados obtidos

A partir da execução dos passos descritos anteriormente e visando os objetivos propostos pode-se apontar as contribuições e resultados a seguir:

- Atualização do estado da arte sobre as estratégias comuns para integrar DxI às metodologias ágeis;
- Definição da abordagem SoftCoDeR utilizada na conduzir pesquisa em cooperação com profissionais da indústria e desenvolver os artefatos para apoiar a integração do DxI do usuário ao processo Scrum;
- Proposta da abordagem ScrumUX para integrar o DxI do usuário ao processo Scrum;
- Definição da métrica  $UI_{ERP}$  para avaliar a usabilidade de sistemas ERP;
- Validação da abordagem ScrumUX através de um estudo experimental;
- Aplicação UX Designer para acompanhamento de projetos de *design*;

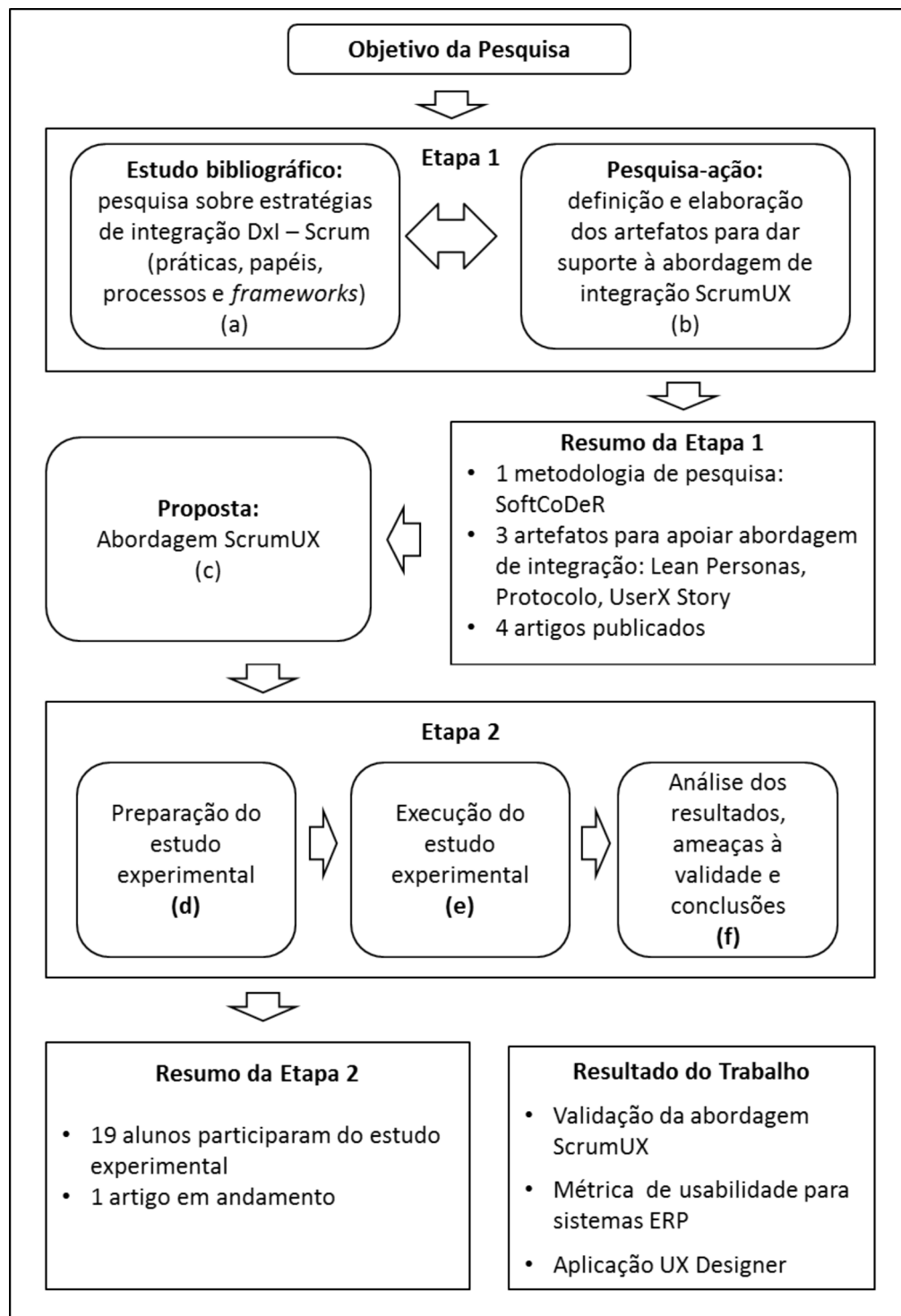


Figura 1.1: Visão geral da metodologia utilizada

## 1.5 Organização do trabalho

Esta dissertação está dividida em outros cinco capítulos, além deste capítulo introdutório. O segundo capítulo apresenta os fundamentos e técnicas que são utilizados ao longo deste trabalho.

---

O terceiro capítulo aborda trabalhos relacionados selecionados a partir de estudos sistemáticos de literatura. O capítulo quatro descreve o processo de pesquisa-ação executado para definir os artefatos para dar suporte à integração do Dxl ao processo Scrum, finalizando com a proposta da abordagem ScrumUX. O quinto capítulo refere-se a validação da abordagem proposta por meio de um estudo experimental, incluindo análise e interpretação dos resultados e validade do estudo. O último capítulo apresenta as conclusões, as contribuições e limitações bem como possibilidades de trabalhos futuros.

## Fundamentos e técnicas

### 2.1 Considerações iniciais

O foco deste capítulo se divide em duas vertentes: o DxI e as metodologias DAS. A Seção 2.2 apresenta os fundamentos e técnicas do DxI conceituando a usabilidade do ponto de vista da UX. A Seção 2.3 apresenta o conjunto de princípios que norteiam as metodologias ágeis de desenvolvimento de software, e destaca o processo Scrum descrevendo sua prática incluindo papéis, cerimônias e artefatos.

### 2.2 Design de Interação (DxI)

Design de Interação é abordagem de UX utilizada para explorar soluções que permitam criar produtos interativos para apoiar o modo como as pessoas se comunicam e interagem (ROGERS; SHARP; PREECE, 2013). Uma experiência adequada está relacionada com atender expectativas, qualidade da experiência e totalidade das percepções dos usuários.

A usabilidade apresenta um papel importante dentro do contexto da UX. Nielsen (2012) define usabilidade como um atributo de qualidade que avalia a facilidade de uso de uma determinada interface por análise de cinco fatores:

1. Capacidade de aprendizagem: a facilidade de utilizar o sistema pela primeira vez;
2. Eficiência: rapidez para executar as tarefas;
3. Memorização: o processo de lembrar como utilizar o sistema, após um tempo sem utilizar;
4. Erros: ausência de erros apresentados pelo sistema; e
5. Satisfação: design agradável.

O Design da Experiência do Usuário (UXD - User Experience Design) busca resolver questões relacionadas à usabilidade e qualidade da experiência de forma interdisciplinar, holística e direcionada a um profundo entendimento sobre comportamento, cognição, capacidades, desejos e contexto humano (NIELSEN J.; NORMAN, 2013). A Figura 2.1 mostra a abrangência do



UXD e as sobreposições do DxD com as demais disciplinas na visão de Saffer (2009), revelando que alcançar uma experiência do usuário de alta qualidade envolve contribuições de várias áreas.

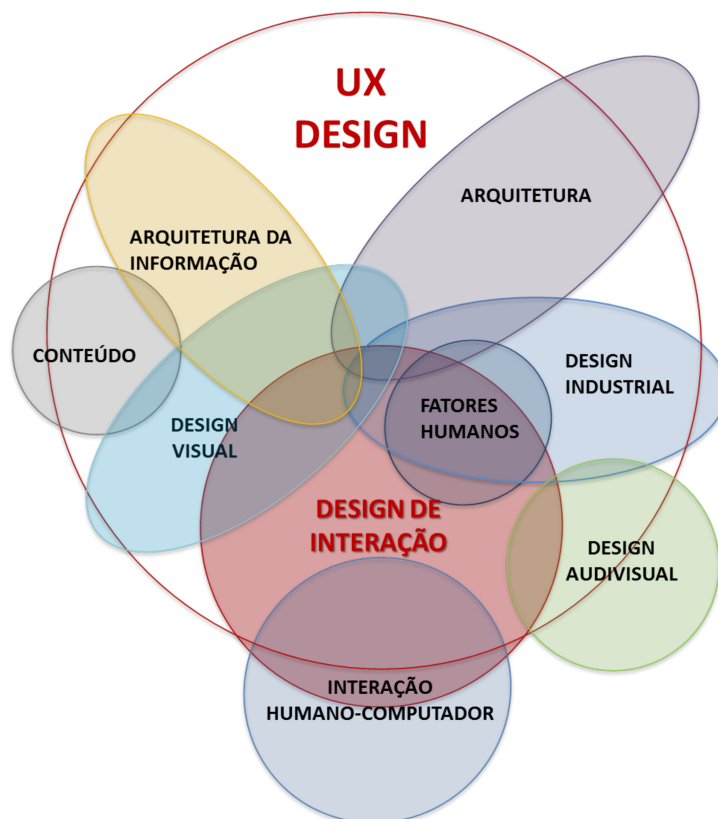


Figura 2.1: Abrangência de UX e as disciplinas em torno do DxD. Adaptado de Saffer (2009)

Dentre os diversos aspectos representados na Figura 2.1, a Arquitetura de Informação, por exemplo, fornece soluções para melhor organizar e rotular o conteúdo para que os usuários encontrem a informação que precisam. O *Design* Visual está relacionado com uma linguagem visual para comunicar conteúdo, tais como uso das fontes, cores e disposição dos elementos de interfaces de usuário. Os Fatores Humanos visam garantir o bem estar físico e psicológico do usuário. A área de Interação Humano-Computador (IHC) está intimamente relacionada com o DxD, pois fornecem técnicas e metodologias que permitem estudar e desenvolver a experiência do usuário para produção de soluções interativas que atendam as expectativas dos usuários (SAFFER, 2009).

Outra área relacionada com o DxD, embora não apareça na visão de Saffer (2009), é a área de Engenharia de Usabilidade (EU), responsável por fornecer métodos estruturados para avaliar e fazer recomendações para melhorar a usabilidade do *design* de interface do usuário durante o desenvolvimento do produto (MAYHEW, 1999). Embora cada disciplina tenha bem definido seus limites de aplicação, a associação entre elas é natural por conta do objetivo comum que é encontrar soluções que atendam necessidades humanas.

### 2.2.1 Ciclo de vida do DxI

A norma ISO 13407 estabelecida em 1999 é um padrão internacional que fornece orientação sobre atividades de DCU em todo o ciclo de vida de sistemas interativos baseados em computador que geram processos de *design* (ISO/IEC, 1999). As principais atividades de DCU são (1) compreender e especificar o contexto de uso; (2) especificar os requisitos de usuário; (3) produzir soluções de *design*; e (4) avaliar *design* considerando todos os requisitos de usuário (JOKELA et al., 2003). A ISO 13407 destaca que os principais benefícios do DCU são melhorias nos níveis de satisfação e produtividade dos usuários, bem como a redução dos custos de treinamento e suporte. A Figura 2.2 fornece uma representação simplificada do ciclo de vida de DxI baseado no modelo apresentado por Rogers, Sharp e Preece (2013) em consonância com as quatro atividades de DCU descritas na ISO 13407.

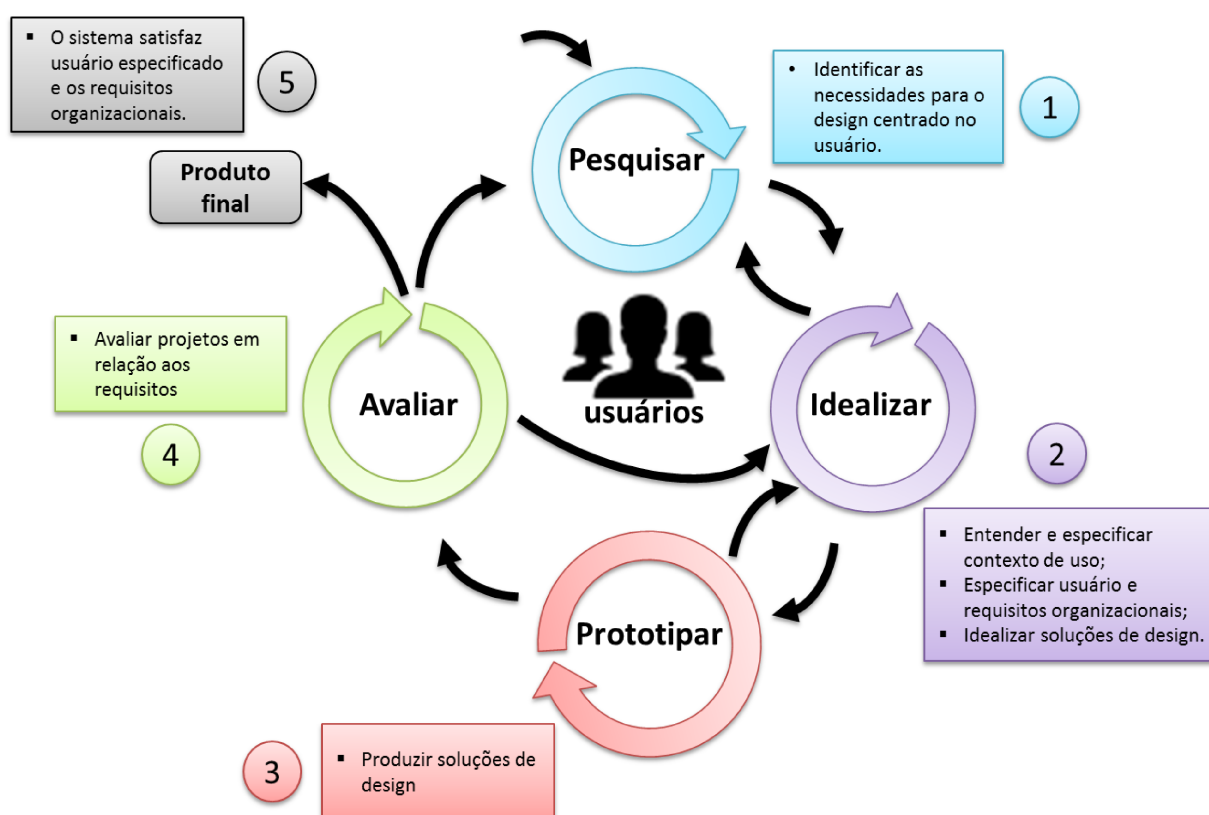


Figura 2.2: Ciclo de vida de DxI em conformidade com as atividades do DCU

O ciclo de DxI inclui quatro atividades básicas: (1) Pesquisa, (2) Ideação, (3) Prototipação e (4) Avaliação. Essas atividades estão interligadas, pois o resultado de uma atividade fornece subsídios para a atividade subsequente. O DxI requer um alto nível de envolvimento do usuário por todo o desenvolvimento com a finalidade de aumentar as chances do produto final ser bem aceito. Os ciclos DxI são iterativos que se repetem até que todos os requisitos do usuário sejam plenamente satisfeitos (ROGERS; SHARP; PREECE, 2013).

O DxI inicia-se com a atividade de *pesquisa*, cujo foco é identificar as necessidades para o *design* centrado no usuário (1). Analisando os dados coletados durante a fase de *pesquisa* é

possível especificar o usuário, o contexto de uso, e os requisitos organizacionais para *idealizar* soluções de *design* (2). Uma vez idealizadas, a próxima atividade será *prototipar* tais soluções e propor alternativas de *design* que atendam as reais necessidades dos usuários identificados como o público alvo da aplicação (3). A finalidade dos protótipos é viabilizar uma forma eficaz para que os usuários possam *avaliar* o *design* do produto em relação às tarefas e requisitos especificados (4). O principal objetivo da avaliação é identificar antecipadamente problemas de interação que poderão ser ajustados em tempo de *design* e de forma iterativa até que todos os requisitos de interação sejam satisfeitos (5) (JOKELA et al., 2003).

Dentro do escopo de UX as fases descritas são habitualmente conhecidas como *User Research* (Pesquisa do Usuário), *UX Ideation* (Ideação), *UX Desing* (Prototipação) e *UX Validation* (Avaliação).

### 2.2.2 Técnicas para o design centrado no usuário

Existe uma ampla variedade de técnicas que podem ser empregadas durante o DxI para conduzir o design centrado no usuário (COURAGE; BAXTER, 2005) (KUNIAVSKY, 2012). O conjunto de técnicas selecionadas para apoiar o processo de *design* dependerá, por exemplo, do tipo de sistema ou aplicativo que será desenvolvido, contexto, cronograma e recursos disponíveis (ROGERS; SHARP; PREECE, 2013). A Tabela 2.1 descreve de forma sucinta as técnicas mais comuns, indicando em quais as fases do DxI costumam ser empregadas, podendo aparecer em uma ou em várias fases do projeto. As atividades do DxI são indicadas por números como segue:

1. Pesquisa: coleta das necessidades dos usuários, tarefas e contexto de uso;
2. Ideação: análise e especificação dos requisitos do usuário e contexto de uso;
3. Prototipação: produção de soluções de *design*;
4. Avaliação: avaliar *design* em relação aos requisitos e coleta de opinião dos usuários a respeito do produto.

Tabela 2.1: Técnicas para o *design* centrado no usuário

Técnica	Descrição	Atividades DxI
Entrevista	Permitem obter informações detalhadas sobre as atitudes, desejos e experiências dos usuários. São adequadas para estudos exploratórios, onde não se sabe bem o que se está procurando, já que o entrevistador pode ajustar a entrevista conforme a situação.	1 e 4
<i>Survey</i>	Fornecem principalmente dados quantitativos sobre o público e o segmento de mercado, onde representantes selecionados aleatoriamente do público são convidados a preencher questionários.	1 e 4

<i>Brainstorming</i>	Busca levantar de forma bastante livre um conjunto grande e abrangente de opiniões dos participantes em torno de um tema, que resulta numa lista priorizada de necessidades e desejo dos usuários.	1
Grupo de Foco	Permite compreender as atitudes, ideias e desejos do usuário. Trata-se de uma discussão moderada com um grupo de usuários (de seis a nove usuários) que são levados a se reunirem para discutir novos conceitos e identificar problemas durante um período de cerca de duas horas. Cada grupo é gerido por um moderador que é responsável por manter o foco do grupo nas questões de interesse.	1
Etnografia	Técnica de observação direta que permite ao pesquisador observar e entender o comportamento natural de usuários enquanto desempenham uma atividade em seu próprio ambiente de atuação. Na etnografia pura não há interação do observador com os participantes, e na etnografia participativa o observador pode adotar diferentes níveis de participação de maneira formal ou informal.	1 e 4
<i>Think-aloud</i>	Técnica utilizada durante as atividades de observação direta em ambientes controlados para coletar o que os usuários pensam sobre <i>design</i> . A técnica requer que as pessoas digam em voz alta o que estão pensando e o que estão tentando fazer durante a execução de tarefas.	1 e 4
Diário	Técnica utilizada durante as atividades de observação indireta para acompanhar as atividades do usuário que são convidados a escrever um diário relatando o que e quando fizeram as atividades, o grau de dificuldades, os problemas encontrados e a maneira utilizada para transpô-los. São especialmente úteis quando os participantes estão dispersos e inacessíveis pessoalmente.	1 e 4
<i>Logs de interação</i>	Técnica de observação indireta que utiliza a instrumentação do software para gravar as ações dos usuários para posterior análise. Alguns exemplos de ações que podem ser gravadas são o movimento do mouse, as teclas pressionadas e o tempo gasto em cada parte do sistema. A forma de registro de interação mais popular é a web <i>analytics</i> .	1 e 4
Personas	Técnica para criar personagens fictícios, arquétipos hipotéticos de um grupo de usuários reais, definindo assim os usuários típicos dentro de um contexto de interação. As personas são utilizadas para elaborar cenários de interação, e ainda podem ser usadas para descrever as tarefas típicas em testes de usabilidade.	2 e 3

Prototipagem	Permite explorar ideias antes de serem implementadas por meio da criação de modelos preliminares ( <i>mocups</i> ). Os protótipos podem ser produzidos com diferentes níveis de fidelidade, desde protótipos de papel ( <i>sketches</i> ) com desenhos representando projeto de telas (baixa fidelidade), esboços de tela implementados com auxílio da tecnologia (média fidelidade), até completamente funcionais (alta fidelidade). <i>Storyboard</i> é o arranjo de protótipos para simular uma sequência de ações.	2, 3 e 4
Cenários de interação	É uma descrição narrativa sobre o contexto de uso, funções e tarefas do usuário. Costumam ser utilizados em conjunto com protótipos e <i>storyboards</i> para adicionar mais informações sobre as ações interativas para execução de tarefas.	2
<i>Card sorting</i>	Busca encontrar padrões na forma como os usuários organizam suas ideias no momento da interação e como classificam determinada informação.	1, 2 e 4
Casos de uso essencial	Este artefato, proposto por Constantine (2004), é utilizado para descrever como os usuários usam um determinado recurso, fornecendo uma visão detalhada de como os usuários interagem com o sistema, incluindo as etapas para realizar cada tarefa. O objetivo de um caso de uso essencial é transmitir os aspectos mais importantes da interação usuário-sistema, incidindo sobre a intenção do usuário (sem referências ao projeto de UI) e sobre o resultado observável do sistema (sem especificar os passos internos do sistema para alcançar o resultado).	2
Investigação Contextual	Estudo de campo com envolvimento intenso do investigador que adota um modelo de mestre-aprendiz exercendo o papel de aprendiz do trabalho do usuário e o usuário no papel de mestre ensinado seu trabalho para o investigador-aprendiz.	1 e 4
Análise de Tarefas	Envolve aprender sobre os objetivos dos usuários, incluindo o que os usuários querem alcançar, e ajuda a entender as tarefas que os usuários irão realizar em atingir seus objetivos.	2

Avaliação de usabilidade	Procedimento para a coleta de dados de uso relacionados à interação de usuário com o sistema, ou para verificar se as propriedades específicas do software alcançam certo grau de usabilidade. Os dois métodos de avaliação amplamente utilizados por praticantes na indústria de desenvolvimento software são os testes de usabilidade e as avaliações analíticas. Os testes de usabilidade são aplicados com o objetivo de medir a usabilidade do software enquanto os usuários executando tarefas no sistema são observados. As avaliações analíticas são inspeções baseadas na análise de peritos. Alguns exemplos são o percurso cognitivo, a inspeção semiótica, sendo que a mais utilizada é a avaliação heurística. A análise heurística consiste em submeter a interface de um determinado aplicativo à avaliação de alguns especialistas em usabilidade, conforme um conjunto previamente determinado de “bons princípios de usabilidade”.	1 e 4
--------------------------	--	-------

As técnicas mais populares para coletar requisitos de usuário são entrevistas, *survey* e etnografia. O *card sorting* e a análise de tarefas são algumas das técnicas utilizadas para auxiliar na análise dos requisitos do usuário, enquanto que personas e cenários de interação são as técnicas mais utilizadas para especificar usuários, tarefas e contexto de uso na fase de ideação (COURAGE; BAXTER, 2005).

A maioria das técnicas de coleta por observação pode ser usada desde a fase inicial do projeto, para compreensão do contexto dos usuários, tarefas e objetivos, até em fases adiantadas do desenvolvimento para investigar, por exemplo, o quão bem o protótipo desenvolvido apoia as tarefas e objetivos dos usuários (KUNIAVSKY, 2012).

A triangulação de técnicas de pesquisa é uma prática comum realizada para validar questões de investigação através de resultados semelhantes coletados por diferentes perspectivas. Contudo, os dados coletados em cada técnica utilizada na triangulação podem ser de tipos diferentes e os resultados analisados podem não ser compatíveis (ROGERS; SHARP; PREECE, 2013).

### 2.2.3 Usabilidade

As diretrizes de usabilidade, também conhecidas como heurísticas, apresentam bons princípios de usabilidade utilizados para nortear as fases de *design* e de avaliação de produtos de software.

Os dois conjuntos de princípios de usabilidade mais conhecidos são as heurísticas de Bastien e Scapin (1993) e as heurísticas de Nielsen (1995). As heurísticas de Bastien e Scapin (1993) são mais voltadas para a área da Ergonomia<sup>1</sup>.

<sup>1</sup>A ergonomia, como disciplina, se vale dos conhecimentos sobre o homem no trabalho, com o objetivo de conceber e avaliar produtos e ferramentas que possam ser utilizados com o máximo de conforto, segurança e eficiência (CYBIS et al., 1998).

As heurísticas de Nielsen (1995), por sua vez, são princípios mais gerais que cobrem todos os aspectos das boas práticas de usabilidade para o *design* de interação, sendo amplamente conhecidas pelos especialistas em UX e utilizadas na indústria de desenvolvimento de software (FERNANDEZ; INSFRAN; ABRAHÃO, 2011). As 10 heurísticas de Nielsen são:

H1. Visibilidade do estado do sistema

“O sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de uma realimentação apropriada dentro de um tempo razoável.”

H2. Concordância entre o sistema e o mundo real

“O sistema deve utilizar a linguagem do usuário, com palavras, frases e conceitos familiares ao usuário, mais do que termos orientados para o sistema. Seguir as convenções do mundo real, fazer a informação aparecer na ordem natural e lógica.”

H3. Controle e liberdade ao usuário

“O sistema deve dar apoio a funções como “fazer” e “desfazer” ou funções que permitam ao usuário utilizar “saídas de emergência” em caso de escolhas de funções erradas ou para sair de um estado não esperado.”

H4. Consistência e padrões

“Devem ser seguidas convenções da plataforma de desenvolvimento e padrões de interface normalmente aceitos. Usuários não devem ter que adivinhar se palavras, situações ou ações diferentes significa a mesma coisa.”

H5. Prevenção de Erros

“O sistema deve prevenir a ocorrência de erros na sua utilização. Melhor do que apresentar boas mensagens de erros, é ter um projeto cuidadoso que previne a ocorrência de um problema, em primeiro lugar.”

H6. Reconhecer ao invés de lembrar

“Tornar objetos, ações e opções visíveis, para que o usuário não tenha que lembrar informações de uma parte do diálogo para outra ação. Instruções para uso do sistema devem estar visíveis, ou facilmente recuperáveis, quando necessário.”

H7. Flexibilidade e eficiência de uso

“Aceleradores (abreviações, teclas de função, etc.) podem tornar mais rápida a interação com o usuário. Permitir aos usuários customizar ações frequentes.”

H8. Projeto minimalista e estético

“Diálogos não devem conter informação irrelevante ou raramente necessária. Todas as unidades extras de informações em um diálogo competem com aquelas que são realmente relevantes, e diminuem sua visibilidade relativa.”

## H9. Reconhecimento, diagnóstico e recuperação de erros

“Mensagens de erros devem ser expressas em linguagem simples (sem códigos), indicando precisamente o problema, e sugerindo construtivamente uma solução.”

## H10. Ajuda e Documentação

“As informações de ajuda e documentação devem ser fáceis de procurar, com foco na tarefa do usuário, listando passos concretos que devem ser seguidos e não serem grandes.”

As heurísticas podem ser utilizadas com guia para a elaboração das soluções e para avaliação do *design*. A avaliação heurística, segundo Nielsen (2012), não fornece uma maneira sistemática para gerar correções para os problemas de usabilidade ou uma forma de avaliar a provável qualidade de quaisquer reformulações. No entanto, a avaliação heurística visa explicar cada problema de usabilidade observado com referência aos princípios de usabilidade estabelecidos.

## 2.3 Desenvolvimento Ágil de Software

Na Engenharia de Software (ES) o processo de software refere-se ao roteiro de desenvolvimento seguido pela organização para produzir sistemas de softwares. Ou seja, é o conjunto de atividades realizadas para resolver e manter sistemas de software. Cada organização pode ter um processo de software diferente, mas algumas atividades são comuns a todos os processos de software (SOMMERVILLE et al., 2011).

Práticas tradicionais de desenvolvimento de software costumam dar maior ênfase ao processo de software e a sua melhoria contínua, enquanto que metodologias de desenvolvimento ágil de software (DAS), ao contrário, direcionam o foco para as pessoas, valorizando o trabalho em equipe e a colaboração entre o cliente e a equipe de desenvolvimento (FERRE; JURISTO; MORENO, 2005).

No desenvolvimento ágil as soluções para os diferentes problemas evoluem através do processo iterativo, onde as tarefas são divididas em pequenos incrementos e os incrementos são construídos em diferentes iterações, que tipicamente duram de uma a quatro semanas. As interações curtas objetivam rápida adaptação às mudanças para minimizar o risco e aumentar as chances de entregar um produto de valor às partes interessadas e dentro do prazo que foi estabelecido (PAELKE; NEBE, 2008).

Diferentes metodologias DAS surgiram a partir dos anos 90, tais como *Extreme Programming*<sup>2</sup> (XP), Scrum<sup>3</sup>, *Feature Driven Development*<sup>4</sup> (FDD), Kanban<sup>5</sup>, *Agile Modeling*<sup>6</sup>, *Lean*<sup>7</sup>, *Dynamic System Development Method*<sup>8</sup> (DSDM), entre outros.

---

<sup>2</sup><http://www.extremeprogramming.org/>

<sup>3</sup><http://scrummethodology.com/>

<sup>4</sup><http://www.featuredrivendevelopment.com/>

<sup>5</sup><http://www.infoq.com/resource/minibooks/priming-kanban-jesper-boeg/en/pdf/>

<sup>6</sup><http://www.agilemodeling.com/>

<sup>7</sup><http://www.leanessays.com/2002/11/principles-of-lean-thinking.html>

<sup>8</sup><http://www.dsdm.org/>



No início de 2001, um grupo de dezessete profissionais representando os diferentes métodos ágeis de desenvolvimento estabeleceram os valores da metodologia através do Manifesto Ágil (BECK et al., 2001), conforme apresentado na Figura 2.3.



Figura 2.3: Manifesto Ágil. Fonte: Beck et al. (2001)

Segundo Beck et al. (2001), os valores por trás do Manifesto Ágil são explicados através dos seguintes princípios:

- A maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
- Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
- Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
- Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.

- Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
- O método mais eficiente e eficaz de transmitir informações para, e dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
- Software funcional é a medida primária de progresso.
- Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
- Contínua atenção a excelência técnica e bom *design* aumenta a agilidade.
- Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
- As melhores arquiteturas, requisitos e *design* emergem de times auto organizáveis.
- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

Os elementos centrais que caracterizam os métodos ágeis são: iterações curtas e bem definidas; processos e equipe ajustados para o desenvolvimento de produtos que agregam valor real para o usuário com eficiência; e constante *feedback* do cliente validando o produto.

Culturalmente, os métodos ágeis cultivam a colaboração, a comunicação face a face, processos simplificados para evitar atividades desnecessárias; o mínimo necessário de documentação interna e o planejamento somente do que for essencial para dar início ao projeto de desenvolvimento (BEYER, 2010).

A empresa de consultoria independente Analysis.net Research realiza anualmente pesquisas para acompanhar o estado das metodologias ágeis no mercado de desenvolvimento de software (VERSIONONE, 2015). Segundo a consultoria, ano após ano, uma mudança significativa tem sido notada com relação à difusão do conhecimento de conceitos ágeis e a prática de desenvolvimento ágil. Em 2011, por exemplo, 81% dos entrevistados tinham conhecimento sobre ágil, e em 2013 essa porcentagem aumentou para 88%. Em 2014, 90% dos entrevistados disseram ter mais de um ano de experiência com técnicas de desenvolvimento ágil, trabalhando diretamente com desenvolvimento de software ou ligados a outros departamentos de TI, contudo a maioria ocupavam cargos de gerentes de projetos, Scrum Master e líderes de equipe de desenvolvimento. A pesquisa apontou ainda que, dentre os métodos ágeis, o Scrum continua sendo por anos consecutivos a metodologia ágil mais usada. O gráfico da Figura 2.4 mostra que 80% dos 3925 profissionais pesquisados têm usado na empresa onde atuam o Scrum puro (56%), o Scrum customizado (8%), ou ainda Scrum combinado com outros métodos, tais como o Scrum com Kanban (Scrumban – 6%) e o Scrum com XP (Scrum/XP Hybrid – 10%).

### 2.3.1 Scrum

Scrum foi desenvolvido por Ken Schwaber e Jeff Sutherland no início de 1990 (SCHWABER, 2004) e, conforme pesquisa apresentada anteriormente, tem sido amplamente utilizado por pro-

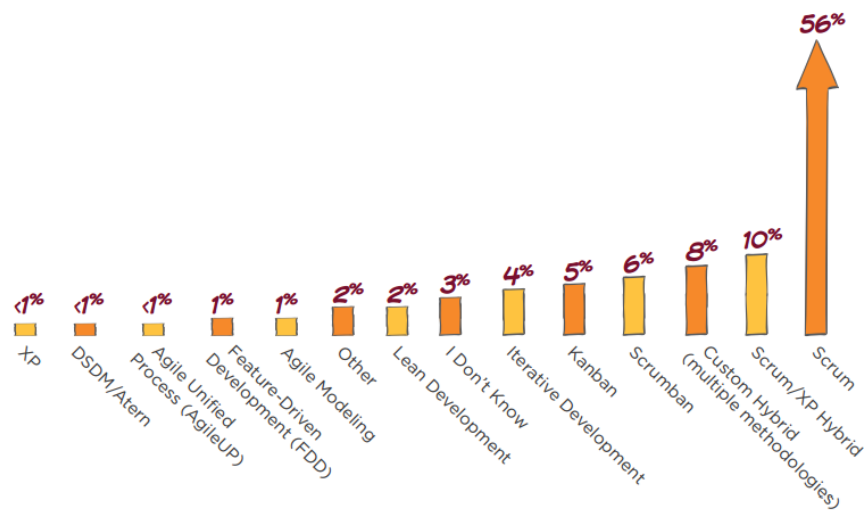


Figura 2.4: Metodologias ágeis utilizadas. Fonte: VersionOne (2015)

fissionais do mercado de software. O nome Scrum refere-se a uma jogada do *rugby*<sup>9</sup>, na qual os jogadores dos dois times desempenham papéis bem definidos e trabalham de forma integrada com foco num único objetivo na disputa de bola.

Schwaber e Sutherland (2011) apresentam o Scrum como um *framework* para suportar o desenvolvimento de software incrementalmente em ambientes complexos, onde os requisitos não são claros ou mudam com muita frequência (SCHWABER; SUTHERLAND, 2011). O Scrum fornece uma estrutura personalizável, para organizar e gerenciar o trabalho em projetos de desenvolvimento de software, baseado em um conjunto de valores, princípios e práticas, que permite adicionar práticas de engenharia e abordagens mais específicas para a realização do trabalho. Desta forma, o Scrum torna-se aderente à outras abordagens ágeis mais técnicas com o objetivo de inserir, por exemplo, diferentes práticas de programação e diferentes tipos de testes. Uma prática de programação amplamente adotada em projetos Scrum é a Programação Pareada (*Pair Programming*) que teve origem na metodologia XP. A Programação Pareada é uma técnica de desenvolvimento de software, onde dois desenvolvedores trabalham em conjunto para resolver um problema de desenvolvimento, evitar e diminuir a possibilidade de defeitos, melhorando assim a qualidade do código fonte gerado (PLONKA et al., 2015). Com relação aos diferentes tipos testes, pode-se citar os testes unitários (*unit testing*) e os testes de aceitação (*acceptance tests*), ambos também praticados no XP (BECK; FOWLER, 2001). Os testes unitários são realizados para determinar se um determinado recurso funciona como pretendido, e para isso cada pedaço de código que está escrito é testado antes de passar para a próxima característica; e os testes de aceitação são realizados para verificar se os requisitos conforme foi entendido pelos programadores satisfazem as necessidades reais do cliente.

Embora o Scrum tenha sido concebido com um *framework*, neste trabalho, este será tratado como processo de desenvolvimento ágil de software.

<sup>9</sup>Esporte inventado em 1823 no Colégio Rugby, na Inglaterra, praticado por duas equipes de 15 jogadores, com uma bola oval, que deve ser levada até o arco adversário (em forma de H), ou passada por cima da barra horizontal, com um chute (FERREIRA, 2004).

### 2.3.1.1 Papéis, cerimônias e artefatos do Scrum

Há três papéis principais no Scrum: o Dono do Produto (PO - Product Owner), a Equipe de Desenvolvimento (ED), e o Scrum Master (SM). Todos são parte do Time Scrum (*Scrum Team*). O PO está fortemente envolvido no processo, sendo responsável por representar os clientes (*stakeholders*) no projeto e por garantir que produto produzido será de grande valor para eles. O PO é responsável ainda pela gestão e priorização da lista de incrementos (*features*) de produto a ser desenvolvido (Backlog do Produto). A ED normalmente é pequena, e costuma ser formada por 5 até 10 desenvolvedores. A ED, responsável por desenvolver os incrementos, são auto-organizáveis e multifuncionais (*cross-functional*). Em uma equipe multifuncional as pessoas com diferentes expertises trabalham juntas para atingir um objetivo comum. No caso da ED Scrum as especialidades/habilidades individuais não recebem destaque, pois todos os membros da equipe respondem conjuntamente pelos resultados do trabalho e devem estar prontos para desempenhar diferentes atividades dentro da equipe quando necessário. O SM representa a gerência para o projeto e é responsável pela aplicação dos valores e práticas do Scrum; tem a função de remover obstáculos, proteger o time das interferências externas, garantir a plena funcionalidade e produtividade da equipe bem como a colaboração entre os diversos papéis e funções (SCHWABER; SUTHERLAND, 2011).

O Scrum prescreve um conjunto de artefatos e cerimônias para o desenvolvimento incremental e iterativo. O desenvolvimento incremental fornece informações importantes que permitem adaptar o esforço de desenvolvimento e ajustar procedimentos ao longo do projeto; enquanto que o desenvolvimento iterativo permite melhorar o produto a cada iteração, enquanto está sendo desenvolvido. No Scrum o trabalho é realizado em iterações ou ciclos, que costumam durar de duas a quatro semanas, chamados de Sprints. A Figura 2.5 apresenta uma visão global do Scrum ilustrando seus artefatos e cerimônias.

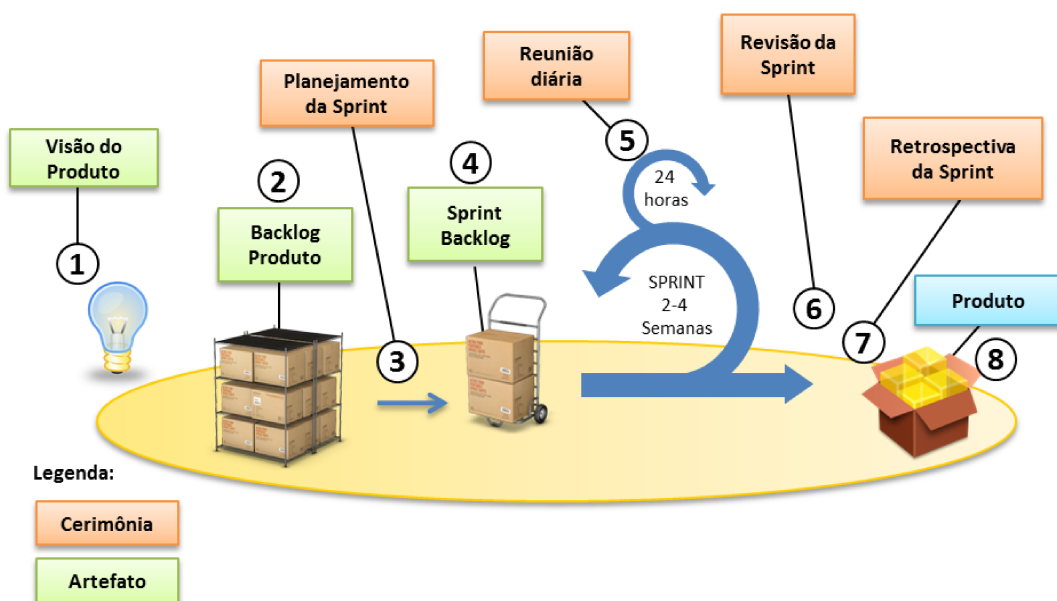


Figura 2.5: Artefatos e cerimônias do Scrum. Adaptado de Schwaber (2004)

Um projeto Scrum começa com uma Visão do Produto (1) que é estabelecida a partir do entendimento das necessidades e expectativas do cliente com relação ao produto final. Assim, a Visão do Produto é o artefato que deve assegurar o alinhamento e o compartilhamento do que deverá ser desenvolvido pela ED. Uma vez que as metodologias ágeis não permitem planejamentos extensos ou muito detalhados a Visão do Produto costuma ser definida simplesmente como o conjunto de características-chave (ou *features*) do produto final que poderão ser detalhadas à medida que o projeto avança (SCHWABER, 2004). Criar a Visão do Produto para pequenos projetos (de atualização ou de manutenção de sistema) pode levar alguns dias ou algumas horas. Entretanto, criar a Visão do Produto para grandes projetos envolvendo produtos complexos e inovadores, pode levar várias semanas ou até meses para que toda informação relevante seja compilada. Segundo Pichler (2009), nestes casos opta-se por executar uma ou mais Sprints para realizar o trabalho que for necessário para se definir a Visão do Produto (PICHLER, 2009).

A partir da Visão do Produto o PO define o Backlog do Produto (2), que é uma lista contendo todas as funcionalidades desejadas para um produto, priorizada conforme valor atribuído pelo cliente a cada funcionalidade. Uma vez definido o Backlog do Produto, a primeira reunião realizada no início de cada Sprint é a Reunião de Planejamento da Sprint (3). Na primeira parte da reunião, o PO apresenta o Backlog do Produto (2) para a ED. Na segunda parte da reunião, a ED decide como o trabalho será feito e define a Sprint Backlog (4) selecionando os itens do Backlog do Produto com os quais estarão comprometidos durante a Sprint. A Reunião diária (5) é uma breve reunião de no máximo 15 minutos realizada diariamente pela ED, em pé, sempre no mesmo horário e local. Durante a Reunião diária (5) cada membro da ED explica o que ele fez desde a última reunião, o que será feito antes da próxima, e ainda comunica se existe algum obstáculo impedindo seu trabalho. Na reunião de Revisão da Sprint (6), a ED demonstra o produto desenvolvido ao PO e aos outros *stakeholders*, avaliando ao final se o objetivo foi alcançado. Na sequência, realiza-se a cerimônia de Retrospectiva da Sprint (7) com o objetivo de inspecionar como foi a última Sprint e avaliar possíveis melhorias que poderão ocorrer nas próximas rodadas. Ao final de cada Sprint é obrigatório entregar um incremento do produto (*product increment*) codificado, testado e potencialmente utilizável (8). O Backlog do Produto é constantemente atualizado ao longo de todo o projeto Scrum e implementado com novos ciclos iterativos que repetem os procedimentos descritos acima (SCHWABER; SUTHERLAND, 2011).

### 2.3.2 User Story

User Story (US - Histórias de Usuário) é um artefato muito utilizado em metodologias ágeis para descrever de maneira simples as características do sistema contadas a partir da perspectiva do usuário, e especialmente úteis para capturar partes específicas de uma funcionalidade. Os itens do Backlog do Produto são frequentemente descritos no formato de US, que devem ser escritas sem muito detalhamento e na linguagem comum dos usuários, evitando-se jargão técnico ou linguagem de negócio. Os detalhes da US são explorados pela ED através de uma conversa com o PO ou diretamente com o cliente. O objetivo da conversa é eliminar qualquer mal-entendido a respeito do que deverá ser implementado. Cada história deve informar o tempo estimado para ser implementada, e essa estimativa é feita em colaboração com a ED (COHN, 2009).

Cohn (2009) recomenda um modelo de US que acabou se tornando um dos mais utilizados pela comunidade ágil, conforme apresentado na Figura 2.6.

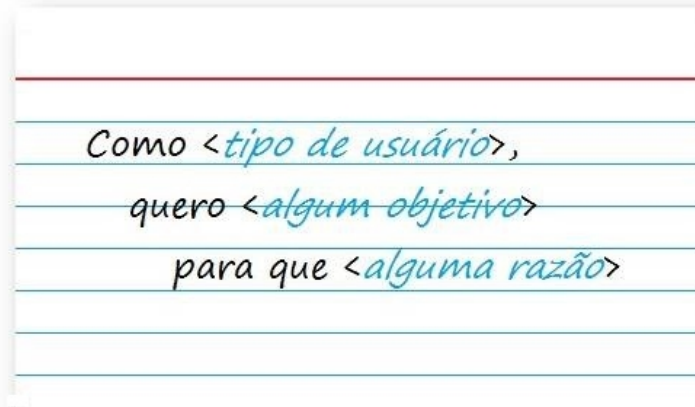


Figura 2.6: User Story (COHN, 2009)

O modelo de US recomendado possui três cláusulas para capturar o “tipo de usuário”, o “objetivo” e a “razão” de uma funcionalidade. Na maioria das vezes as US são escritas em papel de pequena dimensão, geralmente cartões de 3x5 polegadas, para evitar que sejam muito longas. Mas, as histórias podem ser armazenadas também em uma ferramenta de software como meio mais acessível principalmente para as que estão equipes distribuídas.

## 2.4 Considerações finais

Este capítulo apresentou os principais conceitos, técnicas e práticas relacionados com o Design de Interação e o processo Scrum de desenvolvimento ágil de software, que são a base da presente proposta e serão retomados posteriormente.

## Trabalhos Relacionados

### 3.1 Considerações iniciais

Este capítulo descreve os resultados de estudos sistemáticos de literatura realizados com o objetivo de identificar as estratégias mais comuns para integrar DxI em metodologias ágeis.

### 3.2 Estudos sistemáticos de literatura

Estudos sistemáticos de literatura, incluindo revisões, mapeamentos e a técnica *snowballing*, são abordagens de pesquisa utilizadas para identificar, analisar e interpretar todas as evidências disponíveis relacionadas a uma questão de pesquisa específica, permitindo aos pesquisadores chegar a um entendimento comum sobre o estado de uma área de pesquisa em ES (WOHLIN, 2014). A Revisão Sistemática de Literatura (RSL) é um tipo de estudo secundário, cujo processo sistemático de condução de pesquisa segue uma sequência de passos metodológicos explicitamente definidos em um protocolo para possibilitar que os mesmos resultados sejam reproduzidos por outros pesquisadores (KITCHENHAM; CHARTERS, 2007). O Mapeamento Sistemático de Literatura (MSL) destina-se a encontrar e classificar estudos primários em um tópico específico da área; usa o mesmo método para a busca e extração de dados que a RSL e tem sido apontado como um complemento para as RSL (PETERSEN et al., 2008). *Snowballing* (bola de neve) é um processo contínuo, recursivo para buscar trabalhos primários de forma sistemática com base na lista de referência ou nas citações de um artigo com a finalidade de identificar estudos adicionais. *Backward snowballing* (bola de neve para trás) é a pesquisa feita nas referências de determinado artigo, enquanto que *forward snowballing* (bola de neve para frente) é a pesquisa realizada a partir das citações do mesmo (WOHLIN, 2014).

### 3.3 Terminologias

Embora o foco do trabalho seja DxI e o processo Scrum, deve-se esclarecer que a revisão de literatura não está limitada a esses dois termos. Termos como DxI, UX, IHC, e Engenharia de Usabilidade têm sido empregados indistintamente para fazer referência à processos, métodos, comportamentos e práticas que de forma subjacente conduzem à usabilidade por meio do DCU.

No entanto, observou-se que a maioria dos trabalhos relacionados adotaram os termos DCU ou UX, muito provavelmente devido à suas abrangências. Da mesma forma, ao invés de considerar somente trabalhos relacionados com o processo Scrum, ampliou-se o foco da pesquisa para considerar qualquer metodologia ágil.

### 3.4 Revisão Sistemática de Literatura

A introdução de técnicas para o *design* centrado no usuário em metodologias ágeis vêm ganhando crescente interesse, principalmente por causa dos reconhecidos benefícios relacionados à usabilidade como forma de agregar valor ao software, incluindo a satisfação do usuário (LEE; MCCRICKARD, 2007). Silva et al. (2011) realizaram uma RSL com o objetivo de identificar evidências existentes sobre a integração de DCU e DAS abrangendo um período dez anos (de 2001 a 2010). A RSL produzida revelou que o interesse sobre o tema em questão tem aumentado nos últimos anos, tanto por parte da academia com por parte da indústria de desenvolvimento de *software*. As questões que nortearam o processo da revisão de Silva et al. (2011) estavam relacionadas com a identificação (1) dos aspectos de usabilidade abordados em processos ágeis, e (2) das práticas mais comuns utilizadas para resolver problemas em métodos ágeis.

Os autores seguiram as recomendações de Kitchenham e Charters (2007) e de Biolchini et al. (2005) para conduzir a referida RSL, e realizaram a busca automatizada nas seguintes bases eletrônicas de dados: IEEE Explorer Digital Library<sup>1</sup>, Scopus<sup>2</sup>, ACM<sup>3</sup>, Springer<sup>4</sup> e Google Scholar<sup>5</sup>. Utilizaram a seguinte sequência de palavras-chave (*search strings*) para buscar trabalhos nas bases eletrônicas:

scrum OR “extreme programming” OR “lean development” OR “” OR “feature driven usability” “human-computer interaction” OR “computer-human interaction” OR “human factor” OR “user experience” OR “user-centered design” OR “user interface”

AND

agile OR “agile method” OR “agile development” OR “agile practice” OR “agile project” OR “agile lifecycle” OR “development” OR “dynamic system development” OR “agile unified process”

Para determinar os critérios de inclusão e exclusão, e avaliar a qualidade dos artigos, Silva et al. (2011) aplicaram o conjunto de critérios para triagem proposto por Dybå e Dingsøyr (2008). Desta forma, um total de 58 artigos foram selecionados a partir de um conjunto inicial de 309 artigos. Os artigos selecionados foram analisados pelos pesquisadores, que os classificaram dentro dos seguintes tópicos: atividades de DxD, distribuição de papéis, tamanho do grupos de profissionais envolvidos, tipo de projeto, ponto de vista, tipo de abordagem, práticas e artefatos.

Em relação às atividades de DxD, os autores constataram que 78% eram estudos com foco nas fases de pesquisa do usuário e *design*, sendo que 53% do total tinham foco nas três fases do

---

<sup>1</sup><http://ieeexplore.ieee.org>

<sup>2</sup><http://scopus.com>

<sup>3</sup><http://dl.acm.org/>

<sup>4</sup><http://www.springer.com/br/>

<sup>5</sup><http://scholar.google.com.br>



DxI (pesquisa do usuário, *design* ou avaliação).

A maioria dos artigos mencionavam o trabalho de DCU realizados por profissionais especialistas, sendo que as equipes na maioria das vezes eram pequenas e os projetos eram novos. Enquanto 53,44% dos artigos foram escritos a partir do ponto de vista do pesquisador, 34,48% foram escritos na perspectiva dos profissionais de UX. Quanto ao tipo de abordagem, os artigos se concentraram em maior número na categoria de lições aprendidas (59%).

Os autores identificaram as seguintes práticas:

- prototipação;
- Testes de usuário;
- Métodos de inspeção;
- Equipes trabalhando em estreita colaboração.
- Sprint paralela: abordagem onde a equipe de UX executa seu trabalho em paralelo a equipe de desenvolvimento, na mesma iteração;
- Pequeno *design* antecipado (LDUF - *Little Design Up Front*): abordagem que permite a realização de uma parte do trabalho de *design* antes do início do desenvolvimento;
- Grande *design* antecipado (BDUF - *Big Design Up Front*): abordagem onde um bom tempo é dedicado para a investigação usuários antes do início do desenvolvimento;
- *Design* uma iteração à frente (*One Sprint Ahead*): abordagem na qual a equipe de UX trabalha pelo menos uma iteração à frente da equipe de desenvolvimento;
- *Big Picture*: recomendação para não se perder a visão holística do projeto;

As três práticas mais mencionadas (em mais de 30 artigos) foram os testes de usuário, o pequeno *design* antecipado (LDUF) e equipes trabalhando em estreita colaboração.

Dentre os artefatos identificados, os protótipos de baixa, mencionados em 24 artigos, aparecem em primeiro lugar; em segundo lugar as histórias de usuários, em terceiro os cenários e em quarto lugar aparecem as personas. Os demais artefatos identificados pelos autores, que apareceram com menor frequência, foram os modelos de interação, as diretrizes de usabilidade e os casos de uso essencial.

### 3.4.1 Atualização da revisão de literatura

Os resultados da RSL de Silva et al. (2011) foram a base para o processo de investigação deste trabalho com a finalidade de responder quais são estratégias mais comuns para integrar DxI em metodologias ágeis (Q1). Contudo a referida RSL precisava ser atualizada.

O primeiro passo nessa direção foi aplicar a técnica *snowballing* para coletar outras revisões de literatura relacionadas com a RSL de Silva et al. (2011). A Figura 3.1 mostra o esquema de pesquisa *snowballing* e os artigos que foram selecionados nessa fase.

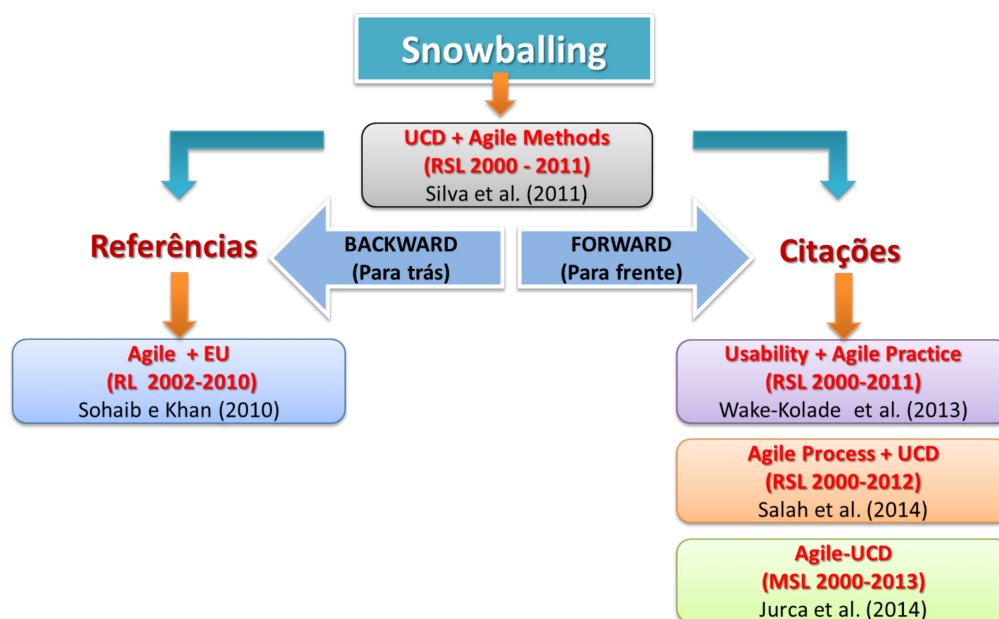


Figura 3.1: Esquema *snowballing*

Na pesquisa *backward snowballing*, o artigo de Sohaib e Khan (2010) foi selecionado dentre os artigos referenciados por Silva et al. (2011).

Sohaib e Khan (2010) realizaram uma revisão de literatura para investigar as tensões que tornam difícil a integração entre a Engenharia de Usabilidade em métodos ágeis e as abordagens que têm sido sugeridas para integrar usabilidade e métodos ágeis. As informações extraídas dos 22 artigos selecionados foram divididas em quatro temas: (1) foco no cliente vs foco no usuário final; (2) software funcionando vs software com usabilidade; (3) *design* necessário vs *design* antecipado; e (4) teste unitário vs teste de usabilidade. Analisando os resultados, os autores propuseram as seguintes recomendações:

- Adaptação da Engenharia de Usabilidade para o contexto ágil através do desenvolvimento iterativo e testes ao longo do ciclo de vida do projeto;
- Constituição de equipes multidisciplinares incluindo especialista em usabilidade;
- Colaboração entre clientes, usuários, gerentes de produto, analistas de negócios e desenvolvedores, maximizando eficiência na usabilidade do produto;
- Aplicação de testes unitário, testes de aceitação e testes de usabilidade ao longo projeto;
- Integração de histórias do usuário e *design* baseado em cenário.

A pesquisa *forward snowballing*, realizada na base do Google Scholar, retornou 53 artigos que haviam citado o artigo de Silva et al. (2011), dentre os quais, três foram selecionados por se tratar de revisões de literatura relacionadas ao tema de pesquisa, embora com interesses distintos de investigação.

Wale-Kolade, Nielsen e Päävärinta (2013) investigaram: (1) as recomendações mais comuns para realização do trabalho de usabilidade dentro do contexto ágil e (2) os fatores que influenciam tais recomendações, bem como a natureza dessas influências. Inicialmente, os autores selecionaram 38 artigos da RSL de Silva et al. (2011). Posteriormente, atualizaram a lista de artigos adicionando mais 11 artigos, que foram coletados em duas bases de pesquisa (Scopus<sup>6</sup> e Google Scholar<sup>7</sup>) utilizando os mesmos termos de pesquisa utilizados por Silva et al. (2011).

Analisando os 49 artigos selecionados, Wale-Kolade, Nielsen e Päävärinta (2013) identificaram as sete recomendações mais comuns sobre a integração do trabalho ágil e de usabilidade: (1) *design up front*, (2) protótipo de baixa fidelidade, (3) teste com usuário entre iterações, (4) *designers* e desenvolvedores trabalhando em paralelo, (5) presença de *designers* de usabilidade no projeto, (6) *designers* integrados na equipe de desenvolvimento e (7) envolvimento de usuários no projeto. Os autores analisaram a medida que tais recomendações foram validadas no contexto de desenvolvimento de *software*, considerando os argumentos favoráveis e desfavoráveis à prática extraídos da literatura revisada.

A RSL de Salah, Paige e Cairns (2014c) visava investigar: (1) os desafios que dificultam a integração DCU-Ágil e (2) as práticas propostas para lidar com esses desafios. Os autores selecionaram 71 artigos referente ao período de 2000 a 2012, sendo que o maior número de artigos concentrou-se no período de 2007 a 2008 (24 artigos). Dos 71 artigos selecionados, 32 constavam na lista da RSL de Silva et al. (2011). A partir das informações extraídas desse conjunto de artigos, os autores identificaram as práticas propostas na literatura para lidar com os desafios identificados, destacando os fatores de sucesso. Os sete desafios identificados foram:

1. Falta de tempo para antecipar atividades de *design* (*up-front design*);
2. Dificuldade em dividir o *design* em partes menores (*design chunking*), para que “pedaços do design” sejam adicionados ao projeto de desenvolvimento de forma incremental;
3. Dificuldade para priorizar as atividades de DCU;
4. Dificuldade em otimizar a dinâmica de trabalho entre desenvolvedores e os profissionais de DCU, que envolve: compartilhamento da visão de *design*, entendimento dos usuários e sincronização de esforços;
5. Realização de testes de usuários envolvendo: escolha do método mais apropriados, calendário de teste, acesso aos usuários, implementação das melhorias apontadas nos testes;
6. Carga de trabalho dos profissionais de DCU;
7. Falta de documentação.

Jurca, Hellmann e Maurer (2014), por sua vez, realizaram um mapeamento sistemático das publicações discutindo a integração de práticas de UX em metodologias ágeis. Aplicaram a

---

<sup>6</sup><http://scopus.com>

<sup>7</sup><http://scholar.google.com.br>

técnica *snowballing* a partir da RSL de Silva et al. (2011), e atualizaram a lista realizando buscas em quatro bibliotecas digitais (IEEE Explorer Digital Library<sup>8</sup>, Scopus<sup>9</sup>, ACM<sup>10</sup> e Springer<sup>11</sup>).

Jurca, Hellmann e Maurer (2014) incluíram um total de 76 artigos para o mapeamento sistemático, sendo que 29 constavam na lista de Silva et al. (2011), 23 foram coletados através da técnica *snowballing* e 24 coletados durante o processo de busca por trabalhos mais recentes. Com o objetivo de verificar tendências e identificar lacunas na literatura existente, os autores categorizaram, inicialmente, os 76 artigos selecionados através de um sistema de classificação estabelecido por Wieringa et al. (2006).

Por meio do sistema de classificação adotado, os estudos foram distribuídos conforme suas características em seis categorias: avaliação, validação, solução, experiência, filosófico e opinião. Dos 76 artigos 36,84% eram relatos de experiência, 22,36% propostas de soluções, 19,73% estudos de avaliação, 9,21% artigos filosóficos, 9,21% estudos de validação e 2,63% artigos de opinião. Aprofundando sua análise nos estudos de avaliação e validação, os autores apresentaram algumas conclusões relacionadas com a necessidade de apoio organizacional para que práticas de UX possam ser implementadas com sucesso em metodologias ágeis, bem como trazer a figura de *designer* de UX mais próxima das equipes ágeis.

Buscando-se aplicar a recursividade da técnica *snowballing*, realizou-se novamente a pesquisa *forward snowballing* nos artigos de Wale-Kolade, Nielsen e Päivärinta (2013), Salah, Paige e Cairns (2014c) e Jurca, Hellmann e Maurer (2014), para verificar revisões sistemáticas mais recentes relacionadas ao tema. Contudo, não foi possível acessar versão completa do único artigo encontrado de Zapata (2015), citando Jurca, Hellmann e Maurer (2014).

O segundo passo para inserir trabalhos mais recentes à lista de artigos foi conduzir uma nova RSL abrangendo o período remanescente (de 01/2014 a 06/2015), que não havia sido coberto pelas revisões anteriores. As fontes pesquisadas foram: ACM, IEEE Explorer Digital Library, Scopus e Google Scholar. E a sequência de palavras-chave foi a mesma utilizada por Silva et al. (2011).

No processo de seleção dos trabalhos considerou-se os seguintes critérios de inclusão: (1) apenas artigos completos, publicados em conferências ou *journal*, (2) escritos em inglês, e (3) com foco na integração de DCU (ou demais termos relacionados: DxI, UX, IHC, ou EU) em metodologias ágeis. E os critérios de exclusão foram: (1) não atender algum dos critérios de inclusão, (2) documentos de *workshop*, painel, tutorial ou seminário, (3) trabalhos sem contribuições relevantes, (4) revisões de literatura, (5) trabalhos que mencionam estratégias de integração via utilização de tecnologias (componentes ou aplicações), e (6) trabalhos relacionados com modelos de maturidade.

As consultas às bases foram feitas no período de 19 a 27 de julho de 2015. As buscas automatizadas retornaram 191 resultados após remoção dos trabalhos duplicados.

Na etapa seguinte, um total de 141 artigos foram excluídos com base na análise do título e resumo. Analisando-se o conteúdo das 50 publicações selecionadas na fase anterior, 23 artigos foram excluídos conforme motivos listados abaixo:

---

<sup>8</sup><http://ieeexplore.ieee.org>

<sup>9</sup><http://scopus.com>

<sup>10</sup><http://dl.acm.org/>

<sup>11</sup><http://www.springer.com/br/>

- 3 artigos por não evidenciar foco na integração de DCU e metodologias DAS;
- 8 artigos por impossibilidade de acesso ao documento completo;
- 4 artigos sem contribuições relevantes para a investigação;
- 3 artigos de revisão de literatura;
- 2 artigos que tratavam de integração via utilização de tecnologias;
- 3 artigos relacionados com modelos de maturidade.

Os 27 artigos incluídos na atualização foram analisados juntamente com 67 artigos selecionados das revisões anteriores. Contudo, foram selecionados somente os artigos publicados a partir de 2005, seguindo os mesmos critérios de inclusão e exclusão. A lista completa dos artigos selecionados foi anexada a esta dissertação no Apêndice A. O gráfico da Figura 3.2 mostra a distribuição dos artigos selecionados por ano de publicação. Observa-se que o número de publicações em 2014 aumentou significativamente em relação aos anos anteriores.

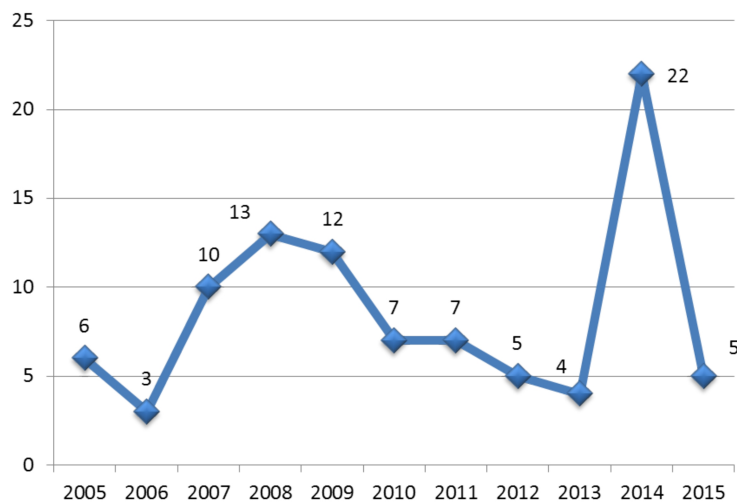


Figura 3.2: Número de artigos sobre integração do DCU em metodologias ágeis por ano.

### 3.4.2 Análise dos artigos selecionados

O sistema adotado para classificar os 94 artigos incluídos na revisão de literatura baseou-se no trabalho de Barksdale e McCrickard (2012), cujo principal objetivo foi investigar a interação social entre equipes de usabilidade e equipes DAS. Os autores consideraram cinco dimensões para classificar os trabalhos relacionados à integração do DCU em metodologias ágeis: práticas, pessoas, social, processos e tecnologia. Os autores classificaram dentro dessas cinco categorias, 65 artigos publicados em período anterior à publicação de Silva et al. (2011) (de 2000 a 2010).

Considerando que a classificação proposta por Barksdale e McCrickard (2012) produz uma visão mais abrangente para categorizar as estratégias de integração DCU-DAS, adotou-se duas

das categorias propostas pelos autores: práticas e processos. A categoria práticas refere-se a incorporação de práticas de DxDI em métodos ágeis, práticas ágeis incorporadas ao DxDI, ou ainda a partilha mútua de práticas DCU e DAS. A categoria processos engloba propostas que combinam métodos ágeis e métodos de *design* completamente distintos, fundidos em um único processo.

Decidiu-se por conveniência adicionar outras duas categorias: profissionais e *frameworks*. A categoria de profissionais tem foco no relacionamento entre os profissionais das duas áreas (DCU e Ágil), envolvendo questões relacionadas à organização do trabalho, comunicação e colaboração entre os profissionais. Considera-se que esta categoria engloba as dimensões “pessoas” e “social” propostas por Barksdale e McCrickard (2012). A categoria frameworks foi definida para destacar propostas de abordagens de integração que fornecem uma estrutura que serve para a agregação de processos, métodos e técnicas de *design* e engenharia de software. *Frameworks* fundamentalmente possui pressupostos, conceitos valores e práticas que orientam o fluxo das atividades para a integração de práticas de DCU em processo DAS.

A categoria tecnologia, como definida por Barksdale e McCrickard (2012), está fora do escopo do presente trabalho, assim sendo os trabalhos relacionados a este tema foram previamente excluídos na fase de seleção.

Uma vez definido o sistema de classificação, os trabalhos selecionados foram analisados e classificados dentro das quatro categorias. A Tabela 3.1 apresenta a distribuição dos artigos pelas categorias estabelecidas. Os trabalhos mais recentes aparecem em negrito.

Tabela 3.1: Artigos por tipo de estratégias

Categoria	Nº	Artigos
Práticas	50	[1] [3] [4] [5] [7] [9] [10] [11] [13] [18][19] [21] [22] [26] [27] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [43] [45] [46] [49] [50] [54] [55] [58] [62] [63] [65] [ <b>71</b> ] [ <b>74</b> ] [ <b>76</b> ] [ <b>77</b> ] [ <b>78</b> ] [ <b>79</b> ] [80] [81] [104] [85] [86] [90] [93]
Profissionais	32	[2] [6] [8] [14] [15] [16] [20] [28] [29][30] [42] [44] [47] [48] [51] [53] [57] [59] [60] [61] [64] [67] [ <b>68</b> ] [ <b>70</b> ] [ <b>73</b> ] [ <b>75</b> ] [82][83] [88] [89] [91] [92]
Processos	6	[12] [17] [23] [25] [66] [ <b>93</b> ]
Frameworks	6	[24][52] [56] [ <b>69</b> ] [ <b>72</b> ] [87]

Dos 94 artigos, 53,19% mencionam alguma prática para integrar o DCU em metodologias DAS; 34,04% mencionam o relacionamento entre profissionais de UX e desenvolvedores; 6,38% referem-se a propostas de processos com as metodologias integradas; e 6,38% referem-se a propostas de *frameworks*. Na sequência destacam-se os principais pontos relacionados com cada categoria.

### 3.4.2.1 Práticas

Dentro da categoria de integração por meio de práticas, incluindo métodos e técnicas, a maioria dos artigos costuma abordar etapas específicas ou apenas alguns aspectos relacionados com o processo de *design*, mas não cobrem o processo de desenvolvimento como um todo.

Os maiores desafios estão relacionados com encontrar um equilíbrio para sincronizar as atividades de DxD com o processo ágil. Quando observa-se o DAS, há evidências de pouco tempo dedicado para levantamento de requisitos; a codificação inicia-se com base nas suposições sobre as necessidades do usuário, assumindo que a informação faltante será preenchida em fases posteriores e de forma incremental através do *feedback* do usuário. Neste ponto as duas abordagens (DAS e DxD) são incompatíveis, pois o trabalho no DxD é mais gradual e requer um período maior antes do início do projeto para a pesquisa com o usuário, análise e documentação de dados coletados sobre necessidades, contexto de uso e tarefas do público-alvo (BLOMKVIST, 2005).

Miller (2005) e Sy (2007) descrevem uma das mais conhecidas práticas para integração de DxD em métodos ágeis, onde as atividades dos desenvolvedores e do *designer* acontecem em faixas paralelas (*Paralell Tracks*). Esta prática introduz o conceito de Sprint Zero, que é uma Sprint de curta duração estabelecida para definir uma visão mais ampla do produto antes de iniciar a implementação do código. A Sprint Zero permite aos *designers* de interação trabalharem um passo a frente da equipe de desenvolvimento para realizar pesquisas com usuários, conhecer os objetivos do projeto e planejar o trabalho de *design* durante as próximas Sprints. A partir do primeiro ciclo (Sprint 1), *designers* trabalham em paralelo à equipe de desenvolvimento projetando o que será implementado no próximo ciclo; e a partir do segundo ciclo (Sprint 2) avaliam o que foi codificado no ciclo anterior, projetam para o ciclo seguinte (Sprint 3) e se necessário realizam pesquisas dois ciclos à frente. Sy (2007) fala da necessidade de quebrar o projeto em pedaços menores (*design chunks*) para serem adicionados aos ciclos de forma incremental. Afirma ainda que, é possível selecionar os métodos mais adequados para pesquisa e avaliação (tais com entrevistas, investigação contextual, e testes de usabilidade) acertando o calendário e a granularidade das investigações, bem como alterando a maneira de apresentar os resultados.

Realizar algumas atividades de *design* antes do início do projeto é uma boa solução, principalmente quando existem restrições de tempo e orçamento. Executar atividades de *design* uma interação à frente da equipe de desenvolvimento (*One Sprint Ahead*), como solução para sincronizar as atividades das equipes, também é uma boa prática. Contudo, o trabalho em paralelo exige que *designers* tenham o controle sobre as suas atividades, sabendo que estas devem ser impulsionadas pelo trabalho dos desenvolvedores ágeis (atender prioridades), caso contrário fica impossível ajustar o fluxo de *design* no ciclo ágil de forma adequada. Existem evidências que apontam ser mais produtivo adaptar o DCU ao processo ágil do que forçar o alinhamento contrário adaptando o DAS ao DCU (ILLMENSEE; MUFF, 2009) (CHO, 2009).

A prototipação é uma prática muito comum, que tem o poder de conectar as áreas de *design* e desenvolvimento de uma forma muito natural. Protótipos têm sido utilizados para mediar a colaboração entre *designers* de interação e desenvolvedores (CHAMBERLAIN; SHARP; MAIDEN, 2006). Protótipos de papel são práticos e rápidos para apoiar o processo criativo, e são muito úteis, pois viabilizam as avaliações informais de usabilidade do *design* logo no início do projeto (LÁRUSDÓTTIR; CAJANDER; GULLIKSEN, 2014).

Os benefícios da avaliação de usabilidade a longo do processo de desenvolvimento são reconhecidos pela equipe de desenvolvimento (LÓPEZ-GIL et al., 2014), contudo não faz sentido

testes de usuários subsequentes a cada iteração, se não houver um conjunto de recursos implementados que justifique a prática (DÜCHTING; ZIMMERMANN; NEBE, 2007).

Silva, Silveira e Maurer (2015) acreditam que reunião de demonstração do produto é uma boa oportunidade para realizar sessões de teste de usuário. Sohaib e Khan (2011) sugerem a avaliação heurística durante os testes de aceitação e testes de usuário no lançamento do produto. Entretanto, não existe um consenso sobre qual a melhor forma de apresentar os resultados da avaliação de usabilidade para as equipes ágeis (BORNOE; STAGE, 2014).

Lárusdóttir, Cajander e Gulliksen (2014) relatam que, às vezes, os resultados da avaliação são utilizados para descrever novas histórias do usuário, e outras vezes os problemas de usabilidade são considerados como alterações de uma história em aberto. Moreno e Yagüe (2012) propõe a incorporação de aspectos de usabilidade em histórias de usuários ágeis, introduzindo o conceito de histórias de usabilidade para representar as histórias contendo os mecanismos necessários de usabilidade. Em contrapartida, histórias de usuários costumam ser utilizadas como base para planejar os testes de usuário. (GOSPER; BINNIE, 2011).

Cleland-Huang, Babar e Mirakhorli (2014) inserem histórias de usuário em personas que são utilizadas para priorizar requisitos arquitetonicamente significativos. Personas tem demonstrado o seu potencial como uma ferramenta eficiente para agrupar os usuários, com foco nas suas necessidades, objetivos e comportamento (CABALLERO; MORENO; SEFFAH, 2014). Além disso, as personas facilitam as sessões de trabalho colaborativo (CHO, 2009), e quando armazenadas em um local de fácil acesso permitem que os desenvolvedores foquem o desenvolvimento em características chave do usuário (BROSCHINSKY; BAKER, 2008) (HUSSAIN; SLANY; HOLZINGER, 2009).

De forma mais isolada algumas práticas são propostas para melhorar a interação entre as equipes de *design* e desenvolvimento. Barksdale, Ragan e McCrickard (2009), por exemplo, apresenta uma abordagem utilizando mapas conceituais para mapear cenários e requisitos de usuários, visando promover ações colaborativas entre engenheiros de usabilidade e a equipe ágil.

### 3.4.2.2 Profissionais

Os artigos classificados nessa categoria discutem sobre a interação e o comportamento dos profissionais de *design* e desenvolvedores trabalhando em projetos ágeis. Relatos de experiência apontam resultados positivos obtidos a partir do engajamento das equipes em um objetivo comum, especialmente quando se reconhece o valor que o DCU pode agregar ao produto (NAJAFI; TOYOSHIBA, 2008). Contudo, existe uma tendência natural para que os profissionais permaneçam concentrados mais nos interesses de sua área em oposição aos interesses gerais do projeto (KROPP; KOISCHWITZ, 2014). Além disso, observa-se que o contraste de estilos de pensamentos (*mindsets*) entre *designers* de UX e engenheiros de software dificulta a comunicação das metas de usabilidade associadas às recomendações de *design* (SALAH; PAIGE; CAIRNS, 2014b)(WALE-KOLADE; NIELSEN, 2015).

A discussão em torno dessas questões evidenciam forte necessidade de (1) acertar o foco das equipes ágeis para atender as necessidades dos usuários, (2) aumentar a cooperação entre *designers* e desenvolvedores através da articulação do trabalho, assim como (3) adaptar práticas para obter formas mais imediatas e eficientes de comunicação (FERREIRA; SHARP; ROBINSON,



2012)(PLONKA et al., 2014).

O desenvolvimento ágil se baseia fortemente em *feedback* do usuário. Algumas abordagens ágeis exigem que o cliente acompanhe o desenvolvimento para fornecer *feedbacks* regulares para a equipe de desenvolvimento. Contudo, na maioria das vezes os clientes não são capazes de identificar problemas de usabilidade que possivelmente seriam apontados por usuários reais do sistema (BLOMKVIST, 2005). O ideal é manter um programa de parceria com usuários para envolvê-los principalmente nas etapas de coleta dos requisitos e de avaliação do produto (WILLIAMS; FERGUSON, 2007). Mas, quando não for possível a participação de usuários reais, os profissionais de UX podem assumir esse papel para representá-los e defender seus interesses (SY, 2007)

Com relação à articulação do trabalho entre *designers* de UX e desenvolvedores, Lievesley e Yee (2006) discutem que é de fundamental importância a definição de papéis para *designers* atuarem mais efetivamente junto às equipes ágeis. Silva et al. (2013) identificaram três papéis essenciais que *designers* de UX podem desempenhar dentro de equipes ágeis, que são: *designer* de UX, desenvolvedor de UI e *designer* de interação.

Observando a prática, Fox, Sillito e Maurer (2008) identificaram três configurações possíveis na articulação do trabalho entre *designers* e equipe de desenvolvedores ágeis. A primeira configuração é uma abordagem especialista, na qual um especialista de UX faz parte da equipe e é responsável pelas atividades de *design* buscando o envolvimento de usuários e demais *stakeholders*. A segunda configuração é uma abordagem generalista onde todos os membros da equipe de desenvolvedores junto com clientes/usuários desempenham atividades de *design*. A terceira configuração é uma abordagem híbrida que permite a participação dos desenvolvedores em atividades de *design*.

As abordagens generalistas e híbridas podem exigir uma atenção maior com relação às atribuições e competências das equipes de desenvolvimento. No caso de equipes pequenas, ou de projetos sem recursos para contratar um especialista de UX, o ideal é investir na capacitação dos desenvolvedores para que eles possam desempenhar atividades de *design*, incluindo as atividades de pesquisas com usuário e de avaliação de usabilidade (LIKKANEN et al., 2014).

Profissionais de UX trabalhando em estreita colaboração com as equipes ágeis são responsáveis por levantar questões potenciais de usabilidade muito mais cedo no processo de *design*, e por manter o fluxo de informações entre todos os envolvidos no projeto (WALE-KOLADE, 2015). As práticas adotadas podem diferir de uma equipe para outra, ou de empresa para empresa, mas uma recomendação comum é para que *designers* de interação sejam continuamente envolvidos no projeto de software, ficando à disposição da equipe de desenvolvimento durante a implementação para esclarecer dúvidas sobre o projeto e advogar em defesa dos usuários (SCHWARTZ, 2014).

Quando se trata de equipes de UX e equipes de desenvolvedores trabalhando em paralelo, é importante organizar o trabalho de UX com base nas horas alocadas para o desenvolvimento, para que o trabalho de *design* não se estenda e atrase a data de lançamento (LESZEK; COURAGE, 2008).

A gerência de projetos deve estar ciente dos desafios que as abordagens ágeis podem impor na integração do DCU, principalmente quando as equipes não estão co-localizadas (DETWEI-

LER, 2007). Pode ser interessante incluir um proprietário de usabilidade de produto (Usability Product Owner) para trabalhar junto com o PO tradicional com o objetivo de facilitar o acordo sobre a visão do projeto em termos de experiência do usuário (SINGH, 2008) (WALE-KOLADE; NIELSEN, 2015).

Brown, Lindgaard e Biddle (2011) estudaram as formas de colaboração entre as equipes ágeis e UX na prática, identificando os artefatos que mediam a interação entre *designers* e desenvolvedores. Blomkvist, Persson e Åberg (2015) tratam os artefatos compartilhados<sup>12</sup> por *designers* de UX e times ágeis distribuídos como objetos de fronteira (Boundary Objects), observando que estes não funcionam de forma autônoma, mas sim como complementos de apoio à coordenação, colaboração e comunicação.

### 3.4.2.3 Processos

Nos últimos dez anos, um número reduzido de artigos apresentam propostas de processos integrando DCU e DAS. Memmel, Gundelsweiler e Reiterer (2007) propõe um ciclo de vida ágil interdisciplinar chamado CRUISER (Cross-Discipline User Interface Design and Software Engineering Lifecycle) que reúne um conjunto de práticas de *design* de interação na base do método ágil XP. O processo CRUISER começa com o levantamento de requisitos uma iteração à frente do desenvolvimento (*LDUF*) que não deve demorar mais do que o tempo de análise de requisitos permitido no XP. Os artefatos de *design* gerados na fase inicial são: casos de uso essenciais, cenários e protótipos de baixa fidelidade. A segunda fase do processo é a fase conceitual onde os protótipos da interface são refinados. A terceira fase é a fase de construção e teste semelhante ao processo incremental e iterativo do XP. Esta fase inicia com o planejamento da iteração e a criação dos testes de aceitação, que são posteriormente utilizados para avaliar as partes da arquitetura do sistema e a interface do usuário. A última etapa do ciclo de vida CRUISER refere-se à fase de implantação e produção. Enquanto os usuários estão trabalhando com o sistema, novas funcionalidades podem ser solicitadas, bem como problemas de usabilidade podem ser identificados. Assim, o ciclo de vida permite um retorno a fases anteriores para atender novas exigências.

Lee e McCrickard (2007) propuseram o processo de *design* baseado em cenário (XSBD - eXtreme Scenario-based Design) incorporando ao método ágil XP conceitos da Engenharia de Usabilidade, onde as etapas de *design* prosseguem em conjunto com as atividades de desenvolvimento de software. Uma fase curta de levantamento de requisitos é realizada no início do projeto para identificação das partes interessadas e análise de tarefas. A representação do *design* de interface é chamado de *Central Design Record* (CDR), que consiste em um conjunto de cenários que descrevem diferentes situações de uso, reivindicações relacionadas com funções específicas da interface e objetivos de *design*. O processo XSBD prescreve avaliações de usabilidade para verificar se o sistema é fácil e intuitivo para usuários finais. Os problemas de usabilidade identificados são rastreados com as histórias de usuários. Avaliações por especialistas (por exemplo, a avaliação heurística) são recomendadas para complementar o processo XP

---

<sup>12</sup>Os artefatos mencionados no estudo são: personas, cenários, mapa de efeitos, *sketches*, especificação de *design*, protótipos, resumos de avaliação, User Stories e páginas demo (BLOMKVIST; PERSSON; ÅBERG, 2015).

orientado a teste, permitindo avaliações em intervalos regulares. Essa abordagem assume que um engenheiro de usabilidade é um membro da equipe e está trabalhando com os desenvolvedores para implementar o sistema. Dessa maneira, engenheiros de usabilidade trabalham dentro dos ciclos incrementais mantendo uma visão de alto nível da interface, e executando avaliações que se encaixam dentro da processo XP.

Paelke e Nebe (2008) propuseram um processo ágil centrado no *design* de interfaces de realidade aumentada (RA) combinando DCU e Scrum. Os autores inserem no processo Scrum a fase preliminar de exploração, análise e *design* (alargada) para estabelecer as necessidades dos usuários, gerar modelos de RA e fornecer *designs* alternativos. Nessa fase de exploração, as equipes de desenvolvimento trabalham para avaliar viabilidades técnicas, enquanto *stakeholders* geram itens do Product Backlog. Especialistas de RA trabalham para identificar potenciais tecnologias e conceitos arquitetônicos adequados para o contexto do aplicativo, enquanto os especialistas de usabilidade analisam as necessidades dos usuários, tarefas, objetivos e do contexto através entrevistas e investigações contextuais. A partir da análise dos requisitos de usabilidade, protótipos são rapidamente elaborados. Os peritos de usabilidade irão acompanhar o processo de desenvolvimento para assegurar que as necessidades do usuário sejam atendidas. Durante fase de desenvolvimento, os autores sugerem o *design* participativo com usuários reais e especialistas de usabilidade, com a finalidade de verificar a facilidade de uso ao longo de todo o processo, permitindo que o projeto evolua de esboços rápidos, protótipos de papel, *mockups* até se tornarem protótipos funcionais. Testes envolvendo usuários finais são empregados para avaliar desde os primeiros esboços até sistema funcionando.

Obendorf e Finck (2008) propuseram um processo chamado XPnUE combinando XP com (1) o Rápido Design Contextual para coletar informações sobre o contexto de uso e responsabilidades e os relacionamentos de usuários finais (BEYER; HOLTZBLATT; BAKER, 2004), e a (2) Engenharia Baseada em Cenários (ROSSON; CARROLL, 2002) para direcionar o projeto de *design*. O Rápido Design Contextual propõe ser mais leve que a técnica de investigação contextual tradicional, pois dá mais ênfase para compreensão parcial do contexto, e menos ênfase para atividades de modelagem. Os cenários são misturados à *sketches* e as histórias de usuários incluem *storyboards*. O *feedback* dos usuários e as investigações contextuais adicionais conduzem a melhoria incremental dos cenários.

Abdallah, Hassan e Azim (2013) se basearam na proposta Lee e McCrickard (2007) e propuseram um processo ágil acoplado com um conjunto de métricas de usabilidade chamado o QXSBD (Quantified eXtreme Scenario Based Design). As métricas de usabilidade foram introduzidas para produzir entrega utilizável em curtos períodos de tempo e facilitar equipes com pouca ou nenhuma experiência de usabilidade. O processo inicia com um *workshop* de um dia para definir os objetivos do *design*, os diagramas de sistema de alto nível, os protótipos de baixa fidelidade e os cenários de usabilidade, a partir do ponto de vista das partes interessadas, por exemplo, engenheiros de usabilidade, desenvolvedores, usuários finais e clientes. Cenários de usabilidade são criados, priorizados e refinados durante o *workshop* e colocados na lista de requisitos. Os cenários de usabilidade são utilizados pelos engenheiros de usabilidade para analisar os requisitos de usabilidade do sistema e identificar preocupações, tais como capacidade de aprendizado, eficiência, etc. Cada cenário de usabilidade é associado com parâmetros crí-

ticos e avaliado de acordo com as medidas especificadas, fornecendo métricas de usabilidade quantificadas para orientar o projeto de desenvolvimento.

Butt e Butt (2015) propuseram um processo de desenvolvimento ágil centrado em corrigir problemas de usabilidade desde as fases iniciais. O processo proposto é composto de cinco etapas. A primeira etapa, chamada de FlexREQ é um processo automatizado de elicitación de requisitos que ajuda o líder da equipe de projeto gerenciar os requisitos do cliente considerando as necessidades dos usuários. Na segunda etapa os *mockups* são criados conforme prioridade dos requisitos. Na terceira etapa os *mockups* são testados um a um com usuários para coleta de *feedback*, detecção de defeitos e identificação de problemas de usabilidade. Na quarta etapa inicia-se a fase de codificação com testes unitários e testes de aceitação, na qual problemas de usabilidade e defeitos são acompanhados. Na última etapa, o software está pronto para implantação no cliente.

O ponto forte da proposta de Memmel, Gundelsweiler e Reiterer (2007) é que estes definem um conjunto de artefatos para cada fase do processo. As propostas de Lee e McCrickard (2007), Abdallah, Hassan e Azim (2013) e de Obendorf e Finck (2008) focam principalmente na fase de concepção e *design*, onde cenários são os principais artefatos de integração. Obendorf e Finck (2008) em um estudo de caso prático conseguiram mapear cenários com as histórias de usuário, onde vários cenários poderiam estar relacionados a uma única história, assim como várias histórias à um único cenário. Lee e McCrickard (2007) são os únicos que indicam a avaliação heurística em intervalos regulares para complementar o processo de testes. O diferencial da proposta de Butt e Butt (2015) é o processo automatizado de elicitación de requisitos que permite gerenciar requisitos incluindo as necessidades de usuários. Contudo, os autores não aplicam muitas práticas de *design*, ficando limitados à prototipação de baixa fidelidade para coleta de *feedback* com usuários.

O processo de Paelke e Nebe (2008) é o único processo integrado ao Scrum. Os autores descrevem detalhadamente as atividades em cada fase, definindo os papéis dos profissionais envolvidos. O ponto forte da abordagem é a fase preliminar ampliada para a exploração, análise e *design* de interfaces de realidade aumentada. Contudo, a fase de exploração no Scrum costuma ser bastante curta. Sendo assim, o ideal seria que a fase de exploração não excedesse o período de uma Sprint (de 2 a 4 semanas). Mas, como a proposta de Paelke e Nebe (2008) é específica para a área de realidade aumentada, os autores incluem atividades de *design* que normalmente o Scrum não inclui, necessitando de mais tempo para concluir todas as atividades de *design* e pesquisa sobre arquiteturas e novas tecnologias específicas para o domínio.

#### 3.4.2.4 Frameworks

O número de trabalhos que apresentam propostas de *frameworks* também é limitado. No geral, os *frameworks*, se compararmos com os processos identificados anteriormente, recomendam um número maior de práticas de *design*.

Wolkerstorfer et al. (2008) apresenta uma abordagem que incorpora ao processo XP técnicas de DxDI: estudos de usuários (grupos de foco, diários e entrevistas); personas; testes unitários estendidos para avaliação automática de usabilidade; testes com usuários e inspeções de usabilidade. Os estudos de usuários fornecem informações que serão utilizadas para a criação das

personas e criação de *mockups*. As personas são fixadas junto com as histórias de usuário. Especialistas de usabilidade fornecem *feedback* quando a implementação de um conjunto de recursos for concluído. O *feedback* dos especialistas serve para refinar o *design* da interface e para orientar os testes de usuário sobre a aplicação. Tanto o *feedback* dos especialistas de usabilidade quanto os resultados dos testes de usuários são adicionados aos testes automatizados como especificações executáveis para a aplicação real. Iterações curtas permitem que o sistema evolua de acordo com as necessidades dos usuários finais e as especificações derivadas de uso real.

O *framework* proposto por Silva et al. (2011) deriva de uma combinação de práticas mais comuns identificadas em sua RSL. As práticas indicadas na atividade de investigação são: entrevistas, investigação contextual e análise de tarefas. Os principais artefatos de apoio são protótipos de papel, cartões de *design*<sup>13</sup> (*design cards*), cartões de funcionalidades<sup>14</sup> (*feature cards*) e as histórias de usuários. As equipes de *design* e de desenvolvimento realizam suas atividades em paralelo conforme a abordagem descrita por Miller (2005) e Sy (2007). A partir da iteração 1, os *designers* projetam para a próxima iteração, avaliam o que foi codificado na iteração atual e fornecem *feedback* aos desenvolvedores para que estes realizem as correções necessárias. Caso não tenham tempo hábil para executar as correções de imediato, as mesmas serão incorporadas na próxima iteração.

Humayoun, Dubinsky e Catarci (2011) propuseram um *framework* onde sugerem um ciclo de vida envolvendo atividades de DCU em conjunto com o desenvolvimento ágil. As atividades de DCU aparecem em sobreposição com as atividades ágeis (*design*, codificação e testes). O ciclo de vida é composto por quatro fases. Na fase 1, usuários e especialistas de DCU (analistas de sistemas, especialistas de usabilidade, *designers* de UI) são envolvidos na atividade de elicitação de requisitos, *design* e prototipação. Na fase 2, artefatos de *design* e protótipos são avaliados por especialistas de DCU, e por vezes por um pequeno grupo de usuários finais através de métodos rápidos de avaliação. Na fase 3, a equipe de software corrige e melhora o *design* de acordo com os resultados da avaliação dos protótipos. Na fase 4, uma avaliação detalhada dos módulos desenvolvidos é realizada por usuários finais e/ou por especialistas de DCU, através de métodos de avaliação rígidos e com apoio de ferramentas automatizadas. Os resultados, comentários e sugestões serão incorporados no planejamento das próximas iterações.

Abdelouhab et al. (2014) propõem um *framework* que combina desenvolvimento ágil, DCU e SOA (Serviço Orientado à Arquitetura) aplicado para o domínio de gestão de desastres. Na fase inicial do projeto o modelo conceitual é transformado em *design* visual e em estruturas de *design*. As proposições de *design* são iterativamente refinadas evoluindo de *mockups* a protótipos interativos. As necessidades dos usuários são mapeadas para funcionalidades do sistema. Problemas de usabilidade são abordados antes da codificação, e protótipos podem ser apresentados para diferentes categorias dos usuários. O *framework* proposto tem 4 fases: (1) fase de estudo da organização e do domínio (análise de negócios, usuários, tarefas e sistemas legados); (2) fase de análise e elicitação dos requisitos (identificação e criação das histórias de usuários, descrição dos cenários de negócio); (3) fase de planejamento e priorização de iterações (prioriza os serviços de negócios); e (4) fase de iteração para versionamento (projeto dos componentes

<sup>13</sup>Cartões com informações a respeito do *design* de interface do usuário (SY, 2007)

<sup>14</sup>Cartões com informações a respeito dos requisitos funcionais do software (SILVA et al., 2011)

SOA, codificação, testes unitário e avaliação com o usuário usando *think-aloud*.

Iqbal e Khan (2014) propuseram um *framework* que incorpora conceitos e práticas da Engenharia de Usabilidade e do DCU no processo Scrum. A Engenharia de Usabilidade é aplicada na fase inicial do processo Scrum para identificar os problemas de usabilidade nos primeiros estágios, fazendo com que o desenvolvedor entenda esses problemas antecipadamente e tente evitá-los. As atividades incorporadas nessa fase são: (1) explorar e analisar o contexto (para determinar a necessidade real do usuário e quem são os potenciais usuários do sistema) (2) análise do usuário, determinar tarefas do usuário (analisar e definir os potenciais usuários e suas tarefas), (3) definir objetivos de usabilidade (determinar as metas de usabilidade), e (4) análise competitiva (pesquisa de mercado). As tarefas definidas são adicionadas ao Product Backlog seguindo 3 passos: descoberta (redefinidas no formato de user stories), experimento (as user stories são experimentadas) e validação (validando as tarefas por meio das user stories). O DCU é aplicado na concepção do produto (antes da codificação) seguindo as etapas de (1) especificar contexto de uso, (2) especificar requisitos, (3) produzir a solução de *design*, (4) avaliar *design*. O *design* validado segue para a implementação na próxima Sprint.

Anwar et al. (2014) propuseram um *framework* integrando Scrum e práticas de DCU. Os autores consideram que em projetos pequenos não há nenhuma necessidade de desenvolver protótipos durante as Sprints; a iteração zero (Sprint 0) é suficiente para que a equipe de *design* desenvolva os protótipos e demais recomendações de *design*. Na fase de planejamento, o PO determina os requisitos do software que será desenvolvido, e em seguida a equipe de *design* realiza a pesquisa com os usuários. As práticas recomendadas para esta fase são: investigação contextual, entrevistas, análise de tarefas, cenários, pesquisa *online*, personas, grupos de foco, questionário SUS<sup>15</sup> (System Usability Scale). Na Sprint 0, a equipe de *design* tem sete semanas para desenvolver protótipos, avaliar o *design* junto com os usuários e recolher *feedback*. A partir da próxima Sprint os *designers* podem refinar o *design* se for necessário. Com apoio de histórias de usuário e protótipos de baixa fidelidade a equipe de desenvolvimento inicia codificação e realiza correções de problemas identificados nas avaliações de usabilidade. Os autores recomendam as seguintes técnicas de avaliação: avaliação heurística, entrevistas, grupos de foco, testes remotos, teste de dispositivos móveis e cenários. Quando o software ficar pronto, executa-se um teste completo com casos de teste.

Dentre os *frameworks* propostos, Wolkerstorfer et al. (2008) e Silva et al. (2011) são os que mencionam uma maior quantidade de práticas e artefatos. Wolkerstorfer et al. (2008) sugerem a técnica de “personas extremas” na qual as personas definidas inicialmente sem muitos detalhes vão sendo refinadas de forma iterativa com os estudos de usuários. Além disso, estas são fixadas ao lado das histórias de usuário para representar o usuário no ambiente de desenvolvimento. Os artefatos definidos por Silva et al. (2011) refletem bem o princípio ágil de simplicidade e agilidade. Os quatro artefatos (protótipos de papel, cartões de *design*, cartões de funcionalidades e as histórias de usuários), juntos apoiam as atividades do desenvolvedor descrevendo tanto requisitos funcionais como as metas de *design*. As propostas de Humayoun, Dubinsky e Catarci (2011) e de Abdelouhab et al. (2014) quase não mencionam artefatos de *design*. Humayoun,

---

<sup>15</sup>O SUS é uma escala de usabilidade utilizada avaliar a satisfação dos usuários com relação à usabilidade do software, desenvolvida por John Brooke em 1996 (BROOKE, 1996).

Dubinsky e Catarci (2011) indicam avaliações rápidas por especialistas e por vezes testes com pequenos grupos de usuário; e Abdelouhab et al. (2014) indicam a técnica *think-aloud*, que é considerada uma técnica de avaliação de baixo custo. Os *frameworks* propostos por Anwar et al. (2014) e Anwar et al. (2014) são os únicos com integração no processo Scrum. Anwar et al. (2014) focam mais nas atividades da Engenharia de Usabilidade e do DCU, sem mencionar artefatos de apoio. Os autores não definem papéis dentro do projeto, trabalhando somente o contexto das disciplinas. A proposta de Anwar et al. (2014) é mais completa, pois abrange todas as fases do projeto, define os papéis de PO, do time de *design* e do time de desenvolvedores. Além disso, recomendam um menu práticas e artefatos de apoio incluindo personas e a avaliação heurística. Os desenvolvedores são apoiados por histórias de usuários e protótipos de baixa, no entanto os autores não mencionam se equipe de *design* acompanha as atividades dos desenvolvedores, e nem como são relatados os problemas de usabilidade.

### 3.5 Considerações Finais

Este capítulo apresentou a revisão de literatura realizada a partir da atualização de estudos sistemáticos sobre a integração de metodologias ágeis com o DCU que é a base do DxI. O conjunto de artigos selecionados para a revisão foram classificados por estratégias de integração. Mais da metade dos estudos mencionam a incorporação de práticas isoladas sem abranger todas as etapas de *design* e desenvolvimento de software. Outra estratégia muito discutida está relacionada com a interação entre os profissionais das duas áreas. Porém, existe uma quantidade limitada de trabalhos que apresentam soluções mais completas, tais como processos e *frameworks*. Processos e *frameworks* são estratégias importantes, pois conseguem tanto reunir práticas, como fornecer orientações para a organização do trabalho dentro de um roteiro de desenvolvimento.

# ScrumUX: da pesquisa-ação para a abordagem proposta

## 4.1 Considerações iniciais

Este capítulo apresenta uma discussão sobre as metodologias utilizadas para definir e validar os artefatos desenvolvidos para apoiar a abordagem para integrar DxD ao processo Scrum denominada ScrumUX.

Como o tema do trabalho tem forte aderência à prática na indústria optou-se por utilizar a metodologia de pesquisa-ação em conjunto com o método DSR (*Design Science Research*) para definição de artefatos que pudessem dar suporte à integração ScrumUX. Acredita-se que os artefatos possuem um significado relevante para os profissionais (times ágeis), pois torna concreto os conceitos aplicados e o momento de sua aplicação (BLOMKVIST; PERSSON; ÅBERG, 2015).

Artefatos são essenciais para o processo de software, pois podem relatar todas as informações criadas durante o período de desenvolvimento. No contexto da ES, artefatos incluem conceitos, ferramentas, técnicas, métodos e metodologias. As equipes de UX e Scrum costumam fazer uso de diferentes tipos de artefatos para apoiar suas atividades e para comunicar informações que são produzidas ao longo do processo de desenvolvimento. Artefatos definidos no Scrum se destinam a maximizar a transparência das informações e apoiar as decisões durante o projeto de desenvolvimento de software (SCHWABER, 2004), enquanto que os artefatos de UX se destinam a fornecer informações a respeito dos aspectos de interação do usuário e das soluções de *design* (BROWN; LINDGAARD; BIDDLE, 2011).

## 4.2 Metodologias de Pesquisa

Esta Seção apresenta as metodologias de pesquisa que deram origem à abordagem de pesquisa adotada para conduzir um projeto de pesquisa e desenvolvimento (P&D) junto a indústria de software.



### 4.2.1 Pesquisa-Ação e o Método Cooperativo de Desenvolvimento

A Pesquisa-Ação (AR - *Action Research*) é uma metodologia de pesquisa que se concentra particularmente na combinação de teoria e prática. Trata-se de um processo iterativo de investigação centrado na descoberta através da ação, que permite pesquisadores e profissionais atuarem juntos para a solução de problemas específicos em determinado contexto (AVISON et al., 1999). As principais atividades do AR estão relacionadas com diagnóstico do problema, intervenção prática e aprendizagem reflexiva (SJOBERG; DYBA; JORGENSEN, 2007).

A AR tem raízes históricas no movimento sociotécnico, cuja origem está associada à aplicação prática da ciência social e da psicologia para investigação de problemas sociais, por Kurt Lewin na década de 1940. Desde então, a AR tem evoluído e tem sido aplicada em diversas áreas científicas, tais como Enfermagem, Educação, Gestão, Sistemas de Informação (BASKERVILLE, 2008).

Santos, Travassos e Zelkowitz (2011) afirmam que a AR vem sendo empregada, de modo crescente, como uma metodologia alternativa para conduzir pesquisas na área de ES. Segundo autores, a AR estimula a investigação aprofundada das práticas de ES em contextos industriais, intensificando a execução de estudos relevantes que agregam valor à pesquisa. No entanto, os autores identificaram que o rigor e controle dos estudos de AR na área de ES ainda precisam ser melhorados.

O Método Cooperativo de Desenvolvimento (CMD - *Cooperative Method Development*) proposto por Dittrich (2002), é uma adaptação de AR voltado para práticas de desenvolvimento de software do chão de fábrica. O CMD combina a pesquisa de campo qualitativa em ciências sociais com métodos orientados à problemas e melhoria processos. O processo de pesquisa do CMD é modelado em ciclos evolutivos constituídos de três fases: (1) Entender a Prática, (2) Deliberar Melhorias, e (3) Implementar e Observar Melhorias.

No CMD a pesquisa começa na fase para *Entender a Prática* através de investigações empíricas qualitativas com o objetivo de compreender e explicar as práticas do ponto de vista dos profissionais, identificando aspectos que são problemáticos para os envolvidos no projeto. A segunda fase para *Deliberar Melhorias* resulta da discussão entre pesquisadores e profissionais na busca por soluções para melhorar a situação que foi identificada na primeira fase. Na terceira fase para *Implementar e Observar Melhorias*, em primeiro lugar, as melhorias são implementadas. Durante a implementação, os pesquisadores acompanham como observadores participantes, e posteriormente, avaliam os resultados em conjunto com os profissionais envolvidos. As três fases do CMD podem ser aplicadas repetidamente para melhorar o processo ou para investigar outras questões de pesquisa dentro do mesmo contexto.

### 4.2.2 Design Science Research

O *Design Science* (DS) surgiu originalmente do trabalho de Herbert Simon intitulado “*The Sciences of the Artificial*” (As Ciências do Artificial), publicado em 1969. O conceito do DS está fundamentado em várias disciplinas, tais como Arquitetura, Economia e Engenharias (BASKERVILLE, 2008). O DS aborda a pesquisa através da construção e avaliação de artefatos projetados para atender necessidades de negócio identificados em determinados contextos. Os

artefatos comumente produzidos pela DS Research (DSR) estão relacionados com conceitos, modelos, métodos e instâncias (VENABLE, 2006) (MARCH; SMITH, 1995).

O DSR tornou-se uma abordagem de pesquisa adotada com mais frequência na área de Sistemas de Informação (SI). Contudo, Iivari e Venable (2009) afirmam que o DSR tem sido implicitamente praticada na área de ES por décadas (IIVARI; VENABLE, 2009).

Hevner et al. (2004) propuseram um *framework* conceitual combinando o DS e a Ciência Comportamental para conduzir pesquisas na área de SI. Em outro trabalho, Hevner (2007) definiu três ciclos para melhorar a compreensão do DSR: Relevância, Rigor e *Design*. A Figura 4.1 apresenta o *framework* proposto por Hevner et al. (2004) ampliando o foco para os três ciclos definidos no processo de pesquisa.

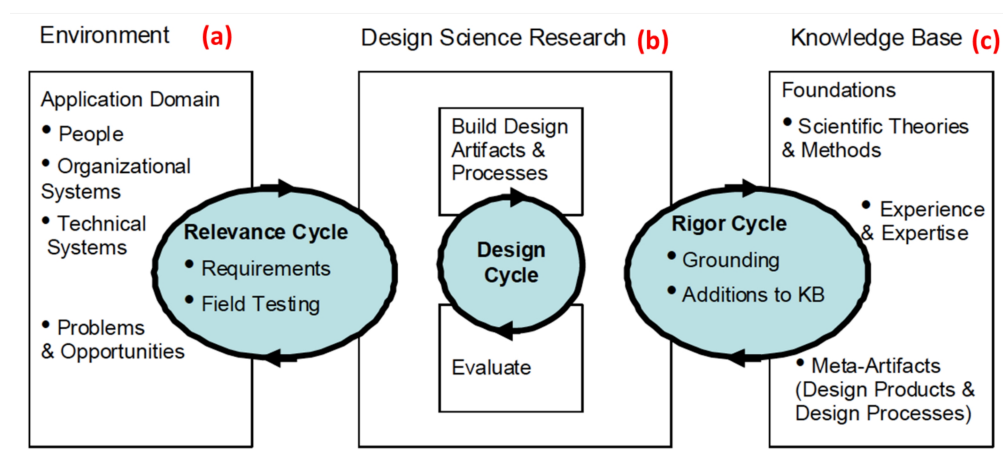


Figura 4.1: Ciclos do DSR Fonte: Hevner (2007)

Segundo Hevner (2007), o Ciclo Relevância conecta o contexto de pesquisa (a) com as atividades do DSR (b). O principal objetivo do Ciclo Relevância é detectar problemas e oportunidades para oferecer soluções que sejam relevantes para contexto estudado. O Ciclo Rigor conecta as atividades do DSR (b) com a Base de Conhecimento (c) fundamentando a pesquisa com teorias e métodos científicos. A Base de Conhecimento (c) fornece ainda processos e artefatos provenientes de experiências anteriores, podendo ser reaproveitados ou readequados para outros contextos. O Ciclo Rigor deve garantir que as atividades de DSR sejam fundamentadas no conhecimento e rigor científico (b), e em contrapartida, que o conhecimento atualizado após aplicação prática re-alimente à Base de Conhecimento (c). O Ciclo de *Design* (localizado no centro da Figura 4.1) opera entre as atividades DSR que consiste em (1) construir e (2) avaliar os processos e artefatos projetados.

### 4.2.3 Trabalhos Relacionados

De acordo com Dittrich et al. (2008), o CMD tem sido aplicado para apoiar diferentes temas de pesquisa na área de ES, tais como projetos de software flexíveis e adaptáveis; desenvolvimento ágil de aplicações para governo eletrônico (e-gov); e integração do Dxl em processos de desenvolvimento de software. Recentemente, Al-Baik e Miller (2014) implementaram o CMD para investigar os prováveis motivos por trás da baixa taxa de sucesso de iniciativas Lean em

organizações de TI. Como resultado, os autores propuseram um modelo para classificar resíduos de TI. Ardito et al. (2014) aplicaram o CMD com o objetivo de averiguar o impacto dos métodos de UX nas práticas de desenvolvimento de software em empresas do sul da Itália. Os autores confirmaram que muitas empresas negligenciavam ou não estavam dando a devida importância para questões de UX e para os fatores de usabilidade durante o processo de desenvolvimento de software.

Em relação ao DSR, Adikari, McDonald e Campbell (2009) adotaram o *framework* proposto por Hevner et al. (2004) para desenvolver uma abordagem integrando o DCU na Engenharia de Requisitos do DAS. Implementando a abordagem em dois projetos ágeis, os autores confirmaram junto às equipes de desenvolvimento que a proposta de integração foi considerada fácil de aprender e fácil de usar. Rodríguez, Kuvaja e Oivo (2014) também aplicaram *framework* de Hevner et al. (2004) em seus projetos de pesquisa para explorar os principais elementos que caracterizam a combinação da metodologia Lean com demais métodos ágeis no desenvolvimento de software. Os autores relatam como o DSR pode ser aplicado em engenharia de software empírica, destacando em lições aprendidas que tal método de pesquisa tem um bom potencial para apoiar a colaboração entre a indústria e a academia.

Algumas abordagens combinando AR e DSR foram propostas para conduzir pesquisa na área de SI. Baskerville, Pries-Heje e Venable (2009), por exemplo, propuseram a abordagem de pesquisa chamada SoftDSR que combina a metodologia SSM (*Soft Systems Methodology*) derivada da AR com o DSR. Sein et al. (2011) apresentaram a abordagem ADR (*Action Design Research*), que define quatro fases de pesquisa: (1) Formulação do problema; (2) Construção, intervenção e avaliação; (3) Reflexão e aprendizagem; e (4) Formalização da aprendizagem. E Wieringa e Morali (2012) propuseram um método de pesquisa chamado TAR (*Technical Action Research*), que usa a AR somente para avaliar os artefatos propostos no processo de DSR.

### 4.3 SoftCoDeR: a abordagem de pesquisa

Considerando que a experiência junto a indústria envolveria a construção de artefatos para apoiar o trabalho das equipes ágeis e de UX durante o processo de desenvolvimento de software, adotou-se uma abordagem de pesquisa chamada SoftCoDeR (*Software Cooperative Design Research*) para conduzir o trabalho de pesquisa-ação envolvendo profissionais da empresa parceira do projeto de P&D. Embora a abordagem SoftCoDeR tenha sido desenvolvida inicialmente para o projeto em questão, considera-se que tal iniciativa é uma contribuição parcial deste trabalho, pois esta poderá ser aplicada em diferentes contextos.

A abordagem SoftCoDeR combina o método de pesquisa-ação CMD proposto por Dittrich (2002) com o *framework* de pesquisa DSR proposto por Hevner et al. (2004). Observou-se que o CMD e o DSR são métodos compatíveis, pois ambos possuem caráter intervencionista e estão intimamente relacionados com a resolução de problemas envolvendo a avaliação das soluções propostas (COLE et al., 2005). Entretanto, ambos possuem focos distintos. Enquanto o CMD tem foco na resolução de problemas por meio de mudança social e organizacional (método centrado na pesquisa através da ação), o DSR tem foco na resolução de problemas, criando e posicionando artefatos em ambiente natural (método centrado na pesquisa através do *design*)

(BASKERVILLE, 2008).

Analisando-se as características de cada método de pesquisa, vislumbrou-se a possibilidade de uni-los em uma única abordagem de pesquisa para funcionarem de forma complementar. O CMD contribui para um processo de pesquisa estruturado, que define claramente as três fases de pesquisa estabelecidas para (1) compreensão da prática, (2) deliberação de melhorias, e (3) implementação e avaliação das melhorias. Além disso, uma das características do CMD é encorajar ciclos iterativos curtos para implementar as melhorias na prática, tornando-o um método bem dinâmico e muito aderente aos princípios ágeis. Enquanto que o DSR, contribui para encorajar o uso de métodos rigorosos para a construção e avaliação de artefatos de valor concebidos para resolver problemas relevantes em contextos bem específicos. A Figura 4.2 apresenta a abordagem SoftCoDeR baseada nas principais características dos dois métodos de pesquisa (CMD e DSR).

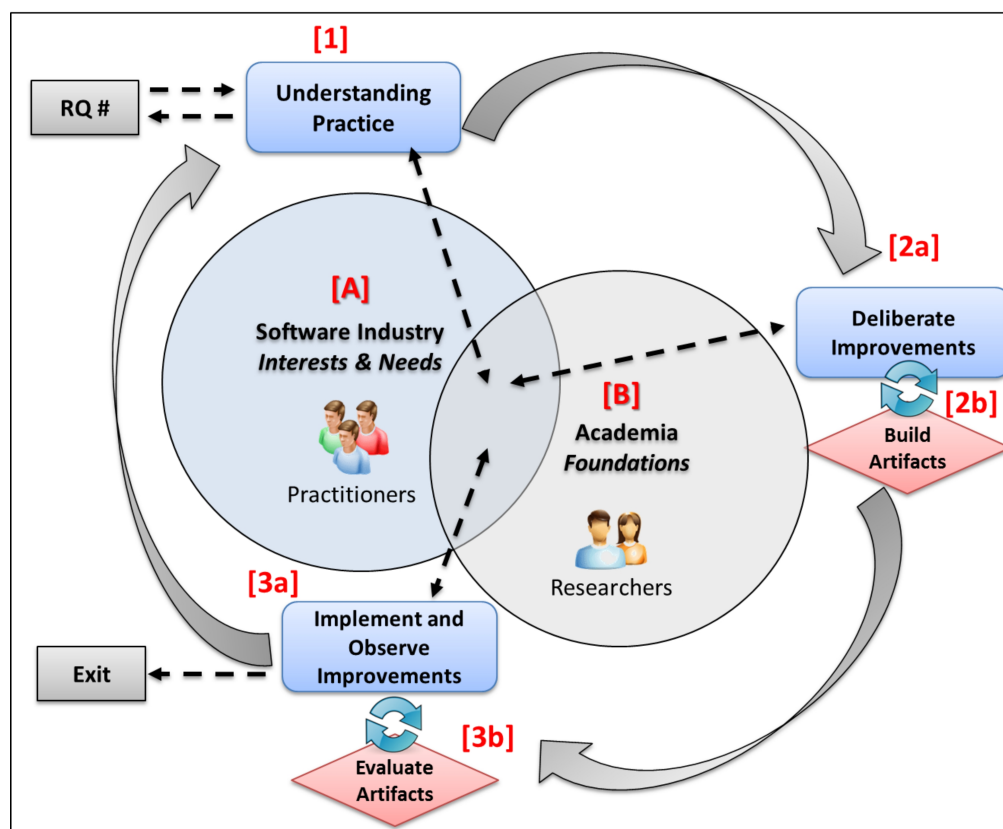


Figura 4.2: SoftCoDeR. Fonte: Choma, Zaina e Silva (2015)

A abordagem SoftCoDeR combina os métodos CMD e DSR por meio de um ciclo iterativo e evolucionário, herdando as três etapas do CMD que são *Entendendo a Prática* (1), *Deliberando Melhorias* (2a) e *Implementando e Observando as Melhorias* (3a); e as duas atividades do DSR que são *Construir Artefatos* (2b), e *Avaliar Artefatos* (3b). A intersecção ( $A \cap B$ ) das necessidades e interesses da *Indústria* (A) com os fundamentos da *Academia* (B) fornecem as orientações necessárias para todas as etapas do ciclo de investigação.

O primeiro passo do SoftCoDeR consiste em compreender as práticas (1) do ponto de vista

dos profissionais, com o objetivo de identificar as necessidades de novas soluções ou melhorias que devem estar alinhadas com os interesses da *Indústria* (A). Os fundamentos da *Academia* (B) irão contribuir para que o pesquisador identifique conceitos usados (ou não) na prática, em contraste com outras teorias e tecnologias que tem sido empregadas na prática por outras empresas. Estudos etnográficos e entrevistas são as técnicas qualitativas recomendadas para coletar dados sobre as práticas de desenvolvimento de software. Uma *questão de pesquisa* (RQ) pode ser antecipada pelos pesquisadores ou formulada durante a primeira etapa de investigação, emergindo de situações observadas na prática.

Uma vez definida a questão de pesquisa, inicia-se a próxima etapa que é a deliberação de melhorias (2a), na qual pesquisadores e profissionais reúnem-se para analisar e discutir os dados colhidos a partir daquilo que foi levantado da prática, construindo cenários que reproduzem a situação-problema que será cuidadosamente examinada. As soluções deverão refletir a experiência da indústria com os conhecimentos da academia. Artefatos de valor, necessários para apoiar o trabalho dos profissionais, também serão identificados considerando-se a função técnica (do ponto de vista acadêmico), e a função pragmática (do ponto de vista da indústria). Na sequência, esses artefatos devem ser elaborados e construídos (2b) pelos pesquisadores junto com profissionais da empresa parceira. Esta etapa é cíclica para permitir novas discussões para refinar os artefatos quando necessário.

Na terceira etapa, os membros da indústria definem e escolhem um projeto prático através do qual será possível implementar e observar as melhorias (3a), e avaliar os artefatos (3b). Os artefatos poderão ser avaliados em termos de praticidade, eficiência ou eficácia, por meio de diferentes métodos de investigação, tais como experimentos de laboratório, experimentos de campo, simulações, estudos de caso, pesquisas e estudos etnográficos, que podem ser aplicados individualmente ou combinados (VENABLE, 2006). Em alguns casos, uma validação experimental é altamente recomendável antes de colocar as melhorias na prática, especialmente para minimizar o impacto da introdução da nova tecnologia (artefato) na prática (IIVARI; VENABLE, 2009).

A fase de avaliação (3a e 3b) deve fornecer *feedback* sobre as melhorias implementadas e os artefatos construídos. Novos ciclos podem ser necessários para refinar os artefatos, ou mesmo sugerir novas melhorias. A partir das lições aprendidas, novas perguntas podem surgir demandando um novo ciclo. Assim, o ciclo pode ser incremental, adicionando a informação para cada novo ciclo ou espiral, uma vez que a informação podem surgir durante a fase de avaliação.

## 4.4 Construção e avaliação dos artefatos de apoio

Os artefatos propostos para dar suporte à abordagem ScrumUX foram desenvolvidos e validados em dois ciclos da abordagem SoftCoDeR aplicados em uma indústria de software desenvolvedora de sistemas ERP (Enterprise Resource Planning). A referida empresa tem sido parceira em um projeto de P&D, cujo tema da pesquisa é “Design de Interação do Usuário para Sistemas ERP”. O objetivo do projeto é a adoção de práticas de DxD no processo de desenvolvimento de software visando melhorar a usabilidade dos sistemas ERP.

A empresa parceira atua há 28 anos no ramo de desenvolvimento de sistemas ERP para diversos segmentos do mercado, tais como Construção Civil, Manufatura, Logística, Combustível

tível e Agronegócio; possui 700 funcionários, cerca de 2.000 clientes e mais de 50.000 usuários. Nos últimos anos, os executivos da empresa têm concentrado os seus esforços para evoluir seus processos de desenvolvimento de software incorporando as melhores práticas para o desenvolvimento de interfaces de usuário através do *design* centrado no usuário. Eles acreditam que a integração de práticas de DxD no desenvolvimento de sistemas de ERP é um aspecto essencial para preencher uma lacuna importante dentro do seu processo de desenvolvimento, podendo ser revertido numa vantagem competitiva dentro do segmento que atuam.

Os pesquisadores envolvidos no projeto, inclusive a autora dessa dissertação, tiveram a oportunidade de trabalhar em estreita colaboração com profissionais de diversas áreas do desenvolvimento de software, atuando em várias ações tais como propor (i) técnicas para identificar as necessidades dos usuários; (ii) normas e diretrizes para o desenvolvimento de interfaces de usuário; (iii) métodos eficazes de avaliação de usabilidade para sistemas ERP; e (iv) mecanismos para melhorar a comunicação de problemas de usabilidade entre designers e equipe de desenvolvimento.

A seguir, descreve-se os dois ciclos SoftCoDeR implementados em contexto industrial para construir e avaliar os artefatos que, posteriormente, serão selecionados para dar suporte na abordagem ScrumUX.

#### 4.4.1 Primeiro ciclo

Durante uma das reuniões, que costumavam ocorrer periodicamente na empresa parceira, líderes de projetos ágeis relataram aos pesquisadores que a comunicação entre as equipes UX e Scrum estava difícil, principalmente, por causa das diferenças de vocabulário empregado por eles. Levando em conta esta questão, os pesquisadores buscaram conceitos que poderiam afinar a comunicação entre as duas equipes. Idealizando possíveis soluções, os pesquisadores concentraram-se em dois conceitos: personas e as heurísticas de Nielsen. Considerando a situação-problema, bem como a utilização dos conceitos mencionados, definiu-se duas questões de pesquisa para direcionar as investigações do primeiro ciclo SoftCoDeR:

- RQ1: Os conceitos de personas e heurísticas de Nielsen podem nivelar a compreensão sobre aspectos de usabilidade entre *designers* de UX e desenvolvedores (programadores e testadores)?
- RQ2: De que forma os conceitos de personas e heurísticas de Nielsen poderiam ser usados como um vocabulário comum na comunicação entre as equipes de UX e Scrum?

##### 4.4.1.1 Entendendo a prática

Especialistas de UX reconhecem as personas e as heurísticas de Nielsen como boas práticas para se manter o desenvolvimento de software focado nas necessidades do usuário e nos fatores de usabilidade (FØLSTAD; LAW; HORNBÆK, 2012) (CABALLERO; MORENO; SEFFAH, 2014).

Personas são utilizadas na indústria de desenvolvimento de software para descrever grupos de usuários reais por meio da criação de arquétipos hipotéticos. Personas podem orientar tanto

o desenvolvimento de cenários de interação como podem ser usadas para descrever as tarefas típicas em testes de usabilidade (COOPER et al., 2014).

A aplicação da técnica de personas provou que, projetar um pequeno conjunto de personas pode atender um número significativo de usuários, agrupados conforme objetivos e características semelhantes. A construção de personas deve ser fundamentada na pesquisa com usuários reais, mas isso pode exigir um longo tempo para coleta e análise de dados. Contudo, personas tem sido adaptadas para serem mais leves e mais enxutas com a finalidade de ampliar sua praticidade em contexto ágil de desenvolvimento de software (MILLER; WILLIAMS, 2006).

As heurísticas de Nielsen, que comumente são utilizadas para apoiar a análise crítica em inspeções de usabilidade (avaliação heurística), tem sido usadas para direcionar o *design* de interfaces interativas, e também para confirmar problemas identificados em testes de usabilidade (FØLSTAD; LAW; HORNBÆK, 2012).

Com o objetivo de entender o que tais conceitos poderiam representar na prática diária dos profissionais [1], e para responder a primeira questão de pesquisa (RQ1), os pesquisadores conduziram, primeiramente, um *workshop* intitulado “Heurísticas de Usabilidade para sistemas ERP”, onde conceitos sobre heurísticas de Nielsen e personas foram apresentados para 59 profissionais de diferentes áreas: desenvolvedores (36), testadores (12), analistas (5), líderes técnicos (4) e arquitetos de software (2).

A maioria dos participantes tinha menos de cinco anos de experiência em desenvolvimento de software, e 56% tinha menos de cinco anos de experiência no desenvolvimento de sistemas de ERP. Apenas 11 participantes (18,64%) tinham mais de dez anos de experiência em desenvolvimento de software, e 9 participantes (15,25%) tinham experiência no desenvolvimento de sistemas de ERP. Durante o *workshop* todos os participantes puderam trabalhar com os conceitos através de atividades práticas incluindo a avaliação heurística de dois módulos de sistemas ERP produzidos pela empresa.

Algumas semanas depois do *workshop*, os pesquisadores entrevistaram 10 profissionais que haviam participado do evento, com o objetivo de investigar (i) os conhecimentos e habilidades obtidos a partir da oficina; e (ii) a contribuição da oficina na mudança de ponto de vista dos participantes sobre os aspectos de usabilidade no desenvolvimento de software no contexto de sistemas de ERP. A Tabela 4.1 apresenta o perfil dos entrevistados.

As entrevistas foram gravadas em áudio com a prévia autorização dos entrevistados, e posteriormente transcritas para um relatório. Durante um dia, realizou-se as entrevistas semi-estruturadas, incluindo as questões que resumem três aspectos: (i) o ponto de vista em relação aos aspectos de usabilidade, (ii) a percepção sobre a aplicação prática das heurísticas de Nielsen e personas durante o desenvolvimento de sistemas de ERP, e (iii) a avaliação das melhorias em sistemas de ERP considerando aspectos de usabilidade.

As respostas dos entrevistados foram analisadas qualitativamente e classificadas a partir dos três aspectos destacados. A Tabela 4.2 apresenta uma visão geral dos comentários dos participante analisados pelos aspectos observados. No geral, observou-se que os assuntos abordados na oficina influenciaram positivamente os profissionais quanto a importância da usabilidade em sistemas ERP. Observou-se também que as heurísticas de Nielsen de alguma forma fixaram vários tópicos de usabilidade na mente dos participantes.

Tabela 4.1: Perfil dos entrevistados

Participantes	Função	Experiência com desenvolvimento de software	Experiência com desenvolvimento de sistemas ERP	Tempo de entrevista
P1	Testador	4 anos	2 anos	17min39s
P2	Desenvolvedor	8 anos	5 anos	16min53s
P3	Testador	13 anos	9 anos	22min45s
P4	Desenvolvedor	11 anos	4 anos	15min23s
P5	Testador	3 anos	3 anos	17min02s
P6	Desenvolvedor	6 anos	3 anos	10min08s
P7	Arquiteto	9 anos	9 anos	20min42s
P8	Testador	4 anos	2 anos	16min46s
P9	Arquiteto	27 anos	27 anos	19min06s
P10	Desenvolvedor	4 anos	2 anos	15min12s

Com base nos resultados de entrevistas, encontraram-se as evidências para responder a primeira pergunta (RQ1), ou seja, a maioria dos participantes considerou que personas e heurísticas de Nielsen podiam orientar o desenvolvimento de produtos e também ser usados como um vocabulário comum entre os desenvolvedores, testadores e *designers* de UX.

#### 4.4.1.2 Deliberando melhorias e construindo artefatos

A partir dos resultados da etapa anterior, os pesquisadores juntamente com líderes de projetos e o *designer* de UX discutiram a possibilidade do uso de personas e heurísticas de Nielsen no processo ágil de desenvolvimento de software [2a]. Um dos líderes de projeto argumentou que seria necessário um esforço muito grande para construir personas caso tivessem que realizar pesquisa prévia dos potenciais usuários, que poderia ser inviável. Contudo, descobriu-se que a empresa mantinha um conhecimento do perfil dos usuários no departamento de pós-venda. Os pesquisadores, então, sugeriram que equipe poderia projetar hipóteses de personas a partir deste conhecimento. Além disso, concluiu-se que muitas personas poderiam ser reaproveitadas para outros projetos.

Outra questão levantada pelos profissionais estava relacionada com a forma que problemas de usabilidade identificados em testes com usuários vinham sendo relatados. Os problemas de usabilidade costumavam ser relatados oralmente ou por anotações em linguagem natural. Segundo os profissionais, a falta de um mecanismo formal para comunicar os problemas de usabilidade às equipes Scrum, muitas vezes, resultava em mal-entendidos entre as duas equipes (Scrum e UX). A partir das discussões sobre as questões levantadas (respondendo RQ2), os pesquisadores e profissionais idealizaram e construíram dois artefatos [2b] baseados em personas e heurísticas de Nielsen:

**a) *Lean Persona*:** artefato para representar um conjunto de usuários finais, tratado por Hipótese de Persona (HP). Baseando-se na estrutura simplificada apresentada por Miller e Williams (2006), definiu-se um modelo para descrever as HP composto pelos seguintes elementos:



Tabela 4.2: Visão geral dos comentários dos participante por aspecto observado

Aspecto observado	Comentário dos participantes	Análise
(i) Ponto de vista em relação aos aspectos de usabilidade	<p>“(...) agora, a gente consegue olhar melhor, direcionar melhor quando olha uma tela nova, consegue identificar [problemas de usabilidade].” [P8].</p> <p>“(...)[minha visão] melhorou bastante, até pelo fato da gente prestar mais atenção e saber que às vezes um ponto ou outro é uma melhoria no sistema que a gente deve fazer, e que muitas vezes passa despercebido, digamos assim, e com o workshop deu para ter uma visão melhor das coisas. [P4].</p> <p>“(...) saber realmente que a heurística está baseada no mercado, então eu consigo ver com base na atualidade realmente como os usuários [personas] estão vendo o sistema. Conseguiu me trazer a visão de como o mercado está olhando para o nosso produto.” [P3].</p> <p>“(...), estamos fazendo as mudanças de mensagens de erro (...), então já tive que pensar em como o usuário enxerga, como o cliente vê o sistema e ai agregou, melhorou na minha análise entre outros pontos também.” [P2].</p>	<p>A maioria dos entrevistados diz que eles estão mais cuidadosos, prestando atenção em detalhes simples, pois acreditam que assim podem melhorar o sistema. Eles passaram a se preocuparem mais com as percepções dos usuários (personas) em relação ao sistema.</p>
(ii) Percepção sobre a aplicação prática das heurísticas de Nielsen e personas durante o desenvolvimento de sistemas de ERP	<p>“(...), as heurísticas mostrou-me que as minhas preocupações não devem ser apenas sobre os aspectos funcionais, eu também tenho que considerar a interface, a disposição do conteúdo, padrões, etc. (...) depois de visualizar as personas, (...) eu me tornei mais crítica sobre as questões de interação do usuário.” [P3].</p> <p>“(...) As heurísticas poderiam ser utilizadas como uma lista de verificação durante os testes de software.” [P7].</p> <p>“Agora toda vez que eu vou fazer os testes, principalmente os de usabilidade, eu procuro ler mais sobre as heurísticas.” [P1].</p> <p>“(...). Teve uma tela nova que foi desenvolvida, que tinha a parte de legendas com cores, e era uma cor que nunca foi usada, percebi e falei com o desenvolvedor “olha eu acho que essa cor não é interessante porque, o usuário não está acostumado”, mostrei outra tela e optamos por um padrão.” [P8].</p>	<p>Todos os entrevistados concordaram com a utilidade das heurísticas e personas na prática do desenvolvimento de ERP. Eles acreditam que as heurísticas poderiam guiar a verificação da usabilidade e ajudar o desenvolvedor na busca de alternativas para resolver problemas de usabilidade. As personas têm uma influência psicológica, porque eles têm uma pessoa concreta que representa os usuários.</p>
(iii) Avaliação das melhorias em sistemas de ERP considerando aspectos de usabilidade	<p>“(...), algumas pessoas, alguns profissionais acham que por ser ERP, por ser um sistema grande não precisa tanto de uma usabilidade assim. Eles acham que o cliente está acostumado com o sistema difícil ou com o sistema não tendo uma facilidade de trabalhar. E eu já não concordo eu acho que mesmo sendo um ERP um sistema grande, um sistema difícil de lidar com o processo; penso que o nosso trabalho é facilitar isso, através das heurísticas, por exemplo.” [P1]</p> <p>“(...), o sistema é realmente complexo, não tenho dúvida quanto a isso, mas (...) sempre tem como melhorar, resta ver a viabilidade, porque tem questões que podem até envolver nível estrutural para melhorar uma questão de UX.” [P9]</p>	<p>Embora reconheçam a complexidade dos sistemas ERP, os entrevistados acreditam que é possível tornar a experiência do usuário mais agradável.</p>

foto, nome, idade, área de especialização, conhecimento (competências e habilidades), objetivos (motivações e preocupações), e padrões de uso (contexto de uso e hábitos). A Figura 4.3 apresenta o modelo para descrever as HP definido juntamente com os profissionais da empresa, incluindo as orientações para o preenchimento de cada item.


	<b>Nome:</b> <i>Digite nome fictício para a persona</i>	<b>Grupo:</b> <i>Nome do grupo de usuários que a persona pertence</i>
	<b>Dados demográficos e pessoais:</b> <i>idade, profissão, etc.</i>	
<b>Conhecimentos, competências e habilidades:</b> <i>Grupo de informações reais, mas generalizada sobre as capacidades da persona.</i>		
<b>Metas, motivos e preocupações:</b> <i>Descrever as reais necessidades dos usuários no grupo de usuário representado pela persona. Se existirem vários agrupamentos, escreva uma persona para cada agrupamento.</i>		
<b>Padrões de utilização:</b> <i>Desenvolver uma compreensão detalhada sobre as funções mais utilizadas e a frequência de uso. Procure qualquer desafio que o sistema deva ajudar o personagem a superar dificuldades. Insira observações relacionadas à aprendizagem e ao estilo de interação para novas funcionalidades. A persona explora o sistema para encontrar uma nova funcionalidade ou tem necessidade de orientação?            (Descreva de forma breve e precisa).</i>		

Figura 4.3: Modelo de Lean Persona

b) **Protocolo para a comunicação de soluções e recomendações de design de interação:** este artefato é gerado a partir da análise dos resultados de testes com usuários. Em primeiro lugar, as questões identificadas durante as sessões de testes de usabilidade são listadas, removendo-se as questões em duplicidade (os mesmos problemas apontados por mais de uma persona) e os falsos positivos (questões que não foram consideradas reais). Em segundo lugar, as questões são analisadas, e em seguida, soluções e/ou recomendações são propostas para cada uma delas. Em terceiro lugar, identifica-se as personas afetados em cada questão, e por último, as questões deverão ser mapeadas com as heurísticas de Nielsen. Formalizando, os itens do Protocolo podem ser representados pela Equação 4.1.

#### 4.4.1.3 Implementando, observando as melhorias e avaliando artefatos

Os pesquisadores e os profissionais (UX designer e líderes de projetos) planejaram dois estudos de caso para avaliar o uso das Lean Personas e do Protocolo de Comunicação [3b]. Com a finalidade de comparar resultados, os dois artefatos seriam implementados somente no segundo estudo de caso [3a]. Todos os profissionais envolvidos em ambos os estudos de caso conheciam tanto o conceito de personas como as heurísticas de Nielsen, pois todos haviam participado do *workshop* “Heurísticas de Usabilidade para sistemas ERP”. Durante os estudos de caso,

$$\text{Protocolo} = \{\text{item: item} \equiv \text{descrição, HN}_{\text{subset}}, \text{HP}_{\text{subset}}\} \quad (4.1)$$

Onde:

$$\text{descrição} \in \{\text{recomendação, solução}\} \quad (4.1a)$$

$$\text{HN}_{\text{subset}} = \{\text{nh: nh} \in \text{HN}\} \quad (4.1b)$$

$$\text{HN} = \{H_1, H_2, \dots, H_{10}\} \quad (4.1c)$$

$$\text{HP}_{\text{subset}} = \{\text{hp: hp} \in \text{HP}\} \quad (4.1d)$$

$$\text{HP} = \{\text{hipPersona}_1, \dots, \text{hipPersona}_j\} \quad (4.1e)$$

NH = Heurística de Nielsen

HP = Hipótese de Persona

realizaram-se testes de usabilidade no submódulo de Cadastro de Colaboradores<sup>1</sup> do módulo de Recursos Humanos.

No primeiro estudo de caso, os pesquisadores apenas observaram as atividades sem intervenção. O *designer* de UX planejou e executou os testes de usabilidade atuando como moderador de teste. Foram recrutados cinco usuários de diferentes perfis. As sessões foram gravadas com prévia autorização dos participantes, por meio de uma câmera simples. As questões identificadas durante as sessões de teste foram listadas em linguagem natural, e em seguida, entregues pelo *designer* de UX para a equipe Scrum durante uma reunião informal. A equipe Scrum ajustou somente seis dos dez problemas relatados. Contrariando a opinião do designer de UX, os desenvolvedores argumentaram que as demais questões não deveriam ser consideradas como problemas de fato.

No segundo estudo de caso, a versão modificada do protótipo funcional foi testada novamente. Os pesquisadores participaram junto com o profissional de UX desde o planejamento do teste, implementando os dois artefatos produzidos e observando melhorias [3a]. A Figura 4.4 apresenta a abordagem que define o processo que foi utilizado para condução dos testes com usuários, contemplando as fases de planejamento, execução e análise, incluindo a geração do Protocolo de Comunicação das soluções/recomendações de usabilidade baseado nos resultados dos testes.

Conforme apresentado na Figura 4.4, na fase de planejamento (a), a equipe de UX seleciona os artefatos (protótipos baixa/média/alta fidelidade) e as hipóteses de personas mais relevantes para os testes). As etapas são executadas para produzir o plano de teste (saída da fase): (i) listar as tarefas críticas para cada hipótese de persona; (ii) determinar as métricas quantitativas e/ou qualitativas de acordo com o objetivo de teste (por exemplo, tempo gasto na tarefa, dificuldades para encontrar recursos, uso de mensagens apropriadas, etc.); e (iii) recrutar pelo menos 5 usuários, cobrindo de forma equilibrada todas as hipóteses de personas. O número de usuários estabelecido para o teste segue a sugestão do estudo de Borsci et al. (2013).

Na fase de execução (b), o especialista de UX realiza o teste em um computador pré-configurado, ou por conveniência, no computador do usuário. Softwares para capturar imagens ou para a observação *online* podem ser empregados para melhorar a coleta de dados durante os testes e permitir posteriormente uma análise mais detalhada.

Na fase de análise (c), o moderador de teste e/ou especialista de UX faz a revisão e a análise

<sup>1</sup>Protótipo funcional (alta fidelidade) orientado a navegação Web, cujo objetivo era a atualização de cadastro dos colaboradores do módulo de Recursos Humanos do Sistema de Gestão Empresarial.

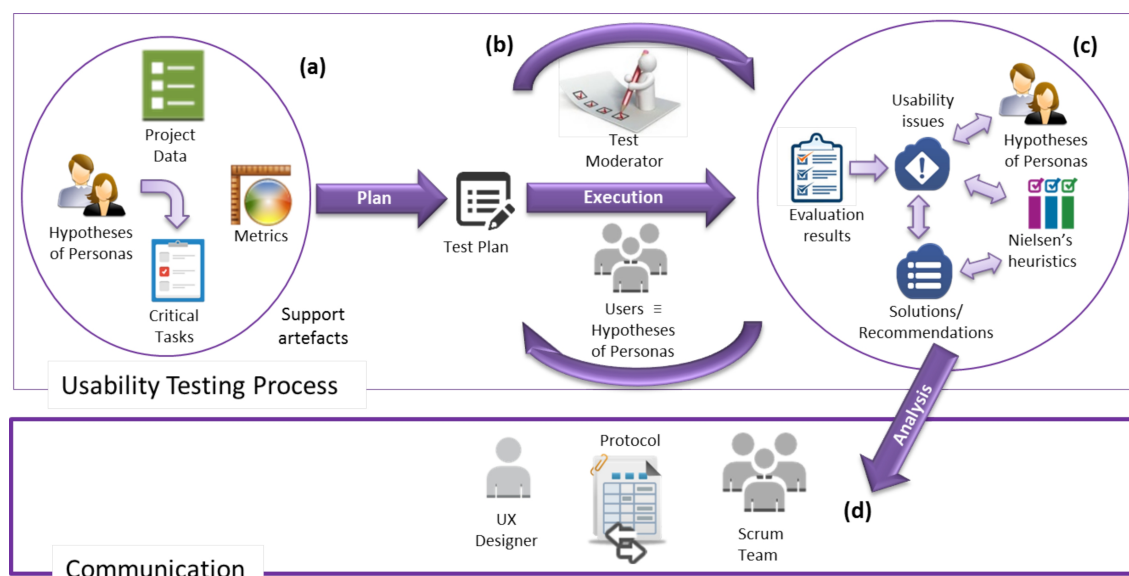


Figura 4.4: Abordagem para conduzir teste com usuários e reportar soluções/recomendações de usabilidade. Fonte: Choma, Zaina e Beraldo (2015)

das notas de teste e, eventualmente, as imagens gravadas durante o teste, executando as quatro etapas: (i) listar as questões capturadas, removendo os problemas duplicados e falsos positivos; (ii) analisar cada questão apontada e propor respectivas soluções e/ou recomendações; (iii) identificar as personas que podem ser afetadas por cada questão; e, finalmente, (iv) mapear cada questão a heurísticas de Nielsen. Como resultado das quatro etapas, o Protocolo é gerado e comunicado na fase seguinte (d)

Seguindo o Protocolo de Comunicação, o especialista de UX relata os resultados à equipe Scrum apresentando soluções e/ou recomendação para orientar futuros ajustes. Os resultados podem ser discutidos durante uma reunião de planejamento ou em reuniões diárias de acordo com o atendimento de prioridades. A equipe avalia os itens relatados, vendo os efeitos deles, as personas afetadas e as heurísticas da Nielsen a serem atendidas quando forem tratados. Após avaliação, a equipe Scrum escreve as histórias de usuários, considerando os itens listados no Protocolo de Comunicação.

Seguindo os procedimentos para realizar os testes com usuários, o *designer* de UX desenvolveu três personas diferentes coletando informações no departamento comercial. As personas criadas representavam três níveis diferentes de expertise em relação ao uso da tecnologia (níveis básico, intermediário e avançado). A Figura 4.5 apresenta a hipótese de persona pertencente ao grupo de usuários de nível avançado que foi desenvolvida pelo *designer* de UX seguindo o modelo proposto (Lean Personas).

As hipóteses de persona foram usadas para recrutar seis participantes (dois participantes por persona). É importante destacar que, nenhum participante recrutado havia participado do teste anterior. Os testes foram realizados e os resultados foram analisados em um único dia. As sessões foram gravadas com prévia autorização dos participantes, por meio de um software de captura de tela. Foram identificadas três novas questões junto com as quatro questões que não


	<b>Name:</b> <i>Eric Oliveira</i>	<b>Expertise:</b> <i>Avançado</i>
	<b>Demographics:</b> <i>32, Arquiteto de Sistemas</i>	
<b>Conhecimentos, competências e habilidades:</b> <i>Eric trabalhou há oito anos na empresa, pesquisando novas tecnologias, processos e ferramentas para desenvolvimento de software. Ele é graduado em Análise de Sistemas.</i>		
<b>Metas, motivos e preocupações:</b> <i>Ele gostaria que atualizar o seu endereço no Cadastro de Empregados rapidamente, em poucos passos.</i>		
<b>Padrões de utilização:</b> <i>Ele teve muita experiência com a tecnologia, devido a sua profissão e hobby. Ele usou duas vezes o sistema atual para atualizar seu número de telefone e endereço.</i>		

Figura 4.5: Exemplo Lean Persona desenvolvida pelo *designer* de UX

havam sido ajustadas pelos desenvolvedores na Sprint anterior.

Seguindo os passos para analisar questões de usabilidade e gerar o Protocolo de Comunicação (baseado na Equação 4.1), o designer UX descreveu as soluções e recomendações de acordo com o modelo apresentado na Figura 4.6. O relatório redigido no formato do Protocolo foi entregue à equipe Scrum. Diferentemente do primeiro estudo de caso, a equipe Scrum concordou com todas as questões relatadas e realizou todos os ajustes recomendados.

Soluções e/ou recomendações de design	Personas			Heurísticas de Nielsen
	A	B	C	
Todos os usuários necessitaram de ajuda para saber o significado das siglas de documentos solicitados no formulário. Recomenda-se evitar o uso de acrônimos, caso contrário inserir links de ajuda com uma explicação mais sucinta e uma linguagem menos técnica.	x	x	x	[H6]
LEGENDA:   A = Básico   B = Intermediário   C = Avançado				

Figura 4.6: Exemplo de recomendação para um dos problemas identificados utilizando o Protocolo de Comunicação

Dias depois, os pesquisadores coletaram a opinião dos profissionais que haviam participado do segundo estudo de caso a respeito dos novos artefatos. Os membros da equipe Scrum disseram que as heurísticas relacionadas no Protocolo de Comunicação permitiu-lhes compreender qual era o impacto das questões identificadas em termos de usabilidade. Disseram ainda que, conseguiam

avaliar a gravidade das questões observando o número e os tipos de personas que estavam sendo afetadas, assim, puderam priorizar melhor os ajustes a serem executados.

#### 4.4.2 Segundo ciclo

Quando o Protocolo de Comunicação foi colocado em prática em outros projetos com a participação de POs, observou-se que alguns tinham dificuldade para compreender o artefato proposto. Observando a prática, os pesquisadores descobriram que, para o PO, o uso de *User Stories* (US) era mais natural. Considerando-se esta questão, os pesquisadores sugeriram uma adaptação de US incorporando a visão da experiência do usuário. O segundo ciclo de pesquisa foi iniciado, a fim de responder à seguinte pergunta:

- RQ3: Como conceitos de personas e heurísticas de Nielsen poderiam ser incorporados às histórias de usuários?

##### 4.4.2.1 Entendendo a prática

Na primeira fase, os pesquisadores fizeram um levantamento na literatura técnica, com o objetivo de investigar o uso de US em práticas ágeis, e fizeram um estudo etnográfico, a fim de saber como os POs estavam escrevendo as US. Verificou-se que os POs escreviam as histórias de usuários no formato mais tradicional, tal como o modelo de Cohn (2009).

Na revisão da literatura, algumas recomendações foram encontradas em relação a como lidar com experiência do usuário e usabilidade junto com histórias do usuário. Moreno e Yagüe (2012), por exemplo, identificaram três maneiras de incorporar os requisitos funcionais de usabilidade com US: (1) adicionar novas histórias para representar os requisitos que são diretamente derivadas de usabilidade (chamadas de histórias de usabilidade); (2) adicionar ou modificar as tarefas em US (detalhando quando necessário); e (3) adicionar ou modificar os critérios de aceitação. Os critérios de aceitação são comumente adicionados às US para orientar o teste de aceitação. O teste de aceitação é o processo de verificar se as histórias foram desenvolvidas da maneira que o cliente espera que funcione, ou seja, é uma forma de garantir a funcionalidade dos requisitos de software (COHN, 2009). Moreno e Yagüe (2012) destacam que requisitos de usabilidade funcionais complementam os requisitos tradicionais, e portanto, histórias de usabilidade podem ser documentadas como US. Com o objetivo de mitigar os conflitos entre equipe ágil e UX na tarefa de comunicar requisitos de interação, Barksdale e McCrickard (2010) recomendaram mapas conceituais para projetar uma visão completa do usuário a partir do detalhamento das conexões entre US e cenários. Sohaib e Khan (2010) também recomendam o uso de cenários juntamente com US na fase de exploração, e a avaliação heurística durante os testes de aceitação.

##### 4.4.2.2 Deliberando melhorias e construindo artefatos

Considerando o que foi observado da prática e as recomendações da literatura, os pesquisadores propuseram às equipes uma nova gramática para US incorporando personas e heurísticas de Nielsen para descrever requisitos de interação (respondendo RQ3). O novo modelo foi nomeado como UserX Story. A Figura 4.7 mostra a primeira versão do artefato proposto [2b].

<b><i>UserX Story (versão 1.0)</i></b>
<b>Como &lt;Persona&gt; Eu quero / preciso &lt;objetivo&gt;, para atender &lt;Heurísticas de Nielsen&gt;</b>

Figura 4.7: UserX Story - primeira versão

Na primeira versão do artefato, a gramática tradicional de US foi modificada, substituindo-se: (1) <tipo de usuário> por <personas>; e (2) <alguma razão> por <heurística (s) de Nielsen>.

O artefato foi submetido à apreciação da equipe de UX, Scrum Masters e de alguns POs. Novas ideias surgiram a partir da conversa entre os pesquisadores e os profissionais, que resultou na segunda versão da UserX Story, conforme Figura 4.8.

<b><i>UserX Story (versão 2.0)</i></b>
<b>Como &lt;persona&gt; Eu quero / preciso &lt;objetivo&gt;, para &lt;interação&gt;, através/quando [ &lt; tarefa &gt; / &lt; contexto &gt; ]. Eu avalio que meu objetivo foi alcançado &lt;feedback&gt;.</b>
<b>Critérios de aceitação:</b>  Teste <ação> com/atraves <conjunto de condições> para satisfazer <Heurística(s) de Nielsen> de <u>ação</u> , e <Heurística(s) de Nielsen > de <u>feedback</u> .

Figura 4.8: UserX Story - versão refinada

As UserX Stories (2ª versão) expandem as US tradicionais a partir dos resultados recolhidos nas pesquisas com os usuários (inclusive testes de usabilidade). As UserX Stories são contadas a partir da perspectiva da persona, que precisa de uma condição particular de interação. Tal condição pode atender várias personas. As histórias descrevem um processo interativo em que a persona tem uma meta a atingir, por isso ele/ela atua sobre a interface (interação), para executar tarefas (passos/recursos) para efetuar a ação em um determinado contexto (padrão de uso). A persona irá avaliar se o objetivo foi alcançado interpretando o *feedback* do sistema. Com o objetivo de verificar se as histórias foram desenvolvidos de acordo com as necessidades de interação do usuário, os critérios de aceitação deverão descrever a ação, o conjunto de condições, e as heurísticas da Nielsen (ação/ *feedback*) que serão satisfeitas uma vez que o objetivo seja alcançado com sucesso.

Antes de implementar o artefato para avaliá-lo na prática, os pesquisadores organizaram um *workshop* intitulado “Histórias de Interação” para treinar seis POs na escrita de UserX Stories.

Todos os participantes tinham mais de dez anos de experiência na área de TI (Tabela 4.3). No entanto, cinco dos seis participantes tinham pouca experiência com metodologia ágil (menos de

um ano), e conseqüentemente pouca familiaridade com User Stories. Nenhum dos participantes havia participado do *workshop* anterior sobre “Heurísticas de Usabilidade para sistemas ERP”, por isso conceitos sobre personas e heurísticas de Nielsen tiveram que ser apresentados.

Tabela 4.3: Perfil dos participantes

Participantes	Experiência na área de TI	Tempo de empresa	Experiência com metodologia ágil
P1	14 anos	13 anos	6 meses
P2	11 anos	2 semanas	2 semanas
P3	15 anos	8 anos	6 meses
P4	15 anos	9 anos	9 meses
P5	27 anos	27 anos	3 meses
P6	17 anos	3 anos	3 anos

O *workshop* foi dividido em três partes: (i) explicação do conceito de personas, heurísticas de Nielsen e User Stories; (ii) apresentação dos artefatos Lean Personas e UserX Stories; e (iii) um exercício para escrever histórias a partir do modelo proposto, incluindo critérios de aceitação. Como material complementar, os pesquisadores forneceram um conjunto de dados que haviam sido coletados com usuários finais por um grupo de colaboradores que haviam participado de um *workshop* anterior sobre *User Research*. Além disso, algumas personas foram incluídas neste material. No final do *workshop*, os pesquisadores combinaram com os POs os próximos passos para implementação das UserX Stories em projetos reais selecionados por eles.

#### 4.4.2.3 Implementando, observando as melhorias e avaliando artefatos

Nesta etapa, os POs tiveram o período de um mês para implementar as UserX Stories em um de seus projetos (3a). Após este período, os pesquisadores realizaram entrevistas individuais com cada um dos POs para recolher as suas experiências com as UserX Stories (3b).

A maioria dos POs aprovaram o uso das UserX Stories. Alguns haviam compartilhado as UserX Stories com as equipes Scrum, que reagiram positivamente. No entanto, dois POs não haviam implementado as UserX Stories em seus projetos, uma vez que eles estavam trabalhando em pequenas mudanças que foram relacionados exclusivamente com regras de negócio e requisitos legais, e tais mudanças não teriam nenhum impacto sobre a interação do usuário. Um dos POs comentou sobre a dificuldade que teve para definir o quanto os critérios de aceitação deveriam ser detalhados. Tal questão foi anotada para pesquisas futuras. A Figura 4.9 apresenta um exemplo de UserX Story escrita por um dos POs referente a um projeto para implementar novas funcionalidades no sub-modulo do módulo de Escrituração Fiscal.

Os pesquisadores reconheceram que mais testes seriam necessários para avaliar o ciclo de UserX Stories escritas a partir de itens relatados no Protocolo de Comunicação. A princípio, acredita-se que os itens relatados no Protocolo devem ser discutidos entre UX *designer* e equipe Scrum (em reuniões diárias ou durante planejamento das Sprints). Caso a equipe Scrum concorde com o item recomendado, o item deverá ser escrito na forma de UserX Story (com a participação do *designer* de UX). Caso contrário, a recomendação é descartada (por exemplo, por limitações técnicas impedindo a implementação da mesma).



<i>UserX Story (submódulo de Escrituração Fiscal)</i>
<p>Eu &lt;Leo Guimarães&gt; <b>preciso</b> &lt;emitir relatórios e balanços financeiros, filtrando por agentes&gt;, <b>para isto</b> &lt;o sistema permite escolher o agente que eu quero filtrar&gt;, <b>através de / quando</b> [&lt;pela emissão do relatório&gt; / &lt;independentemente da organização à qual estou posicionado no sistema, sendo filial ou consolidadora&gt;].</p> <p><b>Avalio que o meu objetivo foi alcançado quando</b> &lt;o relatório listar apenas os lançamentos feitos para o agente selecionado&gt;</p>
<p><b>Critérios de aceitação:</b></p>
<p><b>Teste</b> &lt; se o sistema irá validar se que o código agente pode ser utilizado para a organização selecionada&gt; <b>com/através</b> &lt;filtragem por código de agente&gt; <b>para satisfazer</b> &lt;H5&gt; de ação, e &lt;H9&gt; de feedback.</p> <p><b>Teste</b> &lt; se o sistema irá exibir nome do agente ao lado do código escolhido&gt; <b>com/através</b> &lt;escolher um agente, ou por código do agente ou à procura&gt; <b>para satisfazer</b> &lt;H1&gt; de ação, e &lt;H6&gt; de feedback.</p>

Figura 4.9: Exemplo de UserX Story desenvolvida por um PO

### 4.4.3 Resultados

As atividades de pesquisa-ação desenvolvida junto com colaboradores da indústria possibilitou a definição e validação de três artefatos: Lean Personas, Protocolo de Comunicação e as UserX Stories. As Lean Personas ajudaram as equipes a manterem o foco nas necessidades dos usuários durante a avaliação do produto direcionando os testes de usabilidade. O Protocolo ajudou a melhorar a comunicação entre *designers* e o time ágil, pois sua estrutura faz uso de personas e das heurísticas de Nielsen como vocabulário comum para tratar de questões relacionadas com usabilidade. As heurísticas de Nielsen, em particular, ajudam a fixar muitos conceitos sobre usabilidade sem muito esforço, e demonstraram eficiência para confirmar problemas de usabilidade durante a análise dos resultados dos testes com usuários.

As UserX Stories, que também carregam conceitos de personas e as heurísticas de Nielsen, demonstram ser bem aderentes às práticas ágeis por ter evoluído de um artefato muito utilizado no DAS para documentar requisitos. A estrutura proposta para as UserX Stories permite incorporar requisitos de usabilidade a partir da perspectiva da persona destacando vários elementos do cenário de interação, tais como objetivo, ação, tarefas, contexto e *feedback* do sistema. Adicionalmente, as UserX Stories fornecem um formato para os critérios de aceitação que relaciona os itens de teste às diretrizes de usabilidade.

Observou-se que os artefatos produzidos e implementados na indústria atenderam de forma satisfatória às necessidades dos *designers* e desenvolvedores de projetos ágeis. Diante dos resultados que atestam utilidade prática, tais artefatos serão recomendados para dar suporte à abordagem de integração ScrumUX.

## 4.5 ScrumUX

Considerando as estratégias identificadas no estudo bibliográfico para integrar métodos ágeis e o DCU e os resultados obtidos junto com os profissionais da indústria de software, na elaboração dos artefatos de apoio, este trabalho apresenta uma abordagem de integração, denominada ScrumUX, com o objetivo de integrar o DxI com o processo Scrum.

O DxI e o Scrum não visam resolver precisamente os mesmos problemas no desenvolvimento de software. Valores e práticas ágeis estão mais preocupados com gerenciamento de projetos e questões organizacionais, bem como fazer a codificação mais eficiente. O DxI, por outro lado, tem foco nos métodos de concepção e *design* centrado no usuário. Como já mencionado anteriormente, as diferentes perspectivas em relação à usabilidade pode dificultar a integração das duas áreas.

Blomkvist (2005) delineou um modelo abstrato para preencher as lacunas entre DCU e DAS, descrito a partir de três perspectivas diferentes de integração: (1) os métodos de DCU são integrados à estrutura ágil na qual os valores ágeis não são afetados; (2), as técnicas ágeis são integradas à estrutura do DCU na qual são os valores do *design* centrado no usuário que serão preservados; e (3) a polinização cruzada equilibrada, combinando valores, métodos, técnicas das duas disciplinas se complementam ou são mais bem adaptados. De acordo com Blomkvist (2005), a integração equilibrada é a que tem mais chances de sucesso, e pode ocorrer em diferentes níveis. Em escala menor, um exemplo de polinização cruzada equilibrada seria a adoção de testes de usabilidade para complementar os testes de aceitação. Outro exemplo, seria a adoção de valores ágeis, tais como a comunicação face-a-face, para melhorar a comunicação entre usuários e desenvolvedores, bem como entre os desenvolvedores e os *designers* de usabilidade. Em escala maior, a integração pode resultar em novos métodos híbridos, ou até mesmo processos híbridos completos, que são ao mesmo tempo ágil e centrado no usuário. Alguns exemplos de métodos processos híbridos foram apresentados no capítulo de trabalhos relacionados.

Seguindo a premissa de polinização cruzada equilibrada do modelo conceitual proposto por Blomkvist (2005), a abordagem ScrumUX se baseia na adoção de valores, métodos e técnicas tanto do DxI como do Scrum. Para alcançar uma integração mais equilibrada entre DxI e o Scrum, a abordagem ScrumUX preserva valores que são fundamentais à visão integrada:

- Entrega adiantada e contínua de software de valor, útil e com boa usabilidade;
- Priorizar as funcionalidades do sistema, incluindo os aspectos de usabilidade;
- O *design* deve ter um nível de detalhamento suficiente para caber dentro de iterações curtas, permitindo ser incrementado e melhorado a cada iteração;
- As equipes de *designers* e desenvolvedores devem estar afinados, comunicando-se com objetividade e clareza, seja através de comunicação face a face, seja através de comunicação mediada por artefatos de apoio;
- A avaliação da usabilidade é essencial em diferentes fases de desenvolvimento, de maneira que testes mais simples com um número menor de usuários sejam executados com mais

frequência, e que testes mais completos sejam executados com um conjunto maior de funcionalidades;

- Adoção de artefatos mais simples, tais como protótipos de papel (*sketches*), para permitir explorar e avaliar rapidamente diferentes soluções de *design*;
- Adaptação de artefatos, tais como as histórias de usuário, para descrever os requisitos de usuários e os cenários de interação;
- Sempre que possível, usuários reais (não apenas clientes) devem estar ativamente envolvidos nas fases do desenvolvimento. Os especialistas de usabilidade devem advogar em defesa dos usuários e manter o foco do desenvolvimento em suas necessidades e interesses. É interessante que, membros da equipe de desenvolvimento conheçam os detalhes do contexto dos usuários.

Com relação à organização do trabalho, a abordagem ScrumUX adota uma configuração híbrida (FOX; SILLITO; MAURER, 2008) que permite a participação dos desenvolvedores em atividades de DxI, uma vez que muitos projetos de pequeno porte, como apontado por Liikkanen et al. (2014), desenvolvidos da cultura de olhar para o usuário com foco na sua experiência, podem não dispor de orçamento para contratar profissionais de UX para integrá-los à equipe. Deste modo, a presente proposta pretende beneficiar, inclusive, pequenas equipes com de interesse produzir software de qualidade com foco nas necessidades dos usuários, mas que não são especialistas de UX. Contudo, a abordagem ScrumUX define quatro papéis principais: Product Owner, UX Designer, Scrum Master e a Equipe de desenvolvedores. A Tabela 4.4 descreve os principais papéis da abordagem conforme etapas do Scrum.

A Figura 4.10 apresenta a abordagem ScrumUX, onde na parte superior está representado o processo Scrum, e na parte inferior, as etapas e atividades do DxI. As atividades do DxI estão distribuídas em três etapas: Pesquisa (1), *Design* (2) e Verificação (3). As três etapas são iterativas, contudo a abordagem ScrumUX não prescreve um sequência fixa de passos. As etapas do DxI podem acontecer em qualquer fase do Scrum, e por vezes, simultaneamente às suas atividades. Ou seja, as atividades do DxI podem acontecer em qualquer momento do Scrum, sem a necessidade de se aplicar, obrigatoriamente, todos os passos. Durante a fase de Visão, por exemplo, pode ser necessário realizar atividades de pesquisa e de prototipação (etapas 1 e 2), e programar as atividades de avaliação (etapa 3) para outro momento. Ou ainda, em determinada situação pode não ser necessário realizar atividades de pesquisa, por exemplo, mas protótipos precisam ser refinados e testados com usuários (etapas 2 e 3). As setas no ciclo do DxI representam o fluxo das atividades descritas em cada etapa, permitindo várias possibilidades de combinação para atender diferentes necessidades de projeto.

Em cada etapa do DxI são produzidos artefatos de *design* de dois tipos: (1) artefatos conceituais e (2) artefatos de integração. Os artefatos conceituais são aqueles que fornecem informações para realização de atividades em outras etapas, tais como: personas, cenários de interação, *sketches* e protótipos. Os artefatos de integração são artefatos de *design* compartilhados por *designers* e pela a equipe de desenvolvimento, que apoiam a coordenação, colaboração e comunicação entre os times. Os artefatos de integração são os artefatos que foram elaborados

Tabela 4.4: Principais papéis da abordagem ScrumUX

	Product Owner	UX Designer	Scrum Master	Equipe de desenvolvedores
Visão	Principal responsável por definir a visão do produto, colhendo informações junto à clientes, usuários, time Scrum e demais <i>stakeholders</i> .	Participa de atividades para definir a visão do produto, realizando pesquisas de domínio e usuários.	Não tem participação ativa, mas fornece feedback para a visão do produto.	Não têm participação ativa, mas fornecem <i>feedback</i> para a visão do produto.
Product Backlog	Principal responsável por criar a lista priorizada de necessidades para atender os objetivos do projeto, definidos a partir da visão do produto.	Fornecer informações sobre requisitos de usabilidade e interação do usuário.	Pode auxiliar o PO na elaboração da lista de necessidades.	Não têm participação ativa na definição do Product Backlog, mas fornecem informações, por exemplo, sobre limitações técnicas.
Sprint Backlog	Participa na primeira parte reunião de planejamento da Sprint, compartilhando a meta da Sprint, apresentando o Product Backlog priorizado, e respondendo a quaisquer perguntas que a equipe possa ter em relação aos itens do Product Backlog.	Fornecer informações sobre requisitos de usabilidade e interação do usuário, por meio dos artefatos de <i>design</i> (personas, <i>sketches</i> , protótipos, resultados de avaliações de usabilidade); auxilia na escrita das histórias de interação.	Atua como facilitador da reunião para ajudar a garantir um bom resultado, observando a atividade de planejamento e explorando questões levantadas durante a reunião.	Na primeira parte da reunião de planejamento, estimam e determinam os itens que serão trabalhados na Sprint. Na segunda parte da reunião, colhem mais detalhes sobre os Itens selecionados decompondo-os em tarefas para a Sprint Backlog.
Execução da Sprint	Fica à disposição da equipe de desenvolvimento para esclarecer dúvidas do projeto.	Fica à disposição da equipe de desenvolvimento para esclarecer dúvidas sobre o <i>design</i> e os requisitos de usuários; em paralelo executam atividades de <i>design</i> de interação (pesquisa, ideação, prototipação e avaliação) para as próximas Sprints.	Facilita o trabalho do time, removendo qualquer impedimento.	Auto-organizáveis, determinam a melhor maneira de cumprir a meta da Sprint.
Revisão da Sprint	Após a demonstração da equipe de desenvolvimento, avalia se a meta da Sprint foi alcançada e faz anotações que poderão se transformar em novos itens para o Product Backlog.	Convidado à participar da na reunião de Revisão da Sprint, verifica se necessidades dos usuários foram atendidas.	Atua como facilitador da reunião.	A equipe apresenta para o PO e demais convidados ( <i>designer</i> de UX e <i>stakeholders</i> ) o que a equipe conseguiu fazer durante a Sprint, demonstrando somente o que satisfaz a definição de feito ( <i>Done</i> ), que pode não ser o produto completo, mas um atributo concluído do produto e que pode ser entregue para o cliente para a obtenção de <i>feedback</i> .
Retrospectiva da Sprint	Não participa ativamente da reunião.	Não participa ativamente da reunião.	Reúne-se com a equipe para refletir sobre as lições aprendidas: o que foi bom na última Sprint? O que deve ser melhorado?	Refletem sobre o desempenho do trabalho com o objetivo de aprimorar processo de desenvolvimento para as próximas Sprints.

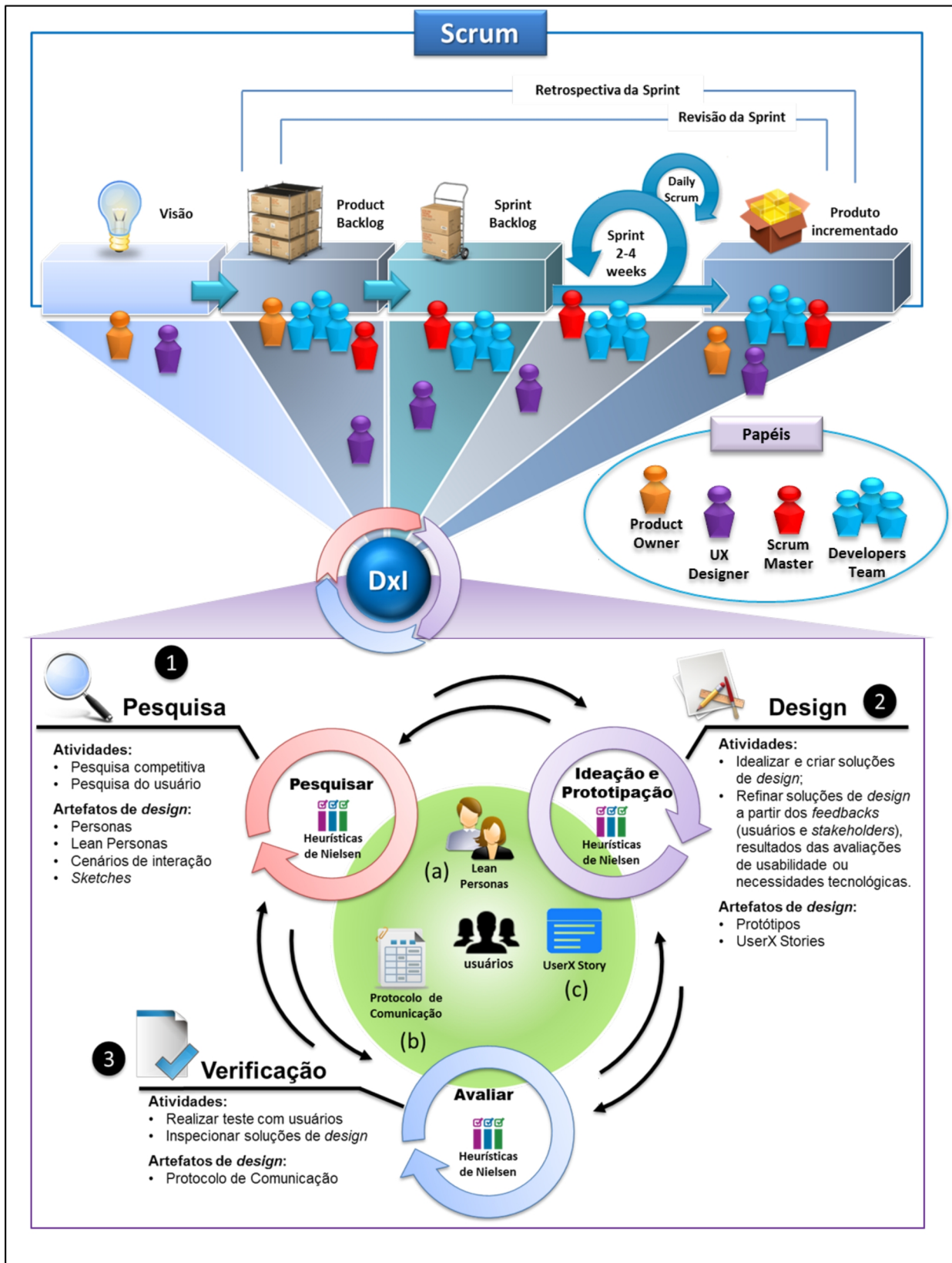


Figura 4.10: Abordagem ScrumUX

e validados junto à profissionais da indústria para o contexto de desenvolvimento ágil: Lean Persona (a), Protocolo de Comunicação (b) e UserX Story (c). Os artefatos de integração aparecem no centro do DxD, pois carregam todas as informações que definem os usuários descrevendo suas necessidades de interação. A participação do usuário em todas as etapas do DxD não é obrigatória, contudo suas necessidades devem nortear as decisões em cada etapa de *design* e desenvolvimento. A heurísticas de Nielsen é um artefato chave que permeia todas as etapas do DxD, tanto para direcionar as soluções de *designer* como para dar suporte às avaliações de usabilidade. A Tabela 4.5 apresenta a descrição de todos os artefatos recomendados na abordagem ScrumUX.

Tabela 4.5: Artefatos de *design* recomendados na abordagem ScrumUX

Artefatos	Descrição
Personas	Técnica utilizada para definir o público alvo através da criação de personagens fictícios; arquétipos hipotéticos representando um grupo de usuários reais.
Lean Personas*	Modelo recomendado na abordagem ScrumUX para descrever as personas com dados que refletem o nível de conhecimento, objetivos e padrões de utilização de usuários típicos.
Cenários de Interação	Narrativa textual detalhando as ações dos usuários e o <i>feedback</i> do sistema, necessários para alcançar um ou mais objetivos, considerando contexto de uso, funções e tarefas dos usuários.
<i>Sketches</i>	Representação de baixa fidelidade de um <i>design</i> (rascunhos de tela) usado para capturar as partes mais importantes da aplicação que será desenvolvida (esqueleto do produto).
Protótipos	Representação de média ou de alta fidelidade do produto final, que possibilita uma simulação da interação entre o usuário e a interface para testar as principais interações de forma similar a experiência que o usuário terá no produto final.
UserX Story*	Modelo recomendado na abordagem ScrumUX adaptando a gramática de User Stories para descrever requisitos de interação do usuário.
Protocolo de Comunicação*	Modelo para comunicar soluções e/ou recomendações de <i>design</i> que são relacionadas com personas e heurísticas de Nielsen.
Heurísticas de Nielsen	<i>Guidelines</i> de usabilidade para nortear atividades de avaliação e <i>design</i> .

\* artefatos de integração

Na etapa de Pesquisa (1), as atividades de pesquisa do usuário e pesquisa competitiva são as atividades básicas do profissional que irá atuar no papel de UX Designer. Para pesquisar sobre as necessidades e desejos dos usuários, o UX Designer pode escolher as técnicas de coleta mais convenientes, tais como etnografia, *survey*, entrevista, *card sorting*, grupo de foco, etc. Quanto à pesquisa competitiva, recomenda-se a inspeção de usabilidade em produtos concorrentes utilizando as heurísticas de Nielsen. A análise dos produtos concorrentes irá fornecer

uma visão das boas práticas e tendências do mercado, e ainda, revelar pontos negativos relacionados à usabilidade que poderão ser resolvidos ou minimizados. Após análise dos dados coletados na atividade de pesquisa, o UX Designer elabora as personas, os cenários de interação e sketches que serão artefatos de entrada para apoiar as atividades na próxima etapa (etapa 2 ou 3). As personas, em particular, deverão ser escritas de forma sucinta utilizando o modelo de Lean Personas (a) (Figura 4.3). Hipóteses de personas podem ser o ponto de partida para determinar os potenciais tipos de usuários no domínio da aplicação, desde que sejam validadas, por exemplo, observando-se usuários reais durante os testes de usabilidade (etapa 3).

Na etapa de *Design* (2), as principais atividades do UX Designer será idealizar e criar soluções de design e/ou refinar protótipos a partir de *feedbacks* de usuários e *stakeholders*, resultados das avaliações de usabilidade (etapa 3) ou necessidades de adequação tecnológica (etapa 1). Os artefatos de entrada fornecem as informações necessárias para orientar as decisões de *design*. Os artefatos gerados nessa etapa são os protótipos (de média ou alta fidelidade) e as UserX Stories (c). Os protótipos são utilizados para apoiar a atividade de avaliação (etapa 3), bem como as atividades dos desenvolvedores. As UserX Stories devem descrever os requisitos funcionais de interação e os critérios de aceitação relacionados; sendo que os requisitos funcionais de interação são extraídos dos cenários de interação (etapa 1) e os critérios de aceitação extraídos dos resultados observados durante os testes com usuários e/ou em inspeções de usabilidade (etapa 3). As UserX Stories incorporam os conceitos de personas e heurísticas de Nielsen, conforme modelo apresentado na Figura 4.8.

Na etapa de Verificação (3), a principal atividade o UX Designer é avaliar a usabilidade das soluções de design em várias fases do projeto através de testes com usuário e/ou inspeções de usabilidade. As heurísticas de Nielsen devem nortear a atividade de inspeção e apoiar a análise dos testes com usuários. A avaliação heurística é considerada uma técnica menos onerosa, e por esta razão, pode ser adotada como um método preditivo e realizada com mais frequência.

Os testes com usuários, por outro lado, consomem mais recursos e exigem um planejamento antecipado. As etapas do processo para realização dos testes são: (i) selecionar os usuários representativos; (ii) determinar as questões específicas de investigação; (iii) selecionar técnicas de coleta e configurar ambiente de teste; (iv) preparar material de apoio (termo de consentimento, questionário de satisfação, roteiro da entrevista, lista das tarefas do usuário, etc.); (v) conduzir as sessões de teste; e (vi) discutir e analisar dados e observações (LAZAR; FENG; HOCHHEISER, 2010).

Os membros do time Scrum e outros *stakeholders* podem ser convidados a participar como observadores das sessões de teste. Os usuários representativos (no mínimo cinco) devem ser recrutados com base nas personas. Eventualmente, questionários de usabilidade tal como o SUS, podem ser aplicados para medir o nível de satisfação dos usuários em relação à usabilidade da aplicação. O monitor de teste grava as sessões, faz as anotações, e ao final, lista todos os problemas de usabilidade identificados, agrupando os problemas apontados por mais de uma persona e eliminando os falsos positivos (questões que não consideradas um problema real). Os problemas identificados podem ser analisados imediatamente às sessões de teste, como sugere Kjeldskov, Skov e Stage (2004). O processo de análise consiste em (i) analisar cada questão identificada, propor soluções e/ou recomendações; (ii) identificar as personas potencialmente afetadas pelo

problema; e, (iii) relacionar cada questão com as heurísticas de Nielsen correspondente. Essas informações são formalizadas através do Protocolo de Comunicação (b), que é o artefato recomendado para relatar soluções e/ou recomendações de *design* à equipe de desenvolvimento (Figura 4.6).

As questões relacionadas no Protocolo de Comunicação (b) podem ser discutidas com a equipe de desenvolvimento durante as reuniões de planejamento ou em reuniões diárias, conforme prioridades. O time deve avaliar os itens relatados no Protocolo observando seus efeitos, as personas afetadas e as heurísticas da Nielsen atendidas, antes de escrever as UserX Stories (c). As novas UserX Stories são adicionadas ao Product Backlog, e posteriormente, decompostas em tarefas que serão selecionadas na Sprint Backlog.

Durante as Sprints, o UX Designer ficará à disposição da equipe de desenvolvimento para esclarecer quaisquer dúvidas dos desenvolvedores relacionadas ao projeto de *design*, assegurando que as necessidades dos usuários sejam atendidas. No final da Sprint, o produto incrementado é submetido a uma avaliação de usabilidade mais completa (etapa 3).

## 4.6 Análise comparativa

A abordagem ScrumUX se baseia em muitos dos conceitos e técnicas mencionados em trabalhos correlatos. Contudo, apresenta suas próprias contribuições no que se refere aos artefatos que dão suporte à abordagem. Os artefatos elaborados a partir da experiência com profissionais da indústria são o núcleo da abordagem e o principal diferencial, em relação às outras propostas de integração, abrangendo processos e *frameworks*. Outro destaque da abordagem ScrumUX são as heurísticas de Nielsen que, adotadas como diretrizes de usabilidade, permeiam todo o processo de *design* e desenvolvimento.

Na categoria de processos, a maior parte dos processos foram propostos com base no processo XP, sendo que o único processo com base no Scrum foi proposto por Paelke e Nebe (2008) para um domínio muito específico, cujas características prescreve um longo período inicial para investigação e análise dos requisitos de usuários e contexto de uso. O XP é focado mais nas práticas de desenvolvimento de software, enquanto o Scrum é mais abrangente, pois tem foco no gerenciamento do projeto, deixando a equipe de desenvolvedores livre para escolher suas práticas de desenvolvimento. Com relação à avaliação de usabilidade, pôde-se observar que todos os processos propostos indicam testes com usuários, contudo somente o processo proposto por Lee e McCrickard (2007) recomenda a avaliação heurística para complementar a etapa de testes do XP. Nenhum dos processos propostos mencionam personas para descrever grupos usuário típicos; e as histórias de usuário são mencionadas somente nos processos de Obendorf e Finck (2008) e de Lee e McCrickard (2007). Obendorf e Finck (2008) propõe incluir *storyboards* em histórias de usuários e misturar protótipos de papel com cenários, entretanto no estudo de caso com profissionais os cenários foram usados para agrupar às histórias de usuário fornecendo uma visão contextual.

Na categoria de *framework*, somente duas propostas são dirigidas ao Scrum. Iqbal e Khan (2014) menciona as atividades básicas da engenharia de usabilidade e do *design* centrado no usuário integradas ao Scrum, contudo não recomendam práticas ou artefatos de *design*. O único



artefato mencionado são as histórias de usuários utilizadas para incluir as tarefas de usuários. Um ponto interessante da proposta de Anwar et al. (2014) é que os autores definem a um tempo máximo para Sprint Zero de sete semanas para desenvolver protótipos, avaliar *design* e recolher *feedback* com usuários; e caso necessário, fazem o *feedback* no decorrer das próximas Sprints para incluir melhorias necessárias identificadas no teste de usabilidade. Contudo, deixam claro que sua proposta é indicada para pequenos projetos. Os autores indicam um leque de técnicas de *User Research* e avaliação dentre as quais personas e a avaliação heurística são mencionadas. Tais técnicas são mencionadas também na proposta de Wolkerstorfer et al. (2008), que propõem inclusive que as personas sejam fixadas junto às histórias de usuários. Nenhuma outra proposta, porém, recomenda artefatos mais adaptados incorporando conceitos das duas áreas (*design* e desenvolvimento).

## 4.7 Considerações finais

A primeira parte deste capítulo descreve e apresenta os resultados da metodologia de pesquisa-ação SoftCoDeR aplicada para construir e validar artefatos para dar suporte à abordagem de integração. As atividades junto aos profissionais de uma indústria de desenvolvimento de software na área de Sistemas ERP são descritas em dois ciclos da abordagem SoftCoDeR. Os três artefatos desenvolvidos foram avaliados em projetos reais. Após a experiência na indústria, a abordagem ScrumUX é proposta incorporando os três artefatos produzidos e concebida a partir de um modelo conceitual que combina valores do Scrum com valores do DxI. Na sequência, apresenta-se as etapas da abordagem descrevendo o fluxo das atividades, os papéis e os artefatos produzidos em cada etapa. O capítulo termina com uma análise da abordagem em relação aos trabalhos correlatos.

# Validação da abordagem ScrumUX: um estudo experimental

## 5.1 Considerações iniciais

Esta pesquisa é apoiada fortemente na experimentação. Estudos experimentais permitem avaliar diferentes tecnologias utilizadas na Engenharia de Software, incluindo a eficácia dos métodos, técnicas e ferramentas em diferentes ambientes de desenvolvimento (TRAVASSOS; GUROV; AMARAL, 2002).

Este capítulo apresenta o estudo experimental executado com o objetivo de validar a abordagem ScrumUX, fornecendo detalhes das atividades realizadas e a sequência em que foram feitas, a fim de enfatizar o rigor e a validade da pesquisa.

## 5.2 Experimentação em Engenharia de Software

A Engenharia de Software Experimental busca aprimorar a ES aplicando a experimentação na construção de novos métodos e técnicas para apoio ao desenvolvimento de software. A experimentação permite avaliar a atividade humana de modo sistemático, disciplinado, computável e controlado (TRAVASSOS; GUROV; AMARAL, 2002). Os experimentos são realizados quando o pesquisador quer ter controle sobre a situação, com manipulação direta, precisa e sistemática do comportamento do fenômeno a ser estudado, e ainda sobre as atividades e distribuição das pessoas nos grupos do experimento (SHULL; CARVER; TRAVASSOS, 2001).

O atributo comum em todos os experimentos é o controle do tratamento, onde uma ou mais variáveis independentes são manipuladas para observar seus efeitos sobre uma ou mais variáveis dependentes. As variáveis independentes são as variáveis manipuladas e controladas em um experimento, enquanto que as variáveis dependentes são aqueles que se deseja estudar para ver o efeito das alterações nas variáveis independentes. O efeito observado nos resultados do experimento é medido e analisado para validar ou refutar as hipóteses previamente formuladas (SHADISH; COOK; CAMPBELL, 2002).

### 5.3 Escopo do estudo experimental

O presente estudo experimental foi desenvolvido tendo em vista a segunda questão de pesquisa (Q2): A abordagem ScrumUX, integrando atividades e artefatos de DxD ao processo Scrum, pode apoiar uma equipe ágil na implementação dos mecanismos de usabilidade durante as atividades de desenvolvimento de software?

Seguindo o paradigma orientado a meta para a medição de produtos e processos de software (GQM - *Goal Question Metric*) proposto por Basili e Rombach (1988), o objetivo do experimento foi:

- **Analisar** o uso da abordagem ScrumUX;
- **Com o propósito** de avaliação;
- **Com respeito** à usabilidade das aplicações desenvolvidas;
- **Do ponto de vista** de desenvolvedores e usuários de software;
- **No contexto** de projeto de desenvolvimento de software para o domínio de sistemas ERP com alunos de graduação em Ciência da Computação.

As aplicações desenvolvidas durante o experimento pelos alunos ficaram restritas a um único cenário: desenvolvimento de um submódulo ERP de vendas específico para o mercado de distribuição de produtos perecíveis. A próxima Seção descreve a métrica estabelecida para verificar a usabilidade de cada uma das aplicações que seriam desenvolvidas durante o experimento.

### 5.4 Métrica de Usabilidade para Sistemas ERP

Dentro do domínio de sistemas ERP, muito esforços têm sido realizados para tornar mais eficientes os métodos para avaliar a usabilidade desses sistemas. Unindo-se a estes esforços, Choma et al. (2015) propuseram as heurísticas por perspectiva ERP (PERP) motivados pela técnica de inspeção por perspectiva de usabilidade proposta por Zhang, Basili e Shneiderman (1999).

Choma et al. (2015) realizaram estudos preliminares para (i) investigar problemas de usabilidade em diferentes sistemas de ERP do mercado. Considerando os resultados da análise dos problemas de usabilidade identificados, os autores elaboraram (ii) uma pesquisa com 22 questões para avaliar o grau de importância atribuído por usuários de sistemas ERP para cada aspecto de usabilidade. A partir dos resultados preliminares (i e ii) os problemas de usabilidade identificados foram agrupados pelas perspectivas de apresentação e suporte à tarefas, e posteriormente foram mapeadas com as heurísticas de Nielsen, analisando-se o impacto das perspectivas sobre as heurísticas. A Tabela 5.1 apresenta o resultado do mapeamento que deu origem às Heurísticas por perspectiva ERP (HPERP).

Na perspectiva de apresentação considera-se como a informação está sendo apresentada ao usuário analisando-se características relacionadas com a programação visual e o layout de interface. As três questões que norteiam a perspectiva de apresentação são: “Estou vendo?”,

Tabela 5.1: Heurísticas por perspectivas de sistemas ERP

Heurísticas de Nielsen		HPERP	
		A	S
H1	Visibilidade do estado do sistema	A1	
H2	Concordância entre o sistema e o mundo real	A2	
H3	Controle e liberdade ao usuário		S3
H4	Consistência e padrões	A4	
H5	Prevenção de Erros	A5	S5
H6	Reconhecer ao invés de lembrar	A6	S6
H7	Flexibilidade e eficiência de uso		S7
H8	Projeto minimalista e estético	A8	
H9	Reconhecimento, diagnóstico e recuperação de erros	A9	S9
H10	Ajuda e Documentação		S10

Legenda:

HPERP - Heurísticas por perspectiva ERP

A - Apresentação | S - Suporte à tarefas

“Estou entendendo?”, “A mensagem está clara?”. Quanto à perspectiva de suporte às tarefas, considera-se o espaço navegacional e a arquitetura da informação para avaliar como os usuários são conduzidos à interação para realizar suas tarefas. A questão que norteia a perspectiva de suporte à tarefas é “Consigo concluir a tarefa sem obstáculos?”.

Com o objetivo de estabelecer uma métrica para avaliar a usabilidade de cada uma das aplicações produzidas durante o estudo experimental, analisou-se os resultados da pesquisa realizada por Choma et al. (2015) junto à usuários de sistemas ERP (ii). A pesquisa havia sido respondida por 37 usuários de sistemas de ERP que atribuíram o grau de importância para cada questão utilizando a escala de Likert (muito importante, significativamente importante, importante, pouco importante ou nada importante). Na sumarização dos resultados, calculou-se o grau de importância de cada questão (aspecto de usabilidade) atribuindo-se *Scores* (pontuação) aos itens da escala *Likert*, conforme Tabela 5.2.

Tabela 5.2: *Scores* para a escala *Likert*

Grau de importância	Score(S)
Muito importante	100
Significativamente importante	75
Importante	50
Pouco importante	25
Nada importante	0

Os valores médios atribuídos pelos usuários (*Scores*) para cada um dos aspectos de usabilidade foi calculado. As Tabelas 5.3 e 5.4 apresentam os aspectos de usabilidade e seus respectivos *Scores*. Adicionalmente, cada aspecto foi mapeado com as HPERP de apresentação e de suporte à tarefas.

Com o intuito de incluir a opinião dos usuários de sistemas ERP, o *Score* de cada aspecto

Tabela 5.3: Grau de importância por Perspectiva de Apresentação.

HN	HPERP	Questão de usabilidade	Score
H1	A1.1	Identificar visualmente a localização no sistema para saber onde está e para onde pode ir.	76,35
	A1.2	Mensagens informativas sobre o que está acontecendo no sistema.	73,65
H2	A2.1	O conteúdo da tela distribuído e posicionado seguindo uma ordem natural e lógica.	85,81
	A2.2	Uso de linguagem familiar que pode ser facilmente interpretada.	75,00
H4	A4.1	Padronização das mensagens, formatos, símbolos e cores do sistema.	75,00
H5	A5.1	Indicação do formato correto de entrada de dados (data, CEP, telefone, etc.).	87,84
	A5.2	Destaque nos campos de formulário que são de preenchimento obrigatório.	86,49
	A5.3	Alerta sobre formatos incorretos ou dados inconsistentes.	90,54
H6	A6.1	Palavras e termos adequados que podem ser facilmente lembrados.	66,89
H7	A7.1	Elementos de interface dispostos de forma a minimizar o esforço de ações físicas e nas buscas visuais.	70,95
H8	A8.1	Tela não contendo informação irrelevante ou raramente necessária.	61,49
H9	A9.1	Mensagens de erro visíveis, simples e de fácil entendimento.	91,22

Tabela 5.4: Grau de importância por Perspectiva de Suporte à Tarefas

HN	HPERP	Questão de usabilidade	Score
H3	S3.1	Controle de localização para voltar ao ponto de partida ou sair de um estado inesperado.	67,57
H5	S5.1	Mensagens que previnem a ocorrência de problemas no caso de ações equivocadas.	72,30
	S5.2	Os botões de ação conforme identificados definem claramente o estado que será atingido após seu acionamento.	77,03
	S5.3	O sistema fornece uma sequência de operações para completar tarefas.	73,65
H7	S7.1	Informação organizada em ordem alfabética ou lógica por relevância de uso.	62,84
	S7.2	Automatização de tarefas rotineiras e redundantes.	72,97
	S7.3	Filtros de busca adequados ao número de itens e informações.	71,62
H9	S9.1	Apoio para fazer e desfazer ações.	73,65
	S9.2	Mensagens que auxiliam na recuperação de erros e mostra como acessar soluções alternativas.	77,70
H10	S10.1	Apoio às tarefas complexas com instruções visíveis e acessíveis.	80,41

de usabilidade ( $S_{UA}$ ) é um dos fatores considerado para cálculo do *Índice de Usabilidade* para sistemas ERP ( $UI_{ERP}$  - ERP Usability Index), proposto como métrica para avaliar a usabilidade das aplicações que serão desenvolvidas durante o experimento.

O outro fator a ser considerado para cálculo do  $UI_{ERP}$  é a gravidade das violações identificadas pelos inspetores de usabilidade em cada uma das aplicações durante a avaliação heurística.

Tradicionalmente, as violações de usabilidade identificadas em avaliações heurísticas são classificadas conforme nível de gravidade, utilizando-se a seguinte escala:

- Cosmética (nível 1): problema estético que não tem urgência de ser corrigido, a menos que haja tempo e recurso disponível.
- Pequena (nível 2): problema com baixa prioridade na correção.
- Grande (nível 3): problema com alta prioridade de correção.
- Catastrófica (nível 4): problema com extrema necessidade de correção antes do lançamento.

O Peso de uma violação ( $W_V$ ) equivale ao nível de gravidade apontado na inspeção heurística, ou seja, o peso da gravidade da violação é o valor correspondente ao nível de gravidade atribuído, por exemplo, uma violação considerada catastrófica de nível 4, seu peso correspondente é 4, e assim para os demais níveis.

Portanto, o índice  $UI_{ERP}$  considera dois fatores: (i) o grau de importância dos aspectos de usabilidade atribuído pelos usuários de sistemas ERP ( $S_{UA}$ ) e (ii) o nível de gravidade das violações heurísticas atribuído pelos inspetores de usabilidade ( $W_V$ ). O índice é calculado pelo quociente da soma do produto dos pesos da gravidade da violação ( $W_V$ ) e *Scores* dos aspectos de usabilidade ( $S_{UA}$ ) pela soma dos *Scores*, conforme Equação 5.1.

$$UI_{ERP} = \frac{\sum(W_V * S_{UA})}{\sum(S_{UA})} \quad (5.1)$$

Onde:

$W_V$  = Peso (*Weight*) da violação (V);

$S_{UA}$  = *Score* do aspecto de usabilidade (UA) (cf. Tabelas 5.3 e 5.4).

Todavia, o índice  $UI_{ERP}$  pode ser variável se a gravidade das violações for considerado um fator opcional. Neste caso, o índice  $UI_{ERP}$  pode ser calculado através da Equação 5.2. Na versão simplificada o índice  $UI_{ERP}$  é calculado pelo quociente da soma do produto do número de violações (V) e *Scores* dos aspectos de usabilidade ( $S_{UA}$ ) pela soma dos *Scores*.

$$UI_{ERP} = \frac{\sum(V * S_{UA})}{\sum(S_{UA})} \quad (5.2)$$

Onde:

V = número de violações;

$S_{UA}$  = *Score* do aspecto de usabilidade (UA) (cf. Tabelas 5.3 e 5.4).

Para calcular o grau de usabilidade das aplicações desenvolvidas pelas equipes do estudo experimental serão analisados os resultados das duas versões da métrica  $UI_{ERP}$ . A priori não foi definida uma escala para classificar o grau de usabilidade de cada aplicação, uma vez que mais testes seriam necessários para determinar uma escala mais precisa. Desta forma, uma visão descritiva mais abrangente será adotada, estabelecendo-se que quanto menor for o índice,

menores são os problemas de usabilidade, e ao contrário, quanto maior for índice, maiores são os problemas de usabilidade.

## 5.5 Desenvolvimento da experimentação

Uma vez delimitado o escopo do experimento e estabelecida a métrica de usabilidade, as próximas atividades são: o planejamento do experimento, a execução do experimento e a análise e interpretação dos resultados (WOHLIN et al., 2012). Nas próximas Subseções serão descritos os detalhes de cada atividade realizada.

### 5.5.1 Planejamento do experimento

A fase de planejamento consistiu em: (a) definir o contexto; (b) formular hipóteses nulas e alternativas, (c) selecionar as variáveis incluindo os tratamentos; (d) selecionar os participantes, (e) definir o desenho experimental; e (f) preparar instrumentação, como por exemplo, formulários de apoio e de coleta de dados e diretrizes para execução das atividades experimentais.

**a) Seleção do contexto:** O experimento foi realizado durante 14 semanas em contexto universitário. Os participantes são alunos do 6º semestre do curso de Bacharelado em Ciência da Computação da Universidade Federal de São Carlos (UFSCar - campus Sorocaba), matriculados nas disciplinas de Projeto de Desenvolvimento de Software (PDS) e Interação Humano-Computador (IHC) (2º semestre de 2014).

**b) Formulação das hipóteses:** O estudo experimental foi utilizado para testar as seguintes hipóteses:

- Hipótese Nula ( $H_0$ ): Não existe diferença na usabilidade da aplicação desenvolvida utilizando o ScrumUX ou o Scrum tradicional.
- Hipótese Alternativa ( $H_A$ ): A utilização da abordagem ScrumUX permite o desenvolvimento de aplicações com melhor usabilidade.

**c) Seleção das variáveis:** A variável dependente selecionada para o experimento é o grau de usabilidade das aplicações desenvolvidas. As variáveis independentes são o processo de desenvolvimento, a aplicação desenvolvida e as tecnologias de desenvolvimento. Uma vez que o objetivo era investigar as consequências do uso da abordagem ScrumUX com relação a usabilidade das aplicações desenvolvidas, o processo de desenvolvimento foi estabelecido como o fator que receberia tratamentos distintos e cujo efeito deveria ser observado na variável dependente. As demais variáveis independentes foram mantidas constantes durante o estudo.

**d) Seleção dos participantes:** A seleção dos participantes foi feita através de amostragem não probabilística por conveniência (WOHLIN et al., 2012). Participaram do estudo 19 alunos do 6º semestre do curso de Bacharelado em Ciência da Computação da UFSCar – campus

Sorocaba, sendo que 15 alunos estavam matriculados tanto na disciplina de Projeto de Desenvolvimento de Software (PDS), como na disciplina de IHC, e 4 alunos estavam matriculados somente na disciplina de Projeto de Desenvolvimento de Software (PDS). Os alunos foram divididos em 5 equipes, sendo 4 equipes de 4 alunos e 1 equipe de 3 alunos. As equipes foram distribuídas em dois grupos: experimental e controle. Com a finalidade de equilibrar os grupos experimentais e de controle, as duas equipes mais fortes (considerando a média final de aprovação em disciplinas passadas) foram distribuídas uma para cada grupo. A equipe de 3 alunos fazia parte do grupo experimental. Todos os alunos do grupo experimental estavam matriculados em ambas as disciplinas (PDS e IHC).

**e) Desenho experimental:** Em desenhos experimentais a seleção dos membros dos grupos de controle e do grupo experimental é feita de forma aleatória (A). Contudo, em ES nem sempre é possível escolher os membros do grupo experimental e de controle de forma aleatória. Neste caso, os desenhos experimentais onde a seleção dos grupos não é aleatória (N) são chamados de quase experimentos (WOHLIN et al., 2012).

A Tabela 5.5 apresenta o modelo experimental selecionado para este estudo que é do tipo um fator (abordagem de desenvolvimento) com dois tratamentos (ScrumUX e Scrum tradicional) (WOHLIN et al., 2012).

Tabela 5.5: Desenho experimental

Equipe	Grupo	<i>design</i>
1	Experimental [E]	N X O
2		
3	Controle [C]	N O'
4		
5		

N = grupos não equivalentes

X = tratamento/intervenção (ScrumUX)

O = observação/medida ( $UI_{ERP}$ )

A intervenção tem um efeito positivo se:

$$\mu UI_{ERP}(O) \ll \mu UI_{ERP}(O')$$

Um código gráfico foi empregue para sintetizar as características do desenho experimental (SHADISH; COOK; CAMPBELL, 2002), onde o **N** representa os grupos não equivalentes (seleção não aleatória), o **X** representa a intervenção sobre a variável experimental, cujos efeitos serão observados e o **O** representa o efeito que será observado ou medido. A intervenção terá um efeito positivo se a média do índice de usabilidade ( $\mu UI_{ERP}$ ) do grupo experimental (O) for menor que a média do índice de usabilidade ( $\mu UI_{ERP}$ ) do grupo de controle (O').

**f) Instrumentação:** Os materiais necessários para apoiar os participantes durante o experimento foram previamente planejados, compreendendo a definição da aplicação que seria desen-



volvida, ferramentas de apoio no gerenciamento do projeto, e outros artefatos para apoiar as atividades das equipes, bem como os instrumentos para coleta de dados e medição.

### 5.5.1.1 Execução do experimento

Seguindo o protocolo que foi definido no planejamento, o experimento foi executado em três etapas:

**1) Preparação:** Nesta etapa elaboraram-se os materiais planejados para a instrumentação do experimento, conforme apresentado a seguir.

**Aplicação.** Definiu-se que ambos os grupos (experimental e de controle) deveriam desenvolver uma aplicação web referente a um módulo ERP para venda de produtos perecíveis (alimentício ou medicamentos). O cenário do problema a ser resolvido foi fornecido por um profissional da indústria de desenvolvimento de software (Apêndice B).

**Ferramentas.** Duas ferramentas foram adotadas para apoiar as equipes no gerenciamento dos projetos e para a coleta de dados. A primeira ferramenta chamada Agile Wrap (TECHNOLOGY, 2011) é uma plataforma web de gerenciamento de projetos ágeis. Embora não seja uma ferramenta específica para projetos Scrum, o ambiente Agile Wrap pode ser personalizado para o uso de terminologias adotadas na metodologia. O sistema permite o planejamento, manutenção e acompanhamento *online* de Backlogs, *User Stories*, tarefas, Sprints, defeitos e novas versões (*releases*), possibilitando o trabalho de equipes distribuídas. A assinatura da ferramenta é gratuita no plano até cinco usuários, portanto cada equipe teria a sua inscrição. A segunda ferramenta, chamada UX Designer, foi desenvolvida pela autora deste projeto juntamente com uma aluna de Iniciação Científica para auxiliar no planejamento e monitoramento das atividades de *design*. Trata-se de um ambiente web onde as equipes do grupo experimental (equipes 1 e 2) concentrariam os artefatos de *design* (protótipos, Lean Personas, cenários de interação e Protocolo de Comunicação) produzidos durante o projeto de desenvolvimento de software seguindo a abordagem ScrumUX. Um conjunto de instruções sobre o uso das duas ferramentas foi preparado para auxiliar as equipes (Apêndices E e F) .

**Artefatos de apoio.** Os modelos dos artefatos recomendados na abordagem ScrumUX (Lean Personas, Protocolo de Comunicação e UserX Stories) foram preparados para apoiar as equipes do grupo experimental. Em relação ao material utilizado na inspeção de usabilidade, foram elaborados dois modelos de formulários para listar problemas de usabilidade identificados, sendo um modelo para o grupo de controle e outro para o grupo experimental, igual ao primeiro, mas com um campo a mais para identificar as personas (Apêndice C). Além dos formulários de inspeção, providenciou-se uma planilha de heurísticas por perspectivas ERP de apresentação e suporte a tarefas com orientações para apoiar a atividade de inspeção (Apêndice D).

**Instrumentos para coleta de dados.** Parte das informações seriam coletadas através das duas ferramentas de apoio (Agile Wrap e UX Designer), e outros dados seriam coletados por meio dos *feedbacks* semanais e entrevistas com membros das equipes.

A Tabela 5.6 resume os materiais usados por cada um dos grupos, separados em cinco categorias: aplicação, ferramentas, artefatos Scrum, artefatos de Dxl e instrumentos de coleta de dados.

Tabela 5.6: Materiais usados pelos grupos experimental e de controle

Categoria	Grupo Experimental		Grupo de Controle		
	Equipe 1	Equipe 2	Equipe 3	Equipe 4	Equipe 5
Aplicação	Venda de e produtos perecíveis (Apêndice B)	Venda de e produtos perecíveis (Apêndice B)	Venda de e produtos perecíveis (Apêndice B)	Venda de e produtos perecíveis (Apêndice B)	Venda de e produtos perecíveis (Apêndice B)
Ferramentas	Agile Wrap + UX Designer	Agile Wrap + UX Designer	Agile Wrap	Agile Wrap	Agile Wrap
Artefatos Scrum	Product Backlog, Sprint Backlog, User Story e quadro de tarefas	Product Backlog, Sprint Backlog, User Story e quadro de tarefas	Product Backlog, Sprint Backlog, User Story e quadro de tarefas	Product Backlog, Sprint Backlog, User Story e quadro de tarefas	Product Backlog, Sprint Backlog, User Story e quadro de tarefas
Artefatos de Dxl	ScrumUX: Lean Personas, Protocolo e UserX Story	ScrumUX: Lean Personas, Protocolo e UserX Story	-	-	-
	Formulário de Inspeção (Apêndice C)	Formulário de Inspeção (Apêndice C)	Formulário de Inspeção (Apêndice C)	Formulário de Inspeção (Apêndice C)	-
	Planilha Heurísticas PERP (Apêndice D)	Planilha Heurísticas PERP (Apêndice D)	Planilha Heurísticas PERP (Apêndice D)	Planilha Heurísticas PERP (Apêndice D)	-
Instrumentos para coleta de dados	Planilhas de acompanhamento (Apêndice G)	Planilhas de acompanhamento (Apêndice G)	Planilhas de acompanhamento (Apêndice G)	Planilhas de acompanhamento (Apêndice G)	Planilhas de acompanhamento (Apêndice G)

Antes da execução dos projetos pelos grupos, foram organizadas duas palestras, sendo a primeira ministrada para as cinco equipes com o objetivo de (1) apresentar as diretrizes gerais do projeto; (2) abordar conceitos Scrum e escrita de *User Stories* seguindo o modelo de Mike Cohn (2010); e (3) sobre procedimento para o uso da ferramenta Agile Wrap junto com as instruções de funcionamento. A segunda palestra foi ministrada somente para as equipes do grupo experimental com o objetivo de: (1) apresentar a ferramenta UX Designer junto com procedimentos de uso e confecção dos artefatos de *design*; e (2) sobre orientações sobre a escrita das UserX Stories.

**2) Execução:** Estabeleceu-se junto às equipes que o projeto seria dividido em 2 Sprints de 6 semanas cada. Todos os conhecimentos necessários para os alunos, tais como conceitos e

técnicas (tanto de PDS como IHC) eram previamente trabalhados em sala de aula. As quatro equipes que faziam parte do projeto integrado (IHC e PDS) seguiram o cronograma de entregas e apresentações dividindo o projeto em cinco fases, conforme Tabela 5.7. A equipe E5 seguiria somente o cronograma de apresentações da disciplina de PDS com duas entregas coincidindo com as últimas semanas de cada Sprint.

Tabela 5.7: Cronograma do projeto integrado (PDS-IHC)

Fases	Entregas e apresentações
Fase 1 - Sprint 1 (2 <sup>a</sup> semana)	- Definição dos requisitos e escrita das histórias - Características do domínio e público alvo
Fase 2 - Sprint 1 (6 <sup>a</sup> semana)	- 50% dos requisitos implementados
Fase 3 - Sprint 2 (4 <sup>a</sup> semana)	- Demais requisitos implementados
Fase 4 - Sprint 2 (5 <sup>a</sup> semana)	- Inspeção heurística por outro grupo
Fase 5 - Sprint 2 (6 <sup>a</sup> semana)	- Produto final com as correções dos problemas de usabilidade apontados na Fase 4

A proponente do ScrumUX acompanhou as atividades dos projetos via ferramenta Agile Wrap na qual as equipes eram estimuladas a manter as informações atualizadas por meio de *feedbacks* semanais. No início as equipes tiveram dificuldades em manter o conteúdo atualizado, por não entender o funcionamento da ferramenta, e por não terem muita prática com a metodologia de desenvolvimento, porém isto foi rapidamente mitigado pela proponente.

Semanalmente foram coletados dados a respeito das tarefas e histórias em aberto, em andamento e as já realizadas, bem como tempo gasto em relação às horas estimadas. As planilhas de acompanhamento com as informações coletadas estão no Apêndice G. A ferramenta de gerenciamento de projeto serviu também para repositório das *User Stories* criadas pelas equipes (Apêndice M). Adicionalmente, as equipes foram entrevistadas na fase final do projeto (6<sup>a</sup> semana) para fornecerem informações sobre o tempo gasto nas atividades desenvolvidas durante o processo de desenvolvimento, técnicas de pesquisa utilizadas e artefatos produzidos. Quando questionadas sobre a divisão de papéis, as equipes E1, E2, E3 e E4 afirmaram que as atividades de *design* não ficaram concentradas em uma única pessoa, ao contrário as equipes atuaram como times multifuncionais (*cross-functional*).

**3) Resultados:** Os resultados englobam tanto as atividades de *design* quanto as atividades de desenvolvimento.

Nas etapas de Pesquisa e Visão do produto, quatro equipes (E1, E2, E3 e E4) realizaram as atividades de Pesquisa Competitiva e Pesquisa do Usuário: (1) analisando comparativamente produtos similares utilizando as heurísticas de Nielsen e (2) coletando informações com o público alvo. A equipe E5 não realizou essas atividades, pois as mesmas estavam relacionadas somente à disciplina de IHC. A Tabela 5.8 apresenta o tempo gasto por cada equipe em cada uma das atividades, incluindo planejamento, coleta e análise.

Tabela 5.8: Pesquisa do domínio e público alvo

Equipe	A equipe coletou e analisou informações sobre o domínio e/ou produtos similares?	A equipe coletou informações sobre público alvo? Quais as técnicas utilizadas para de pesquisa das necessidades do usuário?	
	Tempo gasto <sup>1</sup>	Técnicas de coleta	Tempo gasto <sup>1</sup>
E1	240 min	Card sorting	3 dias
E2	300 min	Entrevista	280 min
E3	600 min	Entrevista	300 min
E4	300 min	Entrevista	480 min
E5	-	-	-

<sup>1</sup> valores aproximados informados pelos membros da equipe

As quatro equipes gastaram em média 6 horas para concluir a análise comparativa de diferentes produtos, selecionados por conveniência atendendo dois critérios: disponibilidade de acesso gratuito e sistemas exclusivamente Web.

No total, foram selecionados nove sistemas: GestãoJa<sup>1</sup>, wEstoque<sup>2</sup>, CGS ERP<sup>3</sup>, WK ERP<sup>4</sup>, ERP Next<sup>5</sup>, WebERP<sup>6</sup>, ERP Now<sup>7</sup>, Tiny<sup>8</sup> e Bling<sup>9</sup>. As equipes E1, E2 e E3 avaliaram dois sistemas, enquanto a equipe E4 avaliou três sistemas distintos. A Tabela 5.9 apresenta os nove módulos dos ERPs inspecionados pelas quatro equipes, bem como as heurísticas violadas por cada sistema.

Tabela 5.9: Análise comparativa de produtos similares

Equipe	ERP	Módulo	Heurísticas violadas									
			H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
E1	gestaoJa	Vendas	1	2	1	3	5	0	1	2	1	1
	wEstoque	Vendas	3	1	1	3	6	2	0	0	1	0
E2	CGS ERP	Vendas	2	1	1	0	4	0	2	2	3	1
	WK ERP <sup>4</sup>	Vendas	2	1	0	0	1	2	1	1	1	1
E3	ERP Next	Estoque	5	0	2	0	3	0	0	0	0	1
	WebERP <sup>6</sup>	Contábil	1	1	2	0	3	0	0	3	2	2
E4	ERP Now	Vendas	1	0	0	2	2	0	0	0	1	2
	Tiny	Vendas	2	0	1	1	0	0	0	1	1	0
	Bling	Vendas	1	1	1	2	1	1	0	1	1	1

Os resultados gerais da análise comparativa dos nove módulos apontam que as heurísticas

<sup>1</sup><https://gestaoja.com.br>

<sup>2</sup><http://westoque.com.br>

<sup>3</sup><http://cybersis.com.br>

<sup>4</sup><http://www.erplitefree.com.br>

<sup>5</sup><https://erpnext.com>

<sup>6</sup><http://www.weberp.org>

<sup>7</sup><https://www.erpnow.com.br>

<sup>8</sup><https://www.tiny.com.br>

<sup>9</sup><https://www.bling.com.br>

mais violadas foram as heurísticas de “Prevenção de Erros (H5)” com 25 violações; “Visibilidade do estado do sistema (H1)” com 18 violações; “Reconhecimento, diagnóstico e recuperação de erros (H9)” e “Consistência e padrões (H4)” com 11 violações cada.

As principais violações identificadas da heurística H5 foram a falta de indicações de como preencher alguns dados de formulário, a falta de indicação de itens de preenchimento obrigatório e a ausência de verificações de consistência de informações; por exemplo, em alguns casos o sistema permitia inserir a data de entrega de um produto anterior a data da venda.

Quanto à heurística H1, as principais violações identificadas estavam relacionadas com a falta de *feedback* do sistema em algumas ações do usuário, por exemplo ao tentar realizar uma busca específica sem preencher nenhum campo, o sistema (de um determinado produto) não deixava claro o seu estado, ou seja, não era possível saber se a busca foi realizada e não encontrou nenhum resultado ou se algum erro ocorreu ao realizar a busca.

As violações da heurística H4 estavam relacionadas com a falta de padrão no uso de terminologias e de cores, por exemplo durante uma consulta ao sistema uma mensagem em vermelho alertando que determinado produto não existe, e a mesma cor usada para um produto que existe, dando a impressão ao usuário de algo errado.

As violações da heurística H9 aparecem, na maioria das vezes, relacionadas com as violações da heurística H5 quando, por exemplo, após uma ação equivocada do usuário, ou o sistema não emite nenhuma mensagem de aviso de erro ou quando exibe, as mensagens não ajudam na recuperação por serem de difícil entendimento (uso de linguagem técnica).

Segundo as declarações das equipes, visualizando as deficiências dos sistemas concorrentes, foi possível destacarem quais seriam os diferenciais das suas aplicações. As equipes E1 e E2 aparentemente tiveram maior foco em minimizar problemas relacionados com prevenção de erros; enquanto que as equipes E3 e E4 pareceram estar mais preocupadas com a apresentação visual, tais como disposição de elementos de interface e o uso de padrões.

Quanto às técnicas utilizadas para coletar informações sobre o público alvo (Tabela 5.8), as equipes E2, E3 e E4 utilizaram entrevistas para definir um perfil dos usuários; cada equipe entrevistou, em média, sete pessoas. A equipe E1 aplicou o *card sorting* com sete potenciais usuários, usando a ferramenta *online* chamada WebSort<sup>10</sup>. O objetivo da equipe foi verificar qual seria o melhor agrupamento para as informações, bem como verificar o entendimento dos usuários sobre as terminologias que seriam utilizadas. A partir dos resultados obtidos na pesquisa do domínio e do público alvo as quatro equipes mencionadas elaboraram as personas e os cenários de interação, gastando em média 50 minutos para execução de cada tarefa (Tabela 5.10).

As equipes E1 e E2 utilizaram o sistema UX Designer para gerar personas conforme modelo recomendado na abordagem ScrumUX (Lean Personas). Alguns exemplos de personas e cenários criados pelas equipes estão nos Apêndices H e I.

Nas etapas de *Design* e criação do Product Backlog, todas as equipes escreveram as histórias de usuário no software Agile Wrap. Entretanto, somente as equipes do grupo experimental (E1 e E2) utilizaram o artefato UserX Stories. Este artefato estimula a definição dos critérios de aceitação, conseqüentemente somente estas equipes usaram este recurso. Antes de expandir

---

<sup>10</sup>Disponível em: [www.websort.net](http://www.websort.net)

Tabela 5.10: Personas e cenários de interação

Equipe	Criaram as personas? Quantas?		Criaram os cenários? Quantos?	
	Qtde.	Tempo gasto <sup>1</sup>	Qtde.	Tempo gasto <sup>1</sup>
E1	2	45 min	4	60 min
E2	1	30 min	4	45 min
E3	2	50 min	2	60 min
E4	2	60 min	3	40 min
E5	-	-	-	-

<sup>1</sup> valores aproximados informados pelos membros da equipe

as histórias de usuário, as equipes E1 e E2 geraram o Protocolo de Comunicação (artefato ScrumUX) na ferramenta UX Designer: (i) analisando as soluções de *design*; (ii) criando o plano de validação; e (iii) listando os itens (soluções/recomendações de *design*) relacionados com as personas e as heurísticas de Nielsen (Apêndice K). Posteriormente, os itens listados no Protocolo foram escritos no formato de UserX Stories (Apêndice L).

A equipe E5 criou somente 5 histórias, enquanto as demais equipes criaram em média 12 histórias gastando de 3 a 4 horas na atividade (Tabela 5.11). Este fato denota que a equipe E5 não trabalhou detalhes das histórias, se concentrando somente nos requisitos globais do sistema; ao passo que as demais equipes quebraram as histórias para atender especificações mais detalhadas conforme a descrição do cenário, incluindo as especificações relacionadas com a interação do usuário.

Tabela 5.11: User Stories e protótipos

Equipe	Criaram as User Stories? Quantos?		Definiram os critérios de aceitação?		Criaram protótipos?		
	Qtde.	Tempo gasto <sup>1</sup>	Qtde.	Tempo gasto <sup>1</sup>	Fidelidade	Nº telas	Tempo gasto <sup>1</sup>
E1	10	180 min	13	100 min	baixa <sup>3</sup>	5	360 min
E2	10	240 min	20	180 min	baixa <sup>2</sup>	3	420 min
E3	14	180 min	-	-	baixa <sup>3</sup>	5	180 min
E4	16	240 min	-	-	-	-	-
E5	5	120 min	-	-	-	-	-

<sup>1</sup> valores aproximados informados pelos membros da equipe

<sup>2</sup> protótipos de papel (*sketches*)

<sup>3</sup> protótipos de baixa fidelidade usando o software Balsamiq

Analisando-se a escrita das histórias (Apêndice M), observa-se que as equipes do grupo experimental desenvolveram melhor o conteúdo a partir das orientações sobre UserX Stories, inserindo personas e as heurísticas de Nielsen em suas histórias. Embora as equipes E3 e E4 tivessem elaborado personas, as mesmas não foram consideradas em suas histórias, além disso ambas as equipes não tiveram a preocupação de definir os critérios de aceitação. Observou-se, ainda, que a equipe E5 não demonstrou foco nas necessidades do usuário, uma vez que suas

histórias foram escritas na perspectiva do *Product Owner*.

Quanto à prototipação, somente três equipes criaram protótipos (Apêndice J). As equipes E1, E2 e E3 elaboraram protótipos de baixa fidelidade, sendo que a equipe E2 optou por protótipos de papel (*sketches*), enquanto as equipes E1 e E3 utilizaram o software Balsamiq<sup>11</sup> como ferramenta de apoio. A equipe E2 utilizou os protótipos de papel para refinar o *design* de forma iterativa, realizando testes com quatro usuários representando diferentes personas. A equipe E2 declarou que os *sketches* foram muito úteis na definição da interface, pois possibilitou validar o *design* diretamente com usuários antes da codificação, coletando informações relacionadas aos campos do formulário, botões, terminologias e observando as dificuldades de interação. Contudo, comentaram que o tempo gasto na preparação dos protótipos de papel pode ser uma desvantagem da técnica, pois a atividade consumiu aproximadamente 12 horas de trabalho para preparar *sketches* de três telas. As outras duas equipes (E1 e E3) não utilizaram os protótipos em papel para validar o *design* com os usuários antes de iniciarem a codificação. A equipe E3 declarou o menor tempo gasto (3 horas) para confeccionar os protótipos de cinco telas.

Nas etapas de planejamento de Sprint e definição de Sprint Backlog, as cinco equipes fizeram a decomposição das histórias de usuários em um conjunto de tarefas, estimando o tempo de conclusão (horas estimadas). Na execução das Sprints, durante a atividade de implementação, atualizavam as informações no sistema Agile Wrap inserindo as horas gastas, conforme as tarefas eram feitas. A Tabela 5.12 apresenta os dados referente o número de tarefas, as horas estimadas e as horas gastas por Sprint, e ainda a média do total de horas gastas nas duas Sprints pelo número total de tarefas.

Tabela 5.12: Atividades de implementação

Equipe	Sprint 1			Sprint 2			Total
	Nº. de Tarefas	Horas estimadas <sup>1</sup>	Horas gastas <sup>1</sup>	Nº. de Tarefas	Horas estimadas <sup>1</sup>	Horas gastas <sup>1</sup>	Horas/Tarefas <sup>2</sup>
E1	18	89	106	7	74	77,5	7,34
E2	14	81	69	19	77	74	4,33
E3	36	150	133,5	22	124	123,25	4,43
E4	13	63	61	19	110	103	5,13
E5	15	74	69,65	8	48	43,35	4,91

<sup>1</sup> valores coletados via software de gerenciamento do projeto Scrum (Agile Wrap)

<sup>2</sup> média do total de horas gastas pelo número total de tarefas (Sprint 1 + Sprint 2)

Observa-se que a equipe E1 foi a única equipe que gastou um pouco mais horas do que havia estimado. A equipe E4 determinou o maior número de tarefas (total de 58 tarefas), enquanto a equipe E5 determinou o menor número de tarefas (total de 23 tarefas). A média geral de horas gastas por tarefa foi de 5,23 horas, sendo que a equipe E1 registrou o maior média de horas por tarefa (7,34). A equipe E2, que havia gasto o maior número de horas na confecção de protótipos em papel, registrou a menor média de horas por tarefa (4,33).

A equipe E5 teve uma média relativamente baixa de horas por tarefa, porém, não realizou o mesmo número atividades que os outros grupos. A equipe E3 também teve uma média baixa

<sup>11</sup>Disponível em: [www.balsamiq.com](http://www.balsamiq.com)

de horas por tarefa, mas não usou os artefatos do ScrumUX e não fez validação com protótipo antes da codificação. A equipe E1 teve a maior média de horas por tarefa, mas determinou um número de tarefas bem menor que a equipe E3.

Observou-se que, as equipes do grupo experimental, em comparação as demais equipes, tiveram que realizar um número maior de atividades por causa da abordagem ScrumUX, contudo pode-se dizer que estas equipes tiveram um melhor direcionamento quanto às práticas de Dxl. A equipe E1, por exemplo, gastou mais tempo nas atividades de pesquisa e avaliação ao aplicar a técnica do *card sorting*.

Quanto aos testes funcionais e às avaliações de usabilidade, todas as equipes declararam ter realizado vários testes funcionais de forma manual durante a codificação; e também realizaram avaliações de usabilidade empregando testes com usuários e a avaliação heurística, com exceção da equipe E5 que fez somente os testes funcionais. A Tabela 5.13 apresenta o tempo que as equipes gastaram com testes funcionais, testes de usabilidade e avaliações heurísticas. Com relação aos testes de usabilidade, apresenta-se o número de usuários envolvidos na atividade; e quanto à avaliação heurística, são apresentados o número de violações identificadas e a média de violações calculada por quantidade de *User Stories*.

Tabela 5.13: Atividades de verificação

Equipe	Testes funcionais		Teste de Usabilidade		Avaliação Heurística		
	Tempo total <sup>1</sup>	Tempo gasto <sup>1</sup>	Tempo gasto <sup>1</sup>	Nº usuários	Tempo gasto <sup>2</sup>	VT <sup>3</sup>	VT/US <sup>4</sup>
E1	183,50 horas	1,50 horas	2 dias	5	90 mim	40	3,64
E2	143,00 horas	3,67 horas	175 min	5	76 min	60	6,00
E3	256,75 horas	45,75 horas	120 min	6	65 min	90	5,33
E4	164,00 horas	8,00 horas	150 min	6	115 min	80	5,71
E5	113,00 horas	15,50 horas	-	-	95 mim	97	19,40

<sup>1</sup>valores aproximados informados pelos membros da equipe

<sup>2</sup>valores informados pela equipe de inspeção

<sup>3</sup>número total de violações

<sup>4</sup>total de violações pela quantidade de User Stories

Com relação ao tempo gasto nos testes funcionais, observa-se que existe uma diferença grande nos tempos declarados entre as equipes. Enquanto as equipes E1 e E2 declararam que o tempo dedicado à atividade de teste, em relação ao tempo total de implementação, foi menor que 5%, a equipe E3, por exemplo, declarou que dedicou, aproximadamente, 18% do tempo de implementação para testes funcionais. Nesta questão as equipes podem ter perdido a noção de quanto tempo foi realmente gasto com testes funcionais, uma vez que não foram orientadas para registrarem o tempo gasto de forma rígida e precisa. Não obstante, essa questão não afeta os resultados do presente estudo experimental.

A respeito dos testes de usabilidade, a equipe E2, como informado anteriormente, foi a única equipe que realizou testes com usuário antes de iniciar a codificação, utilizando protótipos de baixa fidelidade. As demais equipes (E1, E3 e E4) realizaram testes de usabilidade em estado mais adiantado do projeto, utilizando o protótipo funcional. Com exceção da equipe E2, que



levou dois dias para preparar e executar os testes com usuários, as demais equipes gastaram em média 2,5 horas para realizar os testes, envolvendo de 5 a 6 usuários.

Para a atividade de avaliação heurística definiu-se que a equipe E1 faria a inspeção na aplicação da equipe E2, e a equipe E2 inspecionaria a aplicação de E1; enquanto as equipes E3 e E4 inspecionariam a aplicação uma da outra. Uma vez que os resultados da avaliação heurística seriam utilizados para comparar a usabilidade das aplicações desenvolvidas pelas equipes do grupo experimental com as aplicações desenvolvidas pelas equipes do grupo de controle, a aplicação da equipe E5 também foi inspecionada, porém, por uma equipe de alunos aptos à realizar a atividade de inspeção e que não participavam do experimento.

As equipes tiveram o período de uma semana para executar a atividade. Todas as quatro equipes foram orientadas à: (1) utilizar as heurísticas por perspectivas ERP (HPERP), cientes de que cada questão apontada poderia violar mais de uma heurística, considerando as perspectivas de apresentação e de suporte à tarefas; (2) fazer a inspeção com base nas histórias de usuário para compreenderem o que havia sido implementado; e (3) registrar no formulário de inspeção os problemas de usabilidade relacionado-os com as histórias, com a finalidade de identificar problemas recorrentes relacionados com mais de uma história. Somente as equipes E1 e E2 teriam que identificar as personas afetadas, uma vez que as UserX Stories continham tal informação.

As equipes gastaram em média 88,2 minutos na atividade de inspeção. Seguindo o cronograma de entregas e apresentação (Tabela 5.7), cada equipe apresentou os resultados sumarizados e encaminhou o formulário de inspeção às equipes inspecionadas para que pudessem efetuar os acertos finais.

A quantidade total de violações apresentada na Tabela 5.13, inclui o número de histórias relacionadas ao problema identificado e todas as heurísticas violadas. A aplicação com mais heurísticas violadas foi a aplicação da equipe E5 (97 violações). As equipes do grupo experimental (E1 e E2) obtiveram os menores números de violações.

Ao calcular a média do total violações pela quantidade de histórias, observou-se uma discrepância muito grande entre o resultado da equipe E5 e as demais equipes. A média da equipe E5 foi de 19,40 violações/US, ou seja, é quase quatro vezes maior que a média das outras equipes que é de 5,17 violações/US. O baixo número de histórias criadas por esta equipe elevou o número de violações/US. Quanto às demais equipes, observa-se que a diferença não é tão significativa, e que existe um equilíbrio maior com relação à quantidade de histórias e número de violações. É importante destacar que as equipes E3 e E4 desenvolveram a consciência sobre a importância da usabilidade uma vez que estavam imersos na disciplina de IHC, e que embora não estivessem seguindo a abordagem ScrumUX, estas equipes conseguiram atingir resultados que estão mais próximos do grupo de controle.

### 5.5.1.2 Análise e interpretação dos resultados

Na fase de análise e interpretação dos resultados a principal tarefa é verificar se a diferença entre as observações são significativas. A análise das heurísticas mais violadas permite verificar os pontos fracos de cada uma das aplicações desenvolvidas em relação à usabilidade. As violações identificadas pelos inspetores por equipe nas perspectivas de apresentação e suporte à tarefas

são apresentadas, respectivamente, nas Figuras 5.1 e 5.2.

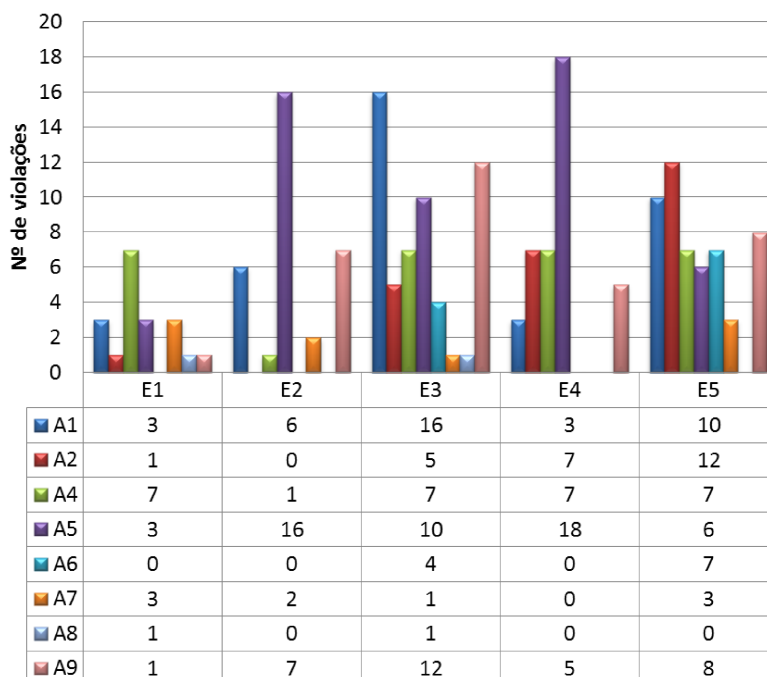


Figura 5.1: Heurísticas violadas por perspectiva de apresentação

Analisando os dados apresentados na Figura 5.1, observa-se que, na perspectiva de apresentação, as equipes E2 e E4 obtiveram um maior número de violações na heurística de “Prevenção de Erros (A5)”; e a equipe E1 na heurística de “Consistência e Padrões (A4)”. A equipe E3 obteve um maior número de violações na heurística de “Visibilidade e estado do sistema (A1)”, mas também um número considerável de violações nas heurísticas A5 e A9. A equipe E5 teve violações concentradas em maior número na heurística “Concordância entre o sistema e o mundo real (A2)” e também na heurística A1. Os problemas de usabilidade identificados na aplicação da equipe E5 demonstram que houve pouca preocupação em usar um linguagem mais próxima do entendimento do usuário, onde muitas convenções não correspondiam ao mundo real.

Na perspectiva de suporte à tarefas, conforme dados apresentados na Figura 5.2, observa-se que as equipes E1, E2 e E4 obtiveram um maior número de violações na heurística de “Prevenção de Erros (S5)”, enquanto que as violações das equipes E3 e E5 se concentraram mais na heurística de “Reconhecimento, diagnóstico e recuperação de erros (S9)”, mas com um número grande de violações também na heurística S5.

No geral, os problemas de usabilidade estavam relacionados mais com campos de validação dos formulários, indicação de campos obrigatórios e indicação dos formatos permitidos. Agravando essa situação, a maioria das aplicações não ofereciam suporte para à recuperação de erros. Pode-se dizer que, em relação às demais equipes, a equipe E5 teve um número de violações bem alto relacionados à falta de cuidados para prevenir usuários de ações equivocadas ou por não oferecer soluções à recuperação de erros, levando-se em conta que o número de funcionalidades implementadas pela equipe foi consideravelmente menor. Ao contrário das demais equipes, a equipe E5 relacionou uma quantidade mínima de itens nos formulários da aplicação, com poucos

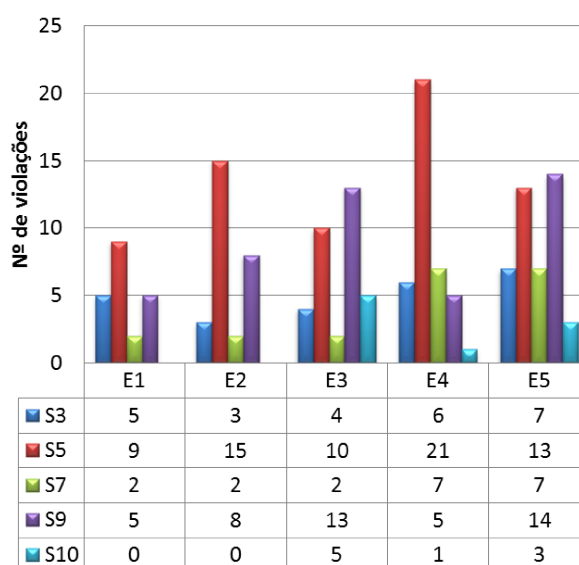


Figura 5.2: Heurísticas violadas por perspectiva de suporte à tarefas

campos de validação. Nesse sentido, as aplicações que foram mais trabalhadas ficaram sujeitas a terem um número maior de violações.

**1) Nível de gravidade:** A Figura 5.3 apresenta o número total de violações em ambas perspectivas (apresentação e suporte à tarefas) distribuído por nível de gravidade atribuído pelos inspetores de usabilidade. No geral, a maior parte das violações identificadas pelas equipes de inspeção foram consideradas de pequena gravidade (49,38%).

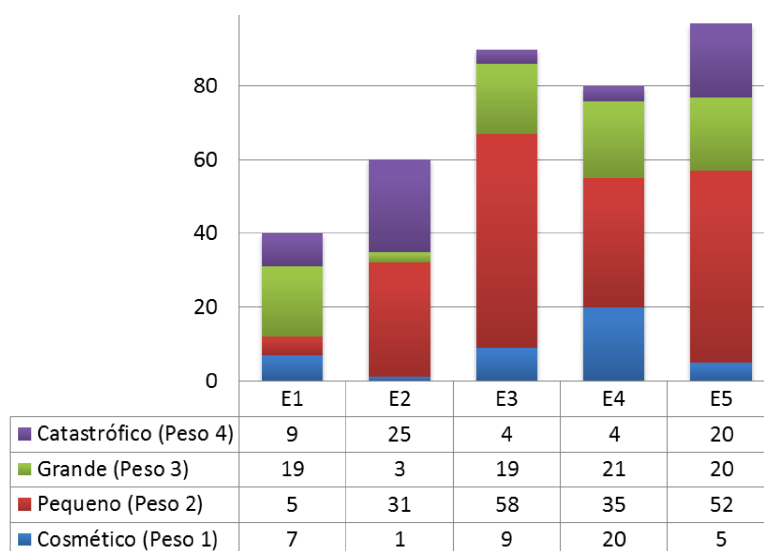


Figura 5.3: Gravidade das violações

Na visão dos inspetores de usabilidade, as equipes E2 e E5 tiveram um número maior de violações com gravidade catastrófica (nível 4), e a equipe E3, apesar de ser a segunda equipe

com maior número de violações, teve 64% de violações consideradas de pequena gravidade (nível 2). A equipe E1, com o menor número de violações, teve 47% das violações consideradas de grande gravidade (nível 3) e 22% das violações consideradas catastróficas. A equipe E4, a terceira equipe com maior número de violações, teve 43,75% de violações pequenas, 25% de violações cosméticas, e assim como a equipe E3 tiveram menos de 1% de violações consideradas catastróficas.

A Tabela 5.14 apresenta as médias de gravidade por perspectivas por equipe e as médias de gravidade geral por equipe e por grupo experimental (E) e de controle (C).

Tabela 5.14: Gravidade das violações heurísticas

Média	Grupo E		Grupo C		
	E1	E2	E3	E4	E5
Nível de Gravidade <small>Apresentação</small>	2,47	2,88	2,21	1,95	2,30
Nível de Gravidade <small>Suporte à tarefas</small>	3,00	2,86	2,18	2,28	2,89
Nível de Gravidade <small>Geral por equipe</small>	2,74	2,87	2,20	2,12	2,60
Nível de Gravidade <small>Geral por grupo</small>	2,80		2,30		

Os valores apresentados na Tabela 5.14 referem-se às médias ponderadas de violações considerando os pesos de cada nível de gravidade, obedecendo-se a escala de 1 a 4. Os números sugerem que as violações do grupo experimental foram em média mais graves do que as violações do grupo de controle. Entretanto, esta análise considera somente a opinião dos inspetores em relação à gravidade das violações. Levando-se em conta que diferentes equipes de inspetores avaliaram cada uma das aplicações, pode-se dizer que, o processo de avaliação utilizado ficou sujeito a um alto grau de subjetividade. Cada equipe de inspetores pode ter tido opiniões muito divergentes no momento de determinar o nível de gravidade das violações identificadas. Presume-se que tal ameaça poderia ter sido minimizada, caso todas as aplicações tivessem sido avaliadas por uma mesma equipe de inspetores.

**2) Índice de Usabilidade:** O principal objetivo da métrica  $UI_{ERP}$  é inserir a visão do usuário ao método de inspeção. Primeiramente, os resultados das avaliações heurísticas foram utilizados para calcular o índice de usabilidade, nas perspectivas de apresentação e suporte à tarefas, considerando tanto o ponto de vista dos usuários de sistemas ERP como o ponto de vista dos inspetores de usabilidade. Para tanto, utilizou-se a primeira versão do índice  $UI_{ERP}$ , conforme Equação 5.1. As planilhas de cálculo dos índices de usabilidade estão no Apêndice N. A Tabela 5.15 sumariza os resultados apresentando o índice de usabilidade obtidos por cada uma das equipes, em ambas as perspectivas, bem como o índice geral por equipe e por grupo (experimental e controle).

Considerando que a métrica  $UI_{ERP}$  mede o grau de usabilidade, e que quanto menor for o seu índice menores são os problemas de usabilidade, pode-se dizer que a equipe E1, comparada às demais equipes, desenvolveu uma aplicação com melhor usabilidade, tanto na perspectiva de apresentação (3,87) quanto na perspectiva de suporte à tarefas (6,27). A aplicação com a pior usabilidade é da equipe E5, embora tenha obtido um índice menor na perspectiva de apresentação comparado ao índice da equipe E3. A equipe E4 conseguiu um índice geral um pouco menor

que a equipe E2 por causa dos aspectos de usabilidade na perspectiva de apresentação. O índice geral da equipe E3 é maior que o índice geral da equipe E2, embora tenha obtido um índice menor na perspectiva de suporte à tarefas. Mas, analisando o índice geral por grupo observa-se que o grupo experimental obteve a menor média, sugerindo um grau maior de usabilidade.

Tabela 5.15: Índice de Usabilidade (Equação 5.1)

Métrica	Grupo E		Grupo C		
	E1	E2	E3	E4	E5
UI Apresentação	3,87	8,46	10,58	7,03	10,33
UI Suporte à tarefas	6,27	8,10	7,56	9,04	12,77
UI Geral por equipe	5,07	8,28	9,07	8,04	11,55
UI Geral por grupo	6,68		9,55		

Aplicando-se a versão simplificada do índice  $UI_{ERP}$  (Equação 5.2), somente o grau de importância dos aspectos de usabilidade atribuído pelos usuários é considerado, deixando-se de fora a visão dos inspetores a respeito da gravidade das violações. A Tabela 5.16 apresenta os índices de usabilidade sem ponderar a gravidade das violações de cada uma das equipes em ambas as perspectivas, o índice geral por equipe e o índice geral do grupo experimental e do grupo de controle.

Tabela 5.16: Índice de Usabilidade (Equação 5.2)

Métrica	Grupo E		Grupo C		
	E1	E2	E3	E4	E5
UI Apresentação	1,55	2,91	4,79	3,55	4,39
UI Suporte à tarefas	2,09	2,83	3,48	3,96	4,41
UI Geral por equipe	1,82	2,87	4,13	3,76	4,40
UI Geral por grupo	2,35		4,10		

Os resultados calculados a partir da visão dos usuários, sugerem uma diferença, aparentemente, mais significativa entre os dois grupos referente ao grau de usabilidade nas duas perspectivas. Os resultados confirmam que, a equipe E1, em relação às demais equipes, desenvolveu uma aplicação com melhor usabilidade tanto na perspectiva de apresentação (2,47) quanto na perspectiva de suporte à tarefas (3,00), e indicam que a equipe E5 obteve o pior índice de usabilidade em ambas perspectivas (respectivamente 4,39 e 4,41). Em relação ao cálculo feito anteriormente, observa-se que a diferença entre as equipes E1 e E5 aumentou sutilmente. Destaca-se que os resultados da equipe E2, nesta versão de cálculo, estão melhores do que na versão anterior que considerou a gravidade das violações. Essa questão implica em refletir o quanto a avaliação da aplicação dessa equipe foi mais rígida que as demais. Nesse sentido, os resultados podem ter sido prejudicados em função da divergência de opiniões dos inspetores ao determinar a gravidade das violações.

**3) Teste das hipóteses:** com a finalidade de confirmar diferenças significativas no grau de usabilidade das aplicações desenvolvidas pelos grupos experimental e de controle, e para testar

$H_0$  e  $H_A$  (respondendo Q2), realizou-se o teste estatístico *t de Student*<sup>12</sup> (*t-test*) considerando um intervalo de confiança de 90% ( $\alpha = 0,10$ ), devido ao pequeno tamanho da amostra (DYBÅ; KAMPENES; SJØBERG, 2006). Utilizou-se o software Minitab17 (MINITAB, 2000) como ferramenta de apoio à análise estatística dos dados.

Os *boxplots* da Figura 5.4, comparando os níveis de gravidade das violações entre os grupos, referem-se aos resultados apresentados na Tabela 5.14. Observa-se que os resultados do grupo experimental e de controle não se sobrepõem, sugerindo que as violações do grupo experimental (ScrumUX) foram mais graves do que as violações do grupo de controle (Scrum Tradicional), segundo visão dos inspetores de usabilidade. O *t-test* confirmou que, no nível geral de gravidade, a diferença entre os grupos experimental e de controle é significativa (P-Value = 0,091), embora não seja significativa se considerar as perspectivas separadamente. Na perspectiva de apresentação, a diferença entre os grupos não foi significativa (P-Value = 0,265), assim como na perspectiva de suporte à tarefas (P-Value = 0,175). Tais resultados, porém, não serão usados para teste das hipóteses, pois não refletem a opinião do usuário.

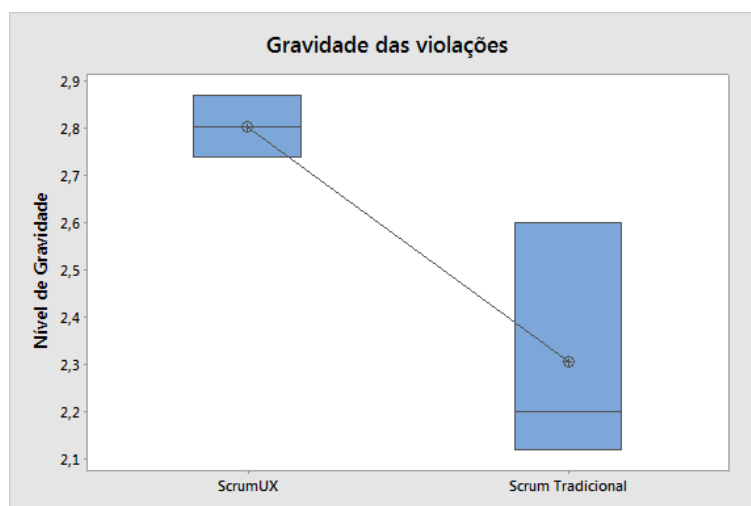


Figura 5.4: Nível de gravidade

Os *boxplots* da Figura 5.5, cujos resultados foram apresentados na Tabela 5.15, permitem comparar os índices de usabilidade de cada grupo, considerando-se o grau de importância dos aspectos de usabilidade e os pesos da violação. No referido gráfico, é possível notar que os resultados do grupo experimental e de controle se sobrepõem ligeiramente, sugerindo que as equipes que utilizaram a abordagem ScrumUX obtiveram um índice geral de usabilidade melhor que o índice das equipes que seguiram o Scrum tradicional. O *t-test*, no entanto, confirma que as diferenças dos índices gerais de usabilidade entre os grupos experimental e de controle não são significativas (P-Value = 0,373), nem na perspectiva de apresentação (P-Value = 0,435), nem na perspectiva de suporte à tarefas (P-Value = 0,285). Portanto, os resultados da análise estatística indicam que “Não existe diferença na usabilidade da aplicação desenvolvida utilizando o ScrumUX ou o Scrum tradicional”. A usabilidade das aplicações do grupo experimental e do

<sup>12</sup>O *t-test* é um teste paramétrico usado para comparar duas amostras independentes e checar se as médias de seus dados são estatisticamente diferentes e, rejeitar ou não uma hipótese nula (WOHLIN et al., 2012).

grupo de controle são similares, quando analisada a partir do ponto de vista dos usuários e dos inspetores de usabilidade. Nestes termos, deve-se aceitar a hipótese nula ( $H_0$ ), rejeitando-se a hipótese alternativa ( $H_A$ ).

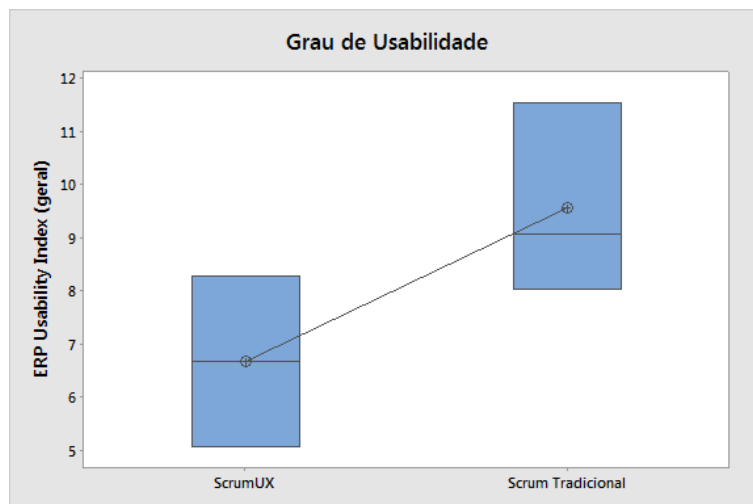


Figura 5.5: Índice de usabilidade geral (Equação 5.1)

A Figura 5.6 apresenta os *boxplots* comparando os índices de usabilidade dos grupos onde as gravidades das violações não foram consideradas (cf. Tabela 5.16).

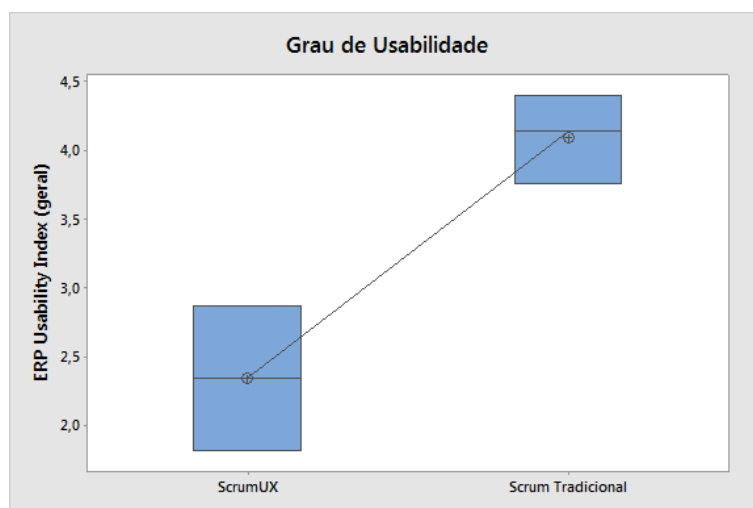


Figura 5.6: Índice de usabilidade geral (Equação 5.2)

Os resultados do grupo experimental e de controle não aparecem em sobreposição, sugerindo novamente que as equipes que utilizaram a abordagem ScrumUX obtiveram um índice geral de usabilidade melhor que o índice das equipes que seguiram o Scrum tradicional. O *t-test* confirmou que não existe diferença significativa no índice geral de usabilidade (P-Value = 0,285) e nem no índice de usabilidade na perspectiva de apresentação (P-Value = 0,233); mas, apontou uma diferença significativa entre os grupos relacionada ao índice de usabilidade

na perspectiva de suporte à tarefas (P-Value = 0,083). Portanto, os resultados da análise estatística na perspectiva de suporte à tarefas indicam que “A utilização da abordagem ScrumUX permite o desenvolvimento de aplicação com melhor usabilidade”, enquanto que na perspectiva de apresentação a usabilidade das aplicações do grupo experimental e do grupo de controle são similares. Contudo, ao considerar as duas perspectivas, deve-se aceitar a hipótese nula ( $H_0$ ), rejeitando-se a hipótese alternativa ( $H_A$ ).

**4) Avaliação qualitativa:** Com o objetivo de avaliar a qualidade as aplicações desenvolvidas pelas equipes, o profissional da indústria, que havia elaborado o cenário para o experimento, foi convidado para fornecer seu parecer a respeito das aplicações, atribuindo notas para cinco questões relacionadas com os requisitos que haviam sido solicitados às equipes. A Tabela 5.17 apresenta as notas atribuídas às equipes pelo profissional.

Tabela 5.17: Avaliação qualitativa

Questões	E1	E2	E3	E4	E5
Solução é aderente ao público alvo?	9,00	7,50	6,00	8,50	6,00
Solução é aderente ao cenário?	9,50	9,00	8,00	9,00	6,00
A funcionalidades implementadas atendem requisitos especificados no cenário?	9,50	9,00	6,00	9,00	5,00
Solução possui atributos visuais que auxiliam a navegação?	8,00	7,50	8,00	9,00	5,50
Solução é de fácil navegação e oferece suporte à tarefas?	9,00	7,00	5,00	8,00	5,00
Média geral	9,00	8,00	6,60	8,70	5,50

Em nenhum momento foi revelado ao profissional quais eram as equipes do grupo experimental. Na opinião do profissional (desenvolvedor de sistemas ERP) as equipes E1 e E4 apresentaram as duas melhores soluções. A equipe E5 recebeu as notas mais baixas, pois deixou de implementar parte dos requisitos especificados no cenário e apresentou uma solução com muitas deficiências, tais como erros graves de programação e falta de cuidado na apresentação das informações. Além disso, o profissional destacou que a referida equipe implementou uma quantidade mínima de campos nos formulários, ficando bem aquém do que foi implementado pelas outras equipes.

### 5.5.1.3 Ameaças à validade do estudo

A validade de um estudo denota a confiabilidade dos resultados, na medida que os resultados não são influenciados pelo ponto de vista subjetivo dos pesquisadores (RUNESON et al., 2012). As ameaças à validade do presente estudo foram consideradas desde a fase de planejamento, no entanto, serão discutidas nessa Seção, abordando quatro aspectos da validade: validade interna, validade externa, validade de construção e validade de conclusão (WOHLIN et al., 2012).

**1) Validade interna:** é o aspecto de validade relacionado com a confiança que se tem de que o efeito observado é realmente devido à manipulação feita, e não em função de outros fatores. Neste aspecto foram consideradas os seguintes tipos de ameaças: ameaça de instrumentação,



ameaça de interação seleção abandono e ameaça de efeito de expectativa do sujeito. Para tratar a ameaça de instrumentação, duas ferramentas para apoiar as equipes, tanto no gerenciamento do projeto Scrum como no projeto de *design*, foram de fundamental importância na coleta de dados e uma boa forma encontrada para observar os grupos à distância e manter o ritmo das equipes. O experimento ficou sujeito à ameaça de interação seleção abandono, por se estender por um longo período (12 semanas). A ameaça de interação seleção abandono é quando algum membro, por desistência, abandona o seu grupo deixando a equipe em desvantagem. Porém, não houve nenhuma desistência e as equipes se mantiveram até o final com a mesma formação original. Quanto a ameaça de efeito de expectativa do sujeito, o principal tratamento foi acompanhar com igual atenção todos os grupos, fornecendo *feedbacks* semanais, com a intenção de motivar as equipes a manterem um ritmo de desenvolvimento e conferir a atualização dos dados nas ferramentas de apoio. Esse tratamento contribuiu para o efeito positivo, conhecido como *Hawthorne Effect* (PRECHELT; UNGER, 2001), que é quando os sujeitos sabem que estão sendo observados. Estudos apontam que engenheiros de software que sabem que estão sendo observados melhoram a produtividade e a qualidade do software gerado.

**2) Validade externa:** este aspecto de validade está preocupado com até que ponto é possível generalizar os resultados, e em que medida os resultados são de interesse para um contexto mais amplo do que aquele que foi selecionado para o estudo. Neste aspecto a principal ameaça esta relacionada com a seleção dos alunos de graduação como sujeitos em ambiente acadêmico, que poderiam impedir a generalização do estudo para um ambiente industrial. No entanto, os sujeitos tiveram acesso à ferramentas, tecnologias de desenvolvimento e equipamentos atualizados e adequados para ambientes industriais, bem como desenvolveram suas aplicações a partir de cenário real, comumente solicitado no mercado. Além disso, os artefatos recomendados na abordagem ScrumUX foram construídos e validados junto com profissionais da indústria. Com relação ao tamanho das Sprints, que costumam ser de 2 a 4 semanas, observa-se que opção de ampliar o período para 6 semanas foi adequada para o experimento, pois considerou-se a carga de trabalhos acadêmicos para não sobrecarregar as equipes.

**3) Validade de construção:** é o aspecto da validade que reflete até que ponto as medidas operacionais realmente representam o que o pesquisador projetou para a pesquisa. A principal ameaça tratada nesse aspecto foi o efeito de expectativa do experimentador, buscando evitar vieses nos resultados causados por avaliação subjetiva do pesquisador. O índice de usabilidade considerando a opinião de usuários e a opinião de inspetores de usabilidade foi proposto para tratar esse efeito.

**4) Validade de conclusão:** este aspecto refere-se à habilidade de concluir corretamente a respeito da relação entre o tratamento e os resultados do experimento, incluindo a escolha dos métodos estatísticos e a forma que o experimento foi executado. O *t-test* é o método estatístico apropriado para o estudo, pois é indicado para desenhos experimentais do tipo um fator e dois tratamentos, sendo considerado muito robusto, até para amostras que não apresentam uma distribuição normal. Uma das limitações do estudo é o tamanho muito pequeno da amostra.

Entretanto, esse é um problema conhecido em ES (WOHLIN et al., 2012), principalmente em situações em que os sujeitos selecionados precisam ser especialistas, com habilidades e competências na área de desenvolvimento de software, e quando o que se deseja observar é resultado de um trabalho em grupo.

## 5.6 Considerações finais

Este capítulo descreve os procedimentos para validação da abordagem ScrumUX, em contexto acadêmico, com alunos de graduação em Ciência da Computação da Universidade Federal de São Carlos - *campus* Sorocaba. Dezenove alunos participaram de um projeto para desenvolver aplicações de um módulo ERP para vendas no segmento de produtos perecíveis a partir de um cenário real fornecido por um profissional da indústria de software. Duas equipes seguiram a abordagem ScrumUX, enquanto as demais seguiram o Scrum tradicional.

Os resultados confirmam que a abordagem ScrumUX produziu softwares com melhor grau de usabilidade, quando considerou-se a importância dos aspectos de usabilidade atribuída pelos usuários de sistemas ERP, principalmente na perspectiva de suporte à tarefas. Ao incluir a gravidade das violações atribuída pelos inspetores de usabilidade no cálculo do índice de usabilidade, a abordagem ScrumUX também se sobressai com melhor grau de usabilidade, mas os testes estatísticos indicam que as diferenças entre os grupos experimental e de controle, neste caso, não são tão significativas.

## Conclusões e trabalhos futuros

Este trabalho propôs uma abordagem de *Design* de Interação do usuário integrada ao processo Scrum. A partir da revisão de literatura observou-se que a integração do *design* centrado no usuário em metodologias ágeis não é um assunto novo. Nos primeiros anos, logo após da publicação do Manifesto Ágil, como destaca Jurca, Hellmann e Maurer (2014), as pessoas já estavam trabalhando para descobrir como integrar as duas filosofias. Atualmente, os estudos nessa área já evoluem para fornecer modelos de maturidade em usabilidade, que buscam diagnosticar o estado de integração e avaliar a eficácia da UX em projetos ágeis, como por exemplo, as propostas de Peres et al. (2014) e de Salah, Paige e Cairns (2014a). Entretanto, observou-se que ainda existe um número bem reduzido de processos fornecendo um roteiro para o desenvolvimento do *design* de interação em métodos ágeis, bem como de *frameworks* reunindo práticas e orientando a organização do trabalho para integrar os dois times. Um número ainda menor se refere à proposta de integração em projetos Scrum.

Na etapa de concepção da abordagem ScrumUX, apresentou-se a metodologia de pesquisa SoftCoDeR aplicada para direcionar as atividades de pesquisa-ação, em estreita colaboração com profissionais de uma indústria de desenvolvimento de software na área de Sistemas ERP, com o objetivo de definir os artefatos de apoio para a abordagem proposta. Os artefatos desenvolvidos junto com os profissionais da indústria foram avaliados em projetos reais. Na sequência, a abordagem ScrumUX foi proposta e avaliada por meio de um estudo experimental envolvendo 19 alunos do 6º semestre de Bacharelado em Ciência da Computação da UFSCar - *campus* Sorocaba. Os participantes formaram cinco equipes, que foram convidadas a participar de um projeto de desenvolvimento de um módulo ERP para vendas no segmento de produtos perecíveis utilizando Scrum, sendo que, duas equipes seguiram a abordagem ScrumUX e as demais equipes seguiram o Scrum tradicional. Estabeleceu-se 2 Sprints, de 6 semanas cada, durante as quais vários dados foram coletados via ferramentas de gerenciamento de projetos e por meio de entrevistas com os membros das equipes durante o calendário de entregas e apresentações dos resultados parciais. Na fase final do projeto, as aplicações desenvolvidas pelas equipes foram avaliadas por inspeção heurística por diferentes equipes de inspetores. Posteriormente, um profissional da indústria de software na área de sistemas ERP foi convidado para fornecer sua opinião sobre a qualidade das aplicações.

Os resultados da avaliação heurística foram utilizados para calcular o índice de usabilidade

de cada aplicação nas perspectivas de apresentação e suporte à tarefas. Para tanto, a métrica  $UI_{ERP}$  foi proposta considerando que a visão analítica das inspeções de usabilidade permite abranger uma gama de situações que se enquadram dentro de potenciais problemas de usabilidade, mas que tal visão não trata necessariamente a experiência do usuário, por vezes, deixando de mensurar aquilo que realmente importa para o usuário dentro do seu contexto de interação. Assim, os índices de usabilidade foram calculados nas duas variações da métrica  $UI_{ERP}$ , que levam em conta a opinião dos usuários.

Na versão simplificada, sem incluir a gravidade das violações atribuídas pelos inspetores, os testes estatísticos atestaram uma diferença significativa entre os grupos experimentais e de controle, apontando que as equipes que utilizaram a abordagem ScrumUX produziram softwares com melhor grau de usabilidade na perspectiva de suporte à tarefas. Na segunda versão da métrica, incluindo a gravidade de violações, constatou-se que, o grupo experimental continuou apresentando um maior grau de usabilidade em relação ao grupo de controle, considerando a média geral por grupo. Contudo, os testes estatísticos revelaram que não houve diferença significativa entre o grupos, sugerindo que a utilização da abordagem ScrumUX permite o desenvolvimento de aplicações com similar grau de usabilidade da abordagem tradicional.

Acredita-se porém, que os resultados foram fortemente influenciados (i) pelo alto grau de subjetividade dos inspetores na avaliação de gravidade, uma vez que foram diferentes equipes de inspetores a avaliar cada aplicação; (ii) pela conscientização das duas equipes do grupo de controle a respeito da importância da usabilidade e com acesso às técnicas de UX, que não utilizaram a abordagem ScrumUX, mas estavam imersos na disciplina de IHC assim como as equipes do grupo experimental; e (iii) pela baixa qualidade da aplicação da equipe que não fazia parte do projeto integrado, pois as aplicações das demais equipes, que foram mais trabalhadas, ficaram sujeitas a terem um número maior de violações.

Além da análise quantitativa, pôde-se observar que a abordagem ScrumUX contribuiu consideravelmente para a qualidade de informação sobre usabilidade que foi gerada pelas equipes através dos artefatos produzidos. Os artefatos recomendados na abordagem reforçam a aplicação das técnicas personas e das heurísticas de Nielsen. Evidências comprovam que as personas foram amplamente utilizadas por ambas as equipes do grupo experimental, mantendo a equipe focada nos usuários típicos da aplicação. As heurísticas de Nielsen, por sua vez, permearam todas as fases do processo de *design* e desenvolvimento, sendo parte integrante de artefatos propostos. Tanto o Protocolo de Comunicação como as UserX Stories, permitiram que as equipes mantivessem contato direto com as principais diretrizes de usabilidade durante o desenvolvimento de suas aplicações.

Dentre as contribuições deste trabalho destaca-se a atualização da revisão da literatura sobre as estratégias de integração do DCU com métodos ágeis; a definição da métrica  $UI_{ERP}$  para inserir a visão da experiência do usuário ao método de inspeção; o desenvolvimento da aplicação web UX Designer como repositório dos artefatos *design* recomendados na abordagem ScrumUX; a abordagem de pesquisa-ação SoftCoDeR que permitiu a construção e avaliação dos artefatos de apoio, em parceria com os profissionais da indústria; e a validação da abordagem ScrumUX por meio de um estudo experimental.

## 6.1 Limitações e trabalhos futuros

Uma limitação do presente trabalho se refere ao contexto para o qual foi desenvolvido e validado. O domínio de sistemas ERP impõe muitos desafios por serem considerados sistemas muito complexos e pela variedade de usuários de diferentes setores do mercado. Apesar da relevância do estudo para o domínio de sistemas ERP, acredita-se que a abordagem pode ser adaptada para projetos Scrum de diferentes tipos de software. Os elementos que compõem a abordagem em si, tais como as práticas e os artefatos não limitam sua aplicação para outros domínios. No entanto, o sistema criado para validar a proposta precisa ser adequado, pois o mesmo considera as características do domínio e as necessidades dos público alvo. Testar a abordagem em contextos diferentes é uma sugestão para trabalhos futuros, assim como verificar se a abordagem pode ser escalada para projetos maiores, com várias equipes ágeis trabalhando co-localizadas e/ou de forma distribuída. Além disso, é preciso uma análise mais aprofundada sobre o impacto da integração do ponto de vista da agilidade, ou seja, verificar a eficácia e eficiência do Scrum com a integração do DxD. Outro ponto a ser considerado em trabalhos futuros é a realização de mais testes para evoluir a métrica de usabilidade ( $UI_{ERP}$ ) para outros domínios de aplicação, bem como amadurecer os conceitos relacionados e definir uma escala mais precisa para classificar os graus de usabilidade.

## 6.2 Publicação e submissões

As pesquisas realizadas para embasar esta dissertação possibilitaram, em colaboração com outros pesquisadores e profissionais da indústria, a publicação dos seguintes artigos:

Choma, J.; Zaina, L. A.; Amaral, A. M.; & Oliveira, P. R. (2014). Identificação das necessidades de interação dos usuários em sistemas ERP: proposta de uma metodologia investigativa. X Simpósio Brasileiro de Sistemas de Informação, 411-422.

Choma J.; Quintale D.; A. M. Zaina L. & Beraldo D. (2015). A Perspective-based Usability Inspection for ERP Systems. In Proceedings of the 17th International Conference on Enterprise Information Systems, ISBN 978-989-758-098-7, pages 57-64. DOI: 10.5220/0005346700570064.

Choma, J.; Zaina, L. A.; & Beraldo, D. (2015). Communication of Design Decisions and Usability Issues: A Protocol Based on Personas and Nielsen's Heuristics. In Human-Computer Interaction: Design and Evaluation (pp. 163-174). Springer International Publishing.

Choma, J.; Zaina, L. A.; & Silva, T. S. (2015). Towards an approach matching CMD and DSR to improve the Academia-Industry software development partnership. In Brazilian Symposium on Software Engineering (SBES), 2015. IEEE.

O artigo "Towards an approach matching CMD and DSR to improve the Academia-Industry software development partnership" foi considerado um dos cinco melhores trabalhos do SBES 2015. Por esta razão, os autores foram convidados a submeter uma versão estendida do artigo ao *Journal of Software Engineering Research and Development*<sup>1</sup>.

Além disso, este trabalho será submetido ao *Journal Information and Software Technology*<sup>2</sup>

---

<sup>1</sup><http://www.jserd.com/>

<sup>2</sup><http://www.journals.elsevier.com/information-and-software-technology/>

# Referências

- ABDALLAH, A.; HASSAN, R.; AZIM, M. A. Quantified extreme scenario based design approach. In: ACM. *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. [S.l.], 2013. p. 1117–1122.
- ABDELOUHAB, K. A. et al. Agile & user centric soa based service design framework applied in disaster management. In: *ICT-DM'2014, 1st IEEE International Conference on Information and Communication Technologies for Disaster Management*. [S.l.: s.n.], 2014.
- ABRAS, C.; MALONEY-KRICHMAR, D.; PREECE, J. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction*. Thousand Oaks: Sage Publications, v. 37, n. 4, p. 445–456, 2004.
- ADIKARI, S.; MCDONALD, C.; CAMPBELL, J. Little design up-front: a design science approach to integrating usability into agile requirements engineering. In: *Human-computer interaction. New trends*. [S.l.]: Springer, 2009. p. 549–558.
- AL-BAIK, O.; MILLER, J. Waste identification and elimination in information technology organizations. *Empirical Software Engineering*, Springer US, v. 19, n. 6, p. 2019–2061, 2014.
- ANWAR, S. et al. User-centered design practices in scrum development process: A distinctive advantage? In: *Multi-Topic Conference (INMIC), 2014 IEEE 17th International*. [S.l.: s.n.], 2014.
- ARDITO, C. et al. Investigating and promoting ux practice in industry: An experimental study. *International Journal of Human-Computer Studies*, Elsevier, v. 72, n. 6, p. 542–551, 2014.
- AVISON, D. E. et al. Action research. *Communications of the ACM*, ACM, v. 42, n. 1, p. 94–97, 1999.
- BARKSDALE, J. T.; MCCRICKARD, D. S. Concept mapping in agile usability: a case study. In: ACM. *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. [S.l.], 2010. p. 4691–4694.
- BARKSDALE, J. T.; MCCRICKARD, D. S. Software product innovation in agile usability teams: an analytical framework of social capital, network governance, and usability knowledge management. *International Journal of Agile and Extreme Software Development*, Inderscience Publishers Ltd, v. 1, n. 1, p. 52–77, 2012.

- BARKSDALE, J. T.; RAGAN, E. D.; MCCRICKARD, S. D. Easing team politics in agile usability: a concept mapping approach. In: IEEE. *Agile Conference, 2009. AGILE'09*. [S.l.], 2009. p. 19–25.
- BASILI, V. R.; ROMBACH, H. D. The tame project: Towards improvement-oriented software environments. *Software Engineering, IEEE Transactions on*, IEEE, v. 14, n. 6, p. 758–773, 1988.
- BASKERVILLE, R. What design science is not. *European Journal of Information Systems*, Nature Publishing Group, v. 17, n. 5, p. 441–443, 2008.
- BASKERVILLE, R.; PRIES-HEJE, J.; VENABLE, J. Soft design science methodology. In: ACM. *proceedings of the 4th international conference on design science research in information systems and technology*. [S.l.], 2009. p. 9.
- BASTIEN, J. C.; SCAPIN, D. L. Ergonomic criteria for the evaluation of human-computer interfaces. 1993.
- BECK, K. et al. *The agile manifesto*. 2001.
- BECK, K.; FOWLER, M. *Planning extreme programming*. [S.l.]: Addison-Wesley Professional, 2001.
- BEYER, H.; HOLTZBLATT, K.; BAKER, L. An agile customer-centered method: rapid contextual design. In: *Extreme Programming and Agile Methods-XP/Agile Universe 2004*. [S.l.]: Springer, 2004. p. 50–59.
- BIOLCHINI, J. et al. Systematic review in software engineering. *System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES*, v. 679, n. 05, p. 45, 2005.
- BLOMKVIST, J. K.; PERSSON, J.; ÅBERG, J. Communication through boundary objects in distributed agile teams. In: ACM. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. [S.l.], 2015. p. 1875–1884.
- BLOMKVIST, S. Towards a model for bridging agile development and user-centered design. In: *Human-centered software engineering—integrating usability in the software development lifecycle*. [S.l.]: Springer, 2005. p. 219–244.
- BORNOE, N.; STAGE, J. Usability engineering in the wild: How do practitioners integrate usability engineering in software development? In: *Human-Centered Software Engineering*. [S.l.]: Springer, 2014. p. 199–216.
- BORSCI, S. et al. Reviewing and extending the five-user assumption: A grounded procedure for interaction evaluation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, ACM, v. 20, n. 5, p. 29, 2013.
- BROOKE, J. Sus—a quick and dirty usability scale. *Usability evaluation in industry*, London, v. 189, n. 194, p. 4–7, 1996.
- BROSHINSKY, D.; BAKER, L. Using persona with xp at landesk software, an avocent company. In: IEEE. *Agile, 2008. AGILE'08. Conference*. [S.l.], 2008. p. 543–548.

- BROWN, J. M.; LINDGAARD, G.; BIDDLE, R. Collaborative events and shared artefacts: Agile interaction designers and developers working toward common aims. In: IEEE. *Agile Conference (AGILE), 2011*. [S.l.], 2011. p. 87–96.
- BUTT, S. M.; BUTT, S. M. Usability evaluation method for agile software development. *International Journal of Software Engineering & Computer Sciences (IJSECS)*, v. 1, p. 27–38, 2015.
- CABALLERO, L.; MORENO, A. M.; SEFFAH, A. Persona as a tool to involving human in agile methods: contributions from hci and marketing. In: *Human-Centered Software Engineering*. [S.l.]: Springer, 2014. p. 283–290.
- CHAMBERLAIN, S.; SHARP, H.; MAIDEN, N. Towards a framework for integrating agile development and user-centred design. In: *Extreme programming and agile processes in software engineering*. [S.l.]: Springer, 2006. p. 143–153.
- CHO, L. Adopting an agile culture. In: IEEE. *2009 Agile Conference*. [S.l.], 2009. p. 416–421.
- CHOMA, J. et al. A perspective-based usability inspection for erp systems. In: *ICEIS - 7th International Conference on Enterprise Information Systems*. [S.l.: s.n.], 2015.
- CHOMA, J.; ZAINA, L. A.; BERALDO, D. Communication of design decisions and usability issues: A protocol based on personas and nielsen’s heuristics. In: *Human-Computer Interaction: Design and Evaluation*. [S.l.]: Springer, 2015. p. 163–174.
- CHOMA, J.; ZAINA, L. A.; SILVA, T. S. d. Towards an approach matching cmd and dsr to improve the academia-industry software development partnership. In: *CBSOFT 2015 - Congresso Brasileiro de Software*. [S.l.: s.n.], 2015.
- CLELAND-HUANG, J.; BABAR, M. A.; MIRAKHORLI, M. An inverted classroom experience: engaging students in architectural thinking for agile projects. In: ACM. *Companion Proceedings of the 36th International Conference on Software Engineering*. [S.l.], 2014. p. 364–371.
- COHN, M. *User stories applied: For agile software development*. [S.l.]: Pearson Education Indiana, 2009.
- COLE, R. et al. Being proactive: where action research meets design research. *ICIS 2005 Proceedings*, p. 27, 2005.
- CONSTANTINE, L. Beyond user-centered design and user experience: Designing for user performance. *Cutter IT Journal*, CUTTER INFORMATION CORP., v. 17, n. 2, p. 16–25, 2004.
- COOPER, A. et al. *About Face: The essentials of interaction design*. [S.l.]: John Wiley & Sons, 2014.
- COURAGE, C.; BAXTER, K. *Understanding your users: a practical guide to user requirements: methods, tools, and techniques*. [S.l.]: Gulf Professional Publishing, 2005.



- CYBIS, W. d. A. et al. Uma abordagem ergonômica para o desenvolvimento de sistemas interativos. In: *Anais do I Workshop sobre Fatores Humanos em sistemas computacionais: compreendendo usuários, construindo interfaces*. Maringá-PR. [S.l.: s.n.], 1998. v. 1, p. 102–111.
- DETWEILER, M. Managing ucd within agile projects. *interactions*, ACM, v. 14, n. 3, p. 40–42, 2007.
- DITTRICH, Y. Doing empirical research on software development: Finding a path between understanding, intervention, and method development. *Social Thinking–Software Practice*, p. 243, 2002.
- DITTRICH, Y. et al. Cooperative method development. *Empirical Software Engineering*, Springer, v. 13, n. 3, p. 231–260, 2008.
- DÜCHTING, M.; ZIMMERMANN, D.; NEBE, K. Incorporating user centered requirement engineering into agile software development. In: *Human-computer interaction. Interaction design and usability*. [S.l.]: Springer, 2007. p. 58–67.
- DYBÅ, T.; DINGSØYR, T. Empirical studies of agile software development: A systematic review. *Information and software technology*, Elsevier, v. 50, n. 9, p. 833–859, 2008.
- DYBÅ, T.; KAMPENES, V. B.; SJØBERG, D. I. A systematic review of statistical power in software engineering experiments. *Information and Software Technology*, Elsevier, v. 48, n. 8, p. 745–755, 2006.
- FERNANDEZ, A.; INSFRAN, E.; ABRAHÃO, S. Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, Elsevier, v. 53, n. 8, p. 789–817, 2011.
- FERRE, X.; JURISTO, N.; MORENO, A. M. Which, when and how usability techniques and activities should be integrated. In: *Human-Centered Software Engineering—Integrating Usability in the Software Development Lifecycle*. [S.l.]: Springer, 2005. p. 173–200.
- FERREIRA, A. d. H. *Novo dicionário Aurélio da língua portuguesa*. [S.l.]: Editora Positivo, 2004.
- FERREIRA, J.; NOBLE, J.; BIDDLE, R. Up-front interaction design in agile development. In: *Agile processes in software engineering and extreme programming*. [S.l.]: Springer, 2007. p. 9–16.
- FERREIRA, J.; SHARP, H.; ROBINSON, H. Agile development and user experience design integration as an ongoing achievement in practice. In: IEEE. *Agile Conference (AGILE), 2012*. [S.l.], 2012. p. 11–20.
- FØLSTAD, A.; LAW, E. L.-C.; HORNBÆK, K. Outliers in usability testing: How to treat usability problems found for only one test participant? In: ACM. *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*. [S.l.], 2012. p. 257–260.

- FOX, D.; SILLITO, J.; MAURER, F. Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In: IEEE. *Agile, 2008. AGILE'08. Conference*. [S.l.], 2008. p. 63–72.
- GOSPER, J.; BINNIE, A. Ux design and agile: a natural fit? *Communications of the ACM*, v. 54, n. 1, p. 54–60, 2011.
- HEVNER, A. R. A three cycle view of design science research. *Scandinavian journal of information systems*, v. 19, n. 2, p. 4, 2007.
- HEVNER, A. R. et al. Design science in information systems research. *MIS quarterly*, Springer, v. 28, n. 1, p. 75–105, 2004.
- HOUSTON, D. X. Agility beyond software development. In: ACM. *Proceedings of the 2014 International Conference on Software and System Process*. [S.l.], 2014. p. 65–69.
- HUMAYOUN, S. R.; DUBINSKY, Y.; CATARCI, T. A three-fold integration framework to incorporate user-centered design into agile software development. *HCI (16)*, Springer, v. 6776, p. 55–64, 2011.
- HUSSAIN, Z.; SLANY, W.; HOLZINGER, A. *Current state of agile user-centered design: A survey*. [S.l.]: Springer, 2009.
- IIVARI, J.; VENABLE, J. Action research and design science research—seemingly similar but decisively dissimilar. In: *European Conference on Information Systems*. [S.l.: s.n.], 2009. v. 17, p. 1–13.
- ILLMENSEE, T.; MUFF, A. 5 users every friday: A case study in applied research. In: IEEE. *Agile Conference, 2009. AGILE'09*. [S.l.], 2009. p. 404–409.
- IQBAL, H.; KHAN, M. F. Assimilation of usability engineering and user-centered design using agile software development approach. *International Journal of Modern Education and Computer Science (IJMECS)*, v. 6, n. 10, p. 23, 2014.
- ISO/IEC, I. 13407: Human-centred design processes for interactive systems. *Geneva: ISO*, 1999.
- JOKELA, T. Guiding designers to the world of usability: determining usability requirements through teamwork. In: *Human-Centered Software Engineering—Integrating Usability in the Software Development Lifecycle*. [S.l.]: Springer, 2005. p. 127–145.
- JOKELA, T. et al. The standard of user-centered design and the standard definition of usability: analyzing iso 13407 against iso 9241-11. In: ACM. *Proceedings of the Latin American conference on Human-computer interaction*. [S.l.], 2003. p. 53–60.
- JURCA, G.; HELLMANN, T. D.; MAURER, F. Integrating agile and user-centered design: a systematic mapping and review of evaluation and validation studies of agile-ux. In: IEEE. *Agile Conference (AGILE), 2014*. [S.l.], 2014. p. 24–32.
- KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2007.

- KJELDSKOV, J.; SKOV, M. B.; STAGE, J. Instant data analysis: conducting usability evaluations in a day. In: ACM. *Proceedings of the third Nordic conference on Human-computer interaction*. [S.l.], 2004. p. 233–240.
- KROPP, E.; KOISCHWITZ, K. User-centered-design in agile re through an on-site user experience consultant. In: IEEE. *Usability and Accessibility Focused Requirements Engineering (UsARE), 2014 IEEE 2nd International Workshop on*. [S.l.], 2014. p. 9–12.
- KUNIAVSKY, M. *Observing the user experience: a practitioner's guide to user research*. [S.l.]: Morgan kaufmann, 2012.
- LÁRUSDÓTTIR, M.; CAJANDER, Å.; GULLIKSEN, J. Informal feedback rather than performance measurements—user-centred evaluation in scrum projects. *Behaviour & Information Technology*, Taylor & Francis, v. 33, n. 11, p. 1118–1135, 2014.
- LAZAR, J.; FENG, J. H.; HOCHHEISER, H. *Research methods in human-computer interaction*. [S.l.]: John Wiley & Sons, 2010.
- LEE, J. C.; MCCRICKARD, D. S. Towards extreme (ly) usable software: Exploring tensions between usability and agile software development. In: IEEE. *Agile Conference (AGILE), 2007*. [S.l.], 2007. p. 59–71.
- LESZEK, A.; COURAGE, C. The doctor is”in—using the office hours concept to make limited resources most effective. In: IEEE. *Agile, 2008. AGILE’08. Conference*. [S.l.], 2008. p. 196–201.
- LIEVESLEY, M. A.; YEE, J. S. The role of the interaction designer in an agile software development process. In: ACM. *CHI’06 Extended Abstracts on Human Factors in Computing Systems*. [S.l.], 2006. p. 1025–1030.
- LIKKANEN, L. A. et al. Lean ux: the next generation of user-centered agile development? In: ACM. *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*. [S.l.], 2014. p. 1095–1100.
- LÓPEZ-GIL, J.-M. et al. Integrating field studies in agile development to evaluate usability on context dependant mobile applications. In: ACM. *Proceedings of the XV International Conference on Human Computer Interaction*. [S.l.], 2014. p. 24.
- MAO, J.-Y. et al. The state of user-centered design practice. *Communications of the ACM*, ACM, v. 48, n. 3, p. 105–109, 2005.
- MARCH, S. T.; SMITH, G. F. Design and natural science research on information technology. *Decision support systems*, Elsevier, v. 15, n. 4, p. 251–266, 1995.
- MAYHEW, D. *The usability engineering lifecycle: a practitioner's handbbok for user interface design*. [S.l.]: San Francisco, California Morgan Kaufmann Publishers: Morgan Kaufmann Publishers, 1999.
- MEMMEL, T.; GUNDELSWEILER, F.; REITERER, H. Agile human-centered software engineering. In: BRITISH COMPUTER SOCIETY. *Proceedings of the 21st British HCI Group Annual Conference on People and Computers*. [S.l.], 2007. p. 167–175.

- MILLER, G.; WILLIAMS, L. Personas: Moving beyond role-based requirements engineering. *Microsoft and North Carolina State University*, 2006.
- MILLER, L. Case study of customer input for a successful product. In: IEEE. *null*. [S.l.], 2005. p. 225–234.
- MINITAB, I. Minitab statistical software. *Minitab Release*, v. 13, 2000.
- MORENO, A. M.; YAGÜE, A. Agile user stories enriched with usability. In: *Agile Processes in Software Engineering and Extreme Programming*. [S.l.]: Springer, 2012. p. 168–176.
- NAJAFI, M.; TOYOSHIBA, L. Two case studies of user experience design and agile development. In: IEEE. *Agile, 2008. AGILE'08. Conference*. [S.l.], 2008. p. 531–536.
- NIELSEN, J. *10 usability heuristics for user interface design*. 1995. Disponível em: <<http://www.nngroup.com/articles/ten-usability-heuristics>>.
- NIELSEN, J. *Usability 101: Introduction to usability*. 2012. Disponível em: <<http://www.nngroup.com/articles/usability-101-introduction-to-usability/>>.
- NIELSEN J.; NORMAN, D. *The Definition of User Experience*. 2013. Disponível em: <<http://www.nngroup.com/articles/definition-user-experience/>>.
- OBENDORF, H.; FINCK, M. Scenario-based usability engineering techniques in agile development processes. In: ACM. *CHI'08 Extended Abstracts on Human Factors in Computing Systems*. [S.l.], 2008. p. 2159–2166.
- PAELKE, V.; NEBE, K. Integrating agile methods for mixed reality design space exploration. In: ACM. *Proceedings of the 7th ACM conference on Designing interactive systems*. [S.l.], 2008. p. 240–249.
- PATTON, J. Hitting the target: adding interaction design to agile software development. In: ACM. *OOPSLA 2002 Practitioners Reports*. [S.l.], 2002. p. 1–ff.
- PERES, A. et al. Agileux model: towards a reference model on integrating ux in developing software using agile methodologies. In: IEEE. *Agile Conference (AGILE), 2014*. [S.l.], 2014. p. 61–63.
- PETERSEN, K. et al. Systematic mapping studies in software engineering. In: SN. *12th International Conference on Evaluation and Assessment in Software Engineering*. [S.l.], 2008. v. 17, n. 1.
- PICHLER, R. *The Product Vision*. 2009. Disponível em: <<http://www.scrumalliance.org/community/articles/2009/january/the-product-vision>>.
- PLONKA, L. et al. Ux design in agile: a dsdm case study. In: *Agile Processes in Software Engineering and Extreme Programming*. [S.l.]: Springer, 2014. p. 1–15.
- PLONKA, L. et al. Knowledge transfer in pair programming: an in-depth analysis. *International Journal of Human-Computer Studies*, Elsevier, v. 73, p. 66–78, 2015.

- PRECHELT, L.; UNGER, B. An experiment measuring the effects of personal software process (psp) training. *Software Engineering, IEEE Transactions on*, IEEE, v. 27, n. 5, p. 465–472, 2001.
- RODRÍGUEZ, P.; KUVAJA, P.; OIVO, M. Lessons learned on applying design science for bridging the collaboration gap between industry and academia in empirical software engineering. In: ACM. *Proceedings of the 2nd International Workshop on Conducting Empirical Studies in Industry*. [S.l.], 2014. p. 9–14.
- ROGERS, Y.; SHARP, H.; PREECE, J. *Design de interação: além da interação humano-computador*. [S.l.]: Porto Alegre: Bookman, 2013.
- ROSSON, M. B.; CARROLL, J. M. *Usability engineering: scenario-based development of human-computer interaction*. [S.l.]: Morgan Kaufmann, 2002.
- RUNESON, P. et al. *Case study research in software engineering: Guidelines and examples*. [S.l.]: John Wiley & Sons, 2012.
- SAFFER, D. *Designing for interaction: creating innovative applications and devices*. [S.l.]: New Riders, 2009.
- SALAH, D.; PAIGE, R.; CAIRNS, P. Integrating agile development processes and user centred design—a place for usability maturity models? In: *Human-Centered Software Engineering*. [S.l.]: Springer, 2014. p. 108–125.
- SALAH, D.; PAIGE, R.; CAIRNS, P. A practitioner perspective on integrating agile and user centred design. In: BCS. *Proceedings of the 28th International BCS Human Computer Interaction Conference on HCI 2014-Sand, Sea and Sky-Holiday HCI*. [S.l.], 2014. p. 100–109.
- SALAH, D.; PAIGE, R. F.; CAIRNS, P. A systematic literature review for agile development processes and user centred design integration. In: ACM. *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. [S.l.], 2014. p. 5.
- SANTOS, P. S. M. D.; TRAVASSOS, G. H.; ZELKOWITZ, M. V. Action research can swing the balance in experimental software engineering. *Advances in Computers*, v. 83, p. 205–276, 2011.
- SCHWABER, K. *Agile project management with Scrum*. [S.l.]: Microsoft Press, 2004.
- SCHWABER, K.; SUTHERLAND, J. The scrum guide. *Scrum Alliance*, 2011.
- SCHWARTZ, L. Agile-user experience design: Does the involvement of usability experts improve the software quality? 2014.
- SEFFAH, A.; GULLIKSEN, J.; DESMARAIS, M. C. *Human-Centered Software Engineering-Integrating Usability in the Software Development Lifecycle*. [S.l.]: Springer Science & Business Media, 2005.
- SEIN, M. et al. Action design research. Management Information Systems Research Center, 2011.

- SHADISH, W. R.; COOK, T. D.; CAMPBELL, D. T. *Experimental and quasi-experimental designs for generalized causal inference*. [S.l.]: Wadsworth Cengage learning, 2002.
- SHULL, F.; CARVER, J.; TRAVASSOS, G. H. An empirical methodology for introducing software processes. *ACM SIGSOFT Software Engineering Notes*, ACM, v. 26, n. 5, p. 288–296, 2001.
- SILVA, T. S. D. et al. User-centered design and agile methods: A systematic review. In: *AGILE*. [S.l.: s.n.], 2011. p. 77–86.
- SILVA, T. S. D. et al. Understanding the ux designer’s role within agile teams. In: *Design, User Experience, and Usability. Design Philosophy, Methods, and Tools*. [S.l.]: Springer, 2013. p. 599–609.
- SILVA, T. Silva da; SILVEIRA, M. S.; MAURER, F. Usability evaluation practices within agile development. In: IEEE. *System Sciences (HICSS), 2015 48th Hawaii International Conference on*. [S.l.], 2015. p. 5133–5142.
- SINGH, M. U-scrum: An agile methodology for promoting usability. In: IEEE. *Agile, 2008. AGILE’08. Conference*. [S.l.], 2008. p. 555–560.
- SJOBERG, D. I.; DYBA, T.; JORGENSEN, M. The future of empirical methods in software engineering research. In: IEEE COMPUTER SOCIETY. *2007 Future of Software Engineering*. [S.l.], 2007. p. 358–378.
- SOHAIB, O.; KHAN, K. Integrating usability engineering and agile software development: A literature review. In: IEEE. *Computer design and applications (ICCD), 2010 international conference on*. [S.l.], 2010. v. 2, p. V2–32.
- SOHAIB, O.; KHAN, K. Incorporating discount usability in extreme programming. *International Journal of Software Engineering and Its Applications*, v. 5, n. 1, p. 51–62, 2011.
- SOMMERVILLE, I. et al. *Engenharia de software*. [S.l.]: Addison Wesley São Paulo, 2011.
- SY, D. Adapting usability investigations for agile user-centered design. *Journal of usability Studies*, Citeseer, v. 2, n. 3, p. 112–132, 2007.
- TECHNOLOGY, I. W. *AGILEWRAP*. 2011. Disponível em: <<http://www.agilewrap.com/index.html>>.
- TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. *Introdução à engenharia de software experimental*. [S.l.]: UFRJ, 2002.
- VENABLE, J. A framework for design science research activities. In: WASHINGTON, DC. *Proceedings of the 2006 Information Resource Management Association Conference*. [S.l.], 2006. p. 21–24.
- VERSIONONE. *9th Annual State of Agile Development Survey*. 2015. Disponível em: <<https://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf>>.

- WALE-KOLADE, A.; NIELSEN, P. A. Apathy towards the integration of usability work: A case of system justification. *Interacting with Computers*, Oxford University Press, p. iwv016, 2015.
- WALE-KOLADE, A.; NIELSEN, P. A.; PÄIVÄRINTA, T. Usability work in agile systems development practice: a systematic review. In: *Building Sustainable Information Systems*. [S.l.]: Springer, 2013. p. 569–582.
- WALE-KOLADE, A. Y. Integrating usability work into a large inter-organisational agile development project: Tactics developed by usability designers. *Journal of Systems and Software*, Elsevier, v. 100, p. 54–66, 2015.
- WIERINGA, R. et al. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, Springer, v. 11, n. 1, p. 102–107, 2006.
- WIERINGA, R.; MORALI, A. Technical action research as a validation method in information systems design science. In: *Design Science Research in Information Systems. Advances in Theory and Practice*. [S.l.: s.n.], 2012. p. 220–238.
- WILLIAMS, H.; FERGUSON, A. The ucd perspective: Before and after agile. In: IEEE. *Agile Conference (AGILE), 2007*. [S.l.], 2007. p. 285–290.
- WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: ACM. *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. [S.l.], 2014. p. 38.
- WOHLIN, C. et al. *Experimentation in software engineering*. [S.l.]: Springer Science & Business Media, 2012.
- WOLKERSTORFER, P. et al. Probing an agile usability process. In: ACM. *CHI'08 Extended Abstracts on Human Factors in Computing Systems*. [S.l.], 2008. p. 2151–2158.
- ZAPATA, C. Integration of usability and agile methodologies: A systematic review. In: *Design, User Experience, and Usability: Design Discourse*. [S.l.]: Springer, 2015. p. 368–378.
- ZHANG, Z.; BASILI, V.; SHNEIDERMAN, B. Perspective-based usability inspection: An empirical validation of efficacy. *Empirical Software Engineering*, Springer, v. 4, n. 1, p. 43–69, 1999.

# Apêndices



## Revisão de Literatura

### A.1 Lista de artigos

Os códigos relacionados na última coluna correspondem às revisões de literatura listadas abaixo:

- 1 - RSL 2001-2010 por Silva, et al., 2011
- 2 - RL 2002-2009 por Sohaib e Khan, 2010
- 3 - RSL 2002-2011 por Wale-Kolade, et al., 2013
- 4 - RSL 2001-2012 por Salah, et al., 2014
- 5 - MSL 2002-2013 por Jurca, et al., 2014
- 6 - RSL 2014-2015 – atualização

Nr.	Título	Autor	Ano	Fonte
1	Case Study of Customer Input For a Successful Product.	L. Miller.	2005	1, 3, 4, 5
2	Experiences Integrating Sophisticated User Experience Design Practices into Agile Processes.	P. Hodgetts.	2005	1, 3, 4, 5
3	Finding the forest in the trees	Jeff Patton	2005	1
4	From Extreme Programming and Usability Engineering to Extreme Usability in Software Engineering Education.	A. Holzinger, M. Errath, G. Searle, B. Thurnher, and W. Slany.	2005	2, 4, 5
5	Towards a Model for Bridging Agile Development and User-Centered Design.	S. Blomkvist.	2005	3, 4
6	UCD in Agile Projects: Dream Team or Odd Couple?	P. McInerney and F. Maurer.	2005	4
7	Adding Usability Testing to an Agile Project.	G. Meszaros and J. Aston.	2006	1, 3, 4, 5
8	The Role of the Interaction Designer in an Agile Software Development Process.	M. Lievesley and J. Yee.	2006	1, 4
9	Towards a Framework for Integrating agile Development and User-Centred Design.	S. Chamberlain, H. Sharp, and N. Maiden.	2006	1, 2, 3, 4, 5
10	Adapting Usability Investigations for Agile User-Centered Design.	D. Sy.	2007	1, 2, 3, 4

11	Agile Development Iterations and UI Design.	J. Ferreira, J. Noble, and R. Biddle.	2007	1, 2, 4, 5
12	Agile Human-Centered Software Engineering.	T. Memmel, F. Gundelsweiler, and H. Reiterer.	2007	2, 3, 4
13	Incorporating User Centered Requirement Engineering into Agile Software Development.	M. Duchting, D. Zimmermann, and K. Nebe.	2007	2, 3, 4, 5
14	Interaction Designers on eXtreme Programming Teams: Two Case Studies from the Real World.	J. Ferreira, J. Noble, and R. Biddle.	2007	1, 4
15	Managing UCD within Agile Projects.	M. Detweiler.	2007	1, 3, 4, 5
16	The UCD Perspective: Before and After Agile.	H. Williams and A. Ferguson.	2007	3, 4, 5
17	Towards Extreme(ly) Usable Software: Exploring Tensions Between Usability and Agile Software Development.	J. C. Lee and S. McCrickard.	2007	1, 4, 5
18	Up-front Interaction Design in Agile Development.	J. Ferreira, J. Noble, and R. Biddle.	2007	1, 2, 3, 4, 5
19	Usability in agile software development: extending the interaction design process with personas approach	J Haikara	2007	1, 3, 5
20	Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfully Integrated in Industry.	D. Fox, J. Sillito, and F. Maurer.	2008	1, 2, 3, 4, 5
21	Agile User Centered Design: Enter the Design Studio - A Case Study.	J. Ungar and J. White.	2008	1, 2, 3, 4
22	Are you sure? Really? A contextual approach to agile user research	Evnin, J.; Pries, M.	2008	1, 3, 5
23	Integrating agile methods for mixed reality design space exploration	V Paelke	2008	1
24	Probing an agile usability process	P Wolkerstorfer, M Tscheligi, R Sefelin	2008	1, 2, 5
25	Scenario-Based Usability Engineering Techniques in Agile Development Processes.	H. Obendorf and M. Finck.	2008	1, 2, 3, 4, 5
26	Stories, Sketches, and Lists: Developers and Interaction Designers Interacting Through Artefacts	J. Brown, G. Lindgaard, and R. Biddle.	2008	4, 5
27	The design studio: Interface design for agile teams	J Ungar	2008	1, 5
28	The Doctor is "In- Using the Office Hours Concept to Make Limited Resources Most Effective	A. Leszek and C. Courage	2008	4
29	Two case studies of user experience design and agile development	Najafi, Maryam; Toyoshiba, Len	2008	1, 2, 3, 4, 5
30	U-SCRUM: An Agile Methodology for Promoting Usability	M. Singh.	2008	1, 2, 3, 4, 5
31	User-Centred Design and Agile Development: Rebuilding the Swedish National Union Catalogue.	H. Lindstrom and M. Malmsten.	2008	4
32	Using Persona with XP at LANDesk Software, an Avocent Company	D. Broschinsky and L. Baker.	2008	1, 3, 4, 5
33	5 users every Friday: A case study in applied research	Illmensee, Tom; Muff, Alyson	2009	1, 3, 5
34	Adopting an Agile Culture: A User Experience Team's Journey	L. Cho	2009	1, 3, 5

35	Current State of Agile User-Centered Design: A Survey	Hussain, Z.; Slany, W.; Holzinger, A.	2009	1
36	Easing team politics in agile usability: A concept mapping approach	Barksdale, Jeremy T.; Ragan, Eric D.; McCrickard, D. Scott	2009	1, 5
37	From cradle to sprint: Creating a full-lifecycle request pipeline at nationwide insurance	Fisher, Kevin G.; Bankston, Arlen	2009	1
38	Improving Obama Campaign Software: Learning from Users	Belchev, B. and P. Baker	2009	3
39	Little Design Up-Front: A Design Science Approach to Integrating Usability into Agile Requirements Engineering.	S. Adikari, C. McDonald, and J. Campbell.	2009	1, 3, 4, 5
40	Satisfying Stakeholders' Needs Balancing Agile and Formal Usability Test Results	Winter, J., Rönkkö, K.	2009	1
41	The Impact of Agile on User-Centered Design: Two Surveys Tell the Story.	D. Dayton and C. Barnum.	2009	4
42	The importance of identity and vision to user experience designers on agile projects	Kollmann, Johanna; Sharp, Helen; Blandford, Ann	2009	1, 2, 3, 5
43	User-Centered Design Meets Feature-Driven Development: An Integrating Approach for Developing the Web Application myPIM	T Krohn, M Kindsmiller	2009	1, 3, 5
44	When User Experience Met Agile: A Case Study.	M. Budwig, S. Jeong, and K. Kelkar.	2009	1, 3, 4, 5
45	A Usability-Pattern-Based Requirements-Analysis Method to Bridge the Gap between User Tasks and Application Features	S. Lee, I. Ko, S. Kang and D. Lee	2010	5
46	Concept Mapping in Agile Usability	J. Barksdale	2010	3, 5
47	Design and Development in the "Agile Room": Trialing Scrum at a Digital Agency.	K. Tzanidou and J. Ferreira.	2010	1, 4
48	Launchpad's Quest for a Better and Agile User Interface.	M. Albisetti.	2010	3, 4, 5
49	The Focus on Usability in Testing Practices in Industry.	M. Larusdottir, E. Bjarnadottir, and J. Gulliksen.	2010	1, 4
50	Using "Rapid Experimentation" to Inform Customer Service Experience Design	S. Meckem and J. Carlson	2010	5
51	Values and Assumptions Shaping Agile Development and User Experience Design in Practice.	J. Ferreira, H. Sharp, and H. Robinson.	2010	1, 4, 5
52	A Three-Fold Integration Framework to Incorporate User-Centered Design into Agile Software Development.	S. R. Humayoun, Y. Dubinsky, and T. Catarci.	2011	3, 4, 5
53	Collaborative Events and Shared Artefacts: Agile Interaction Designers and Developers Working Toward Common Aims.	J. Brown, G. Lindgaard, and R. Biddle.	2011	4
54	Incorporating Discount Usability in Extreme Programming	O. Sohaib and K. Khan,	2011	5
55	Meeting Organisational Needs and Quality Assurance through Balancing Agile and Formal Usability Testing Results Software Engineering Techniques	Winter, J., K. Rönkkö, et al.	2011	3

56	User Centered Design and Agile Methods: A Systematic Review	T. Silva da Silva, A. Martin, F. Maurer and M. S. Silveira	2011	3, 4, 5, 6
57	User Experience Design and Agile Development: Managing Cooperation through Articulation Work	J. Ferreira, H. Sharp, and H. Robinson.	2011	1, 3, 4
58	UX Design and Agile: A Natural Fit?	J. Gosper and A. Binnie	2011	1, 4
59	Agile Development and User Experience Design Integration as an On-going Achievement in Practice.	J. Ferreira, H. Sharp, and H. Robinson.	2012	4
60	Agile Development and UX Design: Towards Understanding Work Cultures to Support Integration	J. Ferreira	2012	5
61	Agile User Experience Development in a Large Software Organization: Good Expertise but Limited Impact	K. Kuusinen, T. Mikkonen and S. Pakarinen	2012	5
62	Agile User Stories Enriched with Usability.	A. Moreno and A. Yagie.	2012	4, 5
63	The Usage of Usability Techniques in Scrum Projects	Y. Jia, M. K. Larusdottir and Ł. Cajander	2012	5
64	Existing but Not Explicit - The User Perspective in Scrum Projects in Practice	Ł. L. M. Cajander and J. Gulliksen	2013	5
65	Informal Cognitive Walkthroughs (ICW): Paring Down and Pairing Up for an Agile World	V. Grigoreanu and M. Mohanna	2013	5
66	Quantified Extreme Scenario Based Design Approach	A. Abdallah, R. Hassan and M. Azim,	2013	5
67	Understanding the UX Designer's Role within Agile Teams	T. Silva da Silva, M. A. Silveira and C. d. O. Melo,	2013	5
68	A Practitioner Perspective on Integrating Agile and User Centred Design.	Salah, D., Paige, R., Cairns, P.	2014	6
69	Agile & user centric SOA based service design framework applied in disaster management.	Ait Abdelouhab, K., Idoughi, D., Kolski, P.	2014	6
70	Agile-User Experience Design: Does the Involvement of Usability Experts Improve the Software Quality?	Schwartz, L.	2014	6
71	An inverted classroom experience: engaging students in architectural thinking for agile projects.	Cleland-Huang, J., Babar, M. A., Mirakhorli, M.	2014	6
72	Assimilation of Usability Engineering and User-Centered Design using Agile Software Development Approach.	Iqbal, H., Khan, M. F.	2014	6
73	Communication through Boundary Objects in Distributed Agile Teams.	Blomkvist, J. K., Persson, J., Łberg, J.	2014	6
74	Creative sprints: an unplanned broad agile evaluation and redesign process.	Garnik, I., Sikorski, M., Cockton, G. (2014, October)	2014	6
75	Customer Feedback and UCD in Agile Software Development.	Stickel, O., Draxler, S., Stevens, G.	2014	6

76	Engineering m-learning using agile user-centered design	Rahim, W.A., Isa, W.M., Lokman, A.M., Taharim, N.F., Wahid, N.D.	2014	6
77	Engineering rural informatics using agile user centered esign,"Information and Communication Technology	Mohd Isa, W.A.R.W.; Mohd Lokman, A.; Syed Aris, S.R.; Abdul Aziz, M.; Taslim, J.; Manaf, M.; Sulaiman, R.	2014	6
78	How much participation is enough?: a comparison of six participatory design projects in terms of outcomes.	Whittle, J.	2014	6
79	Informal feedback rather than performance measurements-user-centred evaluation in Scrum projects.	Lárusdóttir, M., Cajander, Í., Gulliksen, J.	2014	6
80	Integrating Field Studies in Agile Development to Evaluate Usability on Context Dependand Mobile Applications.	López-Gil, J. M., Urretavizcaya, M., Losada, B., Fernández-Castro, I.	2014	6
81	Integrating usability evaluations into Scrum: A case study based on remote synchronous user testing	Lizano, F.a , Sandoval, M.M.b , Stage, J.a	2014	6
82	Integrating usability practices into agile development: A case study	Wale-Kolade, A.Y.a , Nielsen, P.A.b , Päivärinta, T.c	2014	6
83	Lean UX - The next generation of user-centered Agile development?	Liikkanen, L.A., Kilpiö, H., Svan, L., Hiltunen, M.	2014	6
84	Persona as a Tool to Involving Human in Agile Methods: Contributions from HCI and Marketing	Caballero, L.a , Moreno, A.M.b , Seffah, A.c	2014	6
85	Usability engineering in the wild: How do practitioners integrate usability engineering in software development?	Bornoe, N., Stage, J.	2014	6
86	User Feedback in Mobile Development.	Krusche, S., Bruegge, B.	2014	6
87	User-centered design practices in scrum development process: A distinctive advantage?	Anwar, S.; Motla, Y. H.; Siddiq, Y.; Asghar, S.; Shabbir, H. M.; Khan, Z. I.,	2014	6
88	User-centered-design in agile RE through an On-site User Experience Consultant.	Kropp, E., Koischwitz, K.	2014	6
89	UX design in agile: A DSDM case study	Plonka, L.a , Sharp, H.a , Gregory, P.b , 6Taylor, K.b	2014	6
90	Agile development for the creation of proper human-computer interfaces for the elderly	Williams, D.a , Wang, M.-T.b , Chang, C.-H.c , Ahamed, S.I.a , Chu, W.b	2015	6

---

91	Apathy Towards the Integration of Usability Work: A Case of System Justification.	Wale-Kolade, A., Nielsen, P. A.	2015	6
92	Integrating usability work into a large inter-organisational agile development project: Tactics developed by usability designers.	Wale-Kolade, A. Y.	2015	6
93	Usability Evaluation Method for agile software development.	Butt, S. M., Butt, S. M.	2015	6
94	Usability Evaluation Practices within Agile Development	Silva da Silva, T.; Selbach Silveira, M.; Maurer, F.	2015	6

## Estudo experimental: Cenário

### B.1 Cenário

A empresa ArrayEnterprises (fictícia) está entrando no ramo de desenvolvimento de sistemas ERP voltados à distribuição de produtos perecíveis (medicamentos, alimentos). Como se trata de um produto perecível e controlado é obrigação da distribuidora vender apenas produtos próprios para o uso, ou seja, não é permitido vender unidades do produto que já tenham passado da data de validade. Os produtos chegam ao armazém da empresa em lotes. Cada lote possui várias unidades de um único produto, todas com a mesma data de fabricação e de validade. A entrada no estoque é feita, portanto, pelas informações do lote já que necessário o controle sobre esses dados. Ao vender os produtos a abordagem FIFO (*First-In-First-Out*) é usada, essa abordagem também é chamada de fila, ou seja, o produto é vendido de modo que os “mais antigos” no estoque sejam vendidos primeiro. Em outras palavras, aquela unidade que der entrada no estoque primeiro será vendida primeiro. Porém existem algumas exceções:

- Unidades vencidas ou próximas da data de validade: o usuário possui uma área para configurar qual é o limite de dias para vender uma unidade (por exemplo, só serão vendidas unidades que podem ser consumidas nos próximos 20 dias ou mais), caso um determinado lote esteja fora desse limite então suas unidades não são comercializadas e o lote é descartado;
- Escolha manual: o usuário, ao fazer a venda, também pode escolher manualmente de qual lote as unidades serão retiradas. Ao finalizar a venda um relatório deve ser gerado onde indicam de qual lote as unidades devem ser retiradas. É importante observar que uma mesma venda pode ter unidades retiradas de lotes diferentes já que um pedido pode ter mais unidades solicitadas do que um único lote pode suprir.

A Array Enterprises está planejando a criação de quatro módulos de um ERP: o módulo de produto, o módulo de lote, módulo de compra e módulo de venda; os módulos integrados visando o auxílio computacional no controle das compras, vendas e vencimentos dos medicamentos. Contudo a empresa fez uma proposta de trabalhar incrementalmente. Considerando que já existe um sistema legado de controle de estoque por lote e produto e de controle de clientes, este será utilizado para entrada dos dados em estoque. Você terá que para uma primeira entrega desenvolver apenas o módulo de venda do produto. Para isto, deve preparar uma base de dados onde dos dados de produto e lote e do cliente sejam alimentados manualmente (vocês deverão criar as tabelas e entrar com os dados). Cada lote controla somente 1 tipo de produto (por exemplo, dipirona com validade X e fabricação Y pertence ao lote 3; dipirona com validade A e fabricação B pertence ao lote 4, etc.). Os clientes do produto podem ser do ramo de medicamentos (distribuidoras, rede de farmácias, etc.) ou do ramo alimentício (atacadistas de alimentos, hipermercados, rede de *fast-food*, etc.).

#### B.1.1 Descrição do sistema

O ERP da ArrayEnterprises contará com três submódulos: produto, lote e venda.

- O módulo de produto contará com os dados de cada produto e controlará o estoque com a ajuda do submódulo de lote, portanto, no futuro, não é permitida a entrada de material através do produto.

- O módulo do lote contém as informações específicas do lote e se comunica com o submódulo de produto gerenciando o estoque. Além disso, possui informações sobre qual a quantidade original do lote, quantas unidades estão atualmente em estoque. O lote deve conter também os dados do fornecedor (apenas nome).
- O módulo de venda possui informações sobre o cliente, data de compra e dados dos produtos comprados. Uma venda possui um ou mais produtos, o módulo deve lidar com os lotes e fazer o seu controle. O usuário pode escolher de quais lotes cada produto será retirado ou o sistema escolhe pela abordagem FIFO levando em consideração a data de validade dos lotes. Cada venda também gera um relatório onde indica de quais lotes cada produto deve ser retirado.

### **B.1.2 Requisitos do sistema**

- R1. O sistema deverá manter controle de quantidade de estoque por lote.
- R2. O sistema deverá, ao gerenciar a quantidade do lote atualizar a quantidade total de produto no estoque.
- R3. O sistema deverá gerenciar as vendas, contendo os seguintes campos em seu cadastro: nome do cliente, data da venda e dados dos produtos vendidos que contém o nome do produto, quantidade vendida e, opcionalmente, lote escolhido.

### **B.1.3 Glossário**

Gerenciar: Ações de criar, editar e excluir um registro.



# Apêndice **C**

## Estudo experimental: planilhas de inspeção

### C.1 Grupo Experimental

Avaliação Heurística										
Produto inspecionado:				Horário de início:	00h00	Horário de fim:	00h00	Gravidade		
	Descrição da violação	Heurísticas	Persona	User Story	Local da violação	Erromética	Pequeno	Grande	Quantitativa	
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										



# Apêndice D

## Estudo experimental: heurísticas PERP

Heurísticas de Nielsen	Perspectiva Apresentação "Estou vendo?" "Estou entendendo?" "A mensagem está clara?"	Perspectiva Suporte à Tarefa (navegação) "Consigo concluir a tarefa sem obstáculos?"	Orientação do que observar
H1 - Visibilidade do estado do sistema	A1		<ul style="list-style-type: none"> <li>O estado do sistema está visível e pode ser facilmente compreendido (títulos de formulários, rótulos de campos, mensagens informativas, breadcrumbs, etc.);</li> </ul>
H2 - Concordância entre o sistema e o mundo real	A2		<ul style="list-style-type: none"> <li>Informações e opções disponíveis estão estruturadas em ordem natural e lógica (distribuição dos campos, posição de botões, etc.);</li> <li>A saída e entrada são fáceis de entender e interpretar;</li> <li>As terminologias utilizadas no sistema são consistentes e estão de acordo com os conceitos do domínio;</li> </ul>
H3 - Controle e liberdade ao usuário		S3	<ul style="list-style-type: none"> <li>Permite uso de passos mínimos para se mover no sistema;</li> <li>Permite controle de localização para voltar ao ponto de partida ou sair de um estado inesperado (breadcrumbs, botão de confirmação, link);</li> </ul>
H4 - Consistência e padrões	A4		<ul style="list-style-type: none"> <li>Os termos usados são aderentes ao contexto;</li> <li>As mensagens seguem um único padrão (formatação, símbolos, cores);</li> </ul>
H5 - Prevenção de Erros	A5	S5	<ul style="list-style-type: none"> <li>O sistema pode guiar o usuário através da correta sequência de operações para completar um processo de negócio;</li> <li>Destaque dos campos obrigatórios, indicação do formato correto de entrada de dados específica (data, CEP, telefone, etc), alerta sobre formatos incorretos ou dados inconsistentes;</li> <li>As opções de ação definem claramente os estados que serão atingidos.</li> </ul>
H6 - Reconhecer ao invés de lembrar	A6		<ul style="list-style-type: none"> <li>Facilidade para reconhecer dados já fornecidos e as opções para atingir objetivo;</li> <li>Uso de palavras e termos adequados e o usuário não precisa se lembrar o que significa (tooltip);</li> <li>Permite ao usuário visualizar informações relevantes para a interação (navegação);</li> </ul>
H7 - Flexibilidade e eficiência de uso	A7	S7	<ul style="list-style-type: none"> <li>Organização da informação alfabética, lógica ou por relevância de uso (smart default);</li> <li>A disposição dos elementos de interface (links e outras formas de acesso) aumenta a eficiência de uso, minimizando o esforço de ações físicas e nas buscas visuais;</li> <li>O sistema automatiza as tarefas rotineiras e redundantes;</li> <li>Filtros de busca adequados ao número de itens e informações;</li> <li>Mecanismos de busca de informação que auxiliam a entrada de dados (ex.: busca de CEP);</li> </ul>
H8 - Projeto minimalista e estético	A8		<ul style="list-style-type: none"> <li>O conteúdo está equilibrado e não existe excesso de informação e/ou informações desnecessárias que dificultam a legibilidade, distraia a atenção e tire o foco;</li> </ul>
H9 – Reconhecimento (*), diagnóstico (*) e recuperação de erros (**)	A9(*)	S9 (**)	<ul style="list-style-type: none"> <li>Mensagens de erro visíveis, simples e de fácil entendimento;</li> <li>Mensagens auxiliam o usuário a se recuperar e mostra como acessar soluções alternativas de interação;</li> </ul>
H10 - Ajuda		S10	<ul style="list-style-type: none"> <li>Apoio às tarefas complexas.</li> </ul>

# Apêndice E

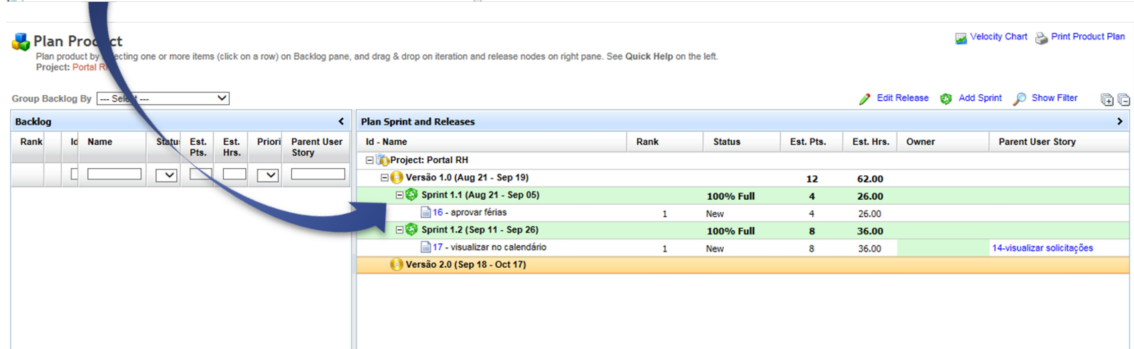
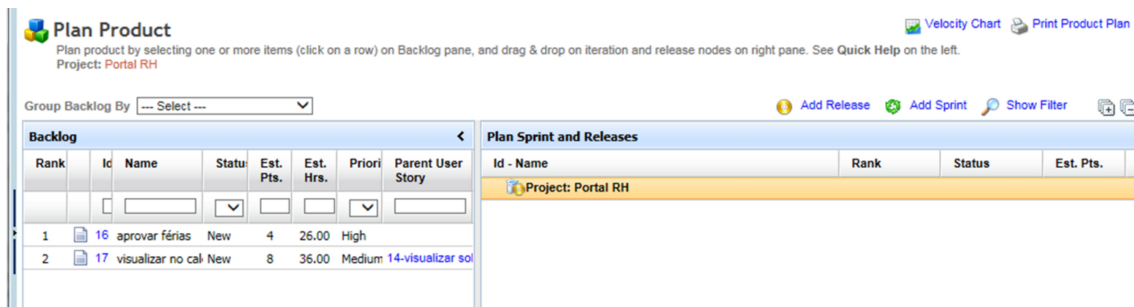
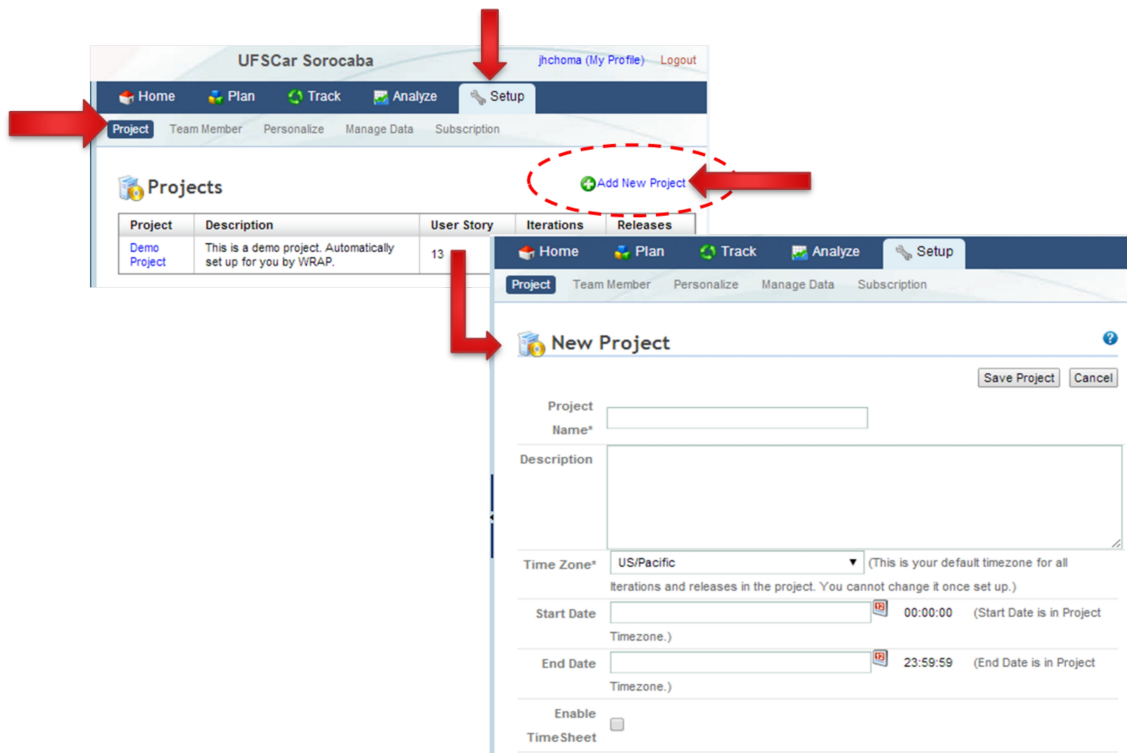
## Tutorial: Agile Wrap

### E.1 Cadastrar equipe

The screenshot shows the 'New Team Member' form in the Agile Wrap application. The form includes fields for First Name, Last Name, Notification Email, and Phone. A dropdown menu for 'Role' is open, showing options: Admin, Manager, Member, and Guest. A red arrow points to the 'Assign Projects' section, and another red arrow points to the 'Personalize' section. A callout box titled 'Papéis' (Roles) lists the roles and their permissions: Administrator (Add users), Manager (Add/Plan projects), Member (Add/Edit story, task & defect), and Guest (Track all). A red dashed circle highlights the 'Save Team Member' and 'Cancel' buttons at the bottom right, with a red box containing the text 'envia um e-mail convite' (sends an email invitation).

Papéis	Permissões
Administrator	Add users (team members).
Manager	Add/Plan projects, iterations & releases. Rank backlog.
Member	Add/Edit story, task & defect.
Guest	Track all. Add comments & attachments.

## E.2 Cadastrar e planejar projeto



### E.3 Criar user stories

**Adicionar User Story**

**User Story**  
Project: Portal RH

Save Save and Add Cancel

**Name\*** visualizar solicitações

**Description**  
Como coordenador de equipe, **eu quero** visualizar todas as solicitações de férias dos meus coordenados **para q** ue sejam analisadas e posteriormente aprovadas ou não.

Tags

**Priority** High

**Status\*** New

**Points Est.** 4

**User Story Author** jchoma

↓ ... Informações adicionais e anexos

### E.4 Criar tarefas

**Add Task**

**Task for User Story: 14- visualizar solicitações**  
Project: Portal RH

Save Save and Add Cancel

**Name\*** Lista solicitações

**Description** Listar os colaboradores com solicitações pendentes.

**Status\*** New

**Priority** High

**Owner** jchoma

**Due Date** 08/29/2014  
Clear

**Est. Hours** 8

Additional Information

## E.5 Acompanhar projeto: quadro de tarefas

The screenshot shows the Agile Wrap Taskboard interface. At the top, there are navigation tabs: Home, Plan, Track, Analyze, and Setup. Below these are sub-tabs: Iteration, Release, Taskboard, User Story, Tasks, and Defects. The main area is titled 'Taskboard' and displays a Kanban-style board with four lanes: 'User Story (13)', 'New (Tasks & Defects) (18)', 'In Progress (Tasks & Defects) (9)', and 'To Verify (Tasks) (13)'. The board is populated with various task cards, each showing a title, owner, estimated hours, and todo hours. Annotations include: 'View cards by Owner' pointing to the 'Task Owner' dropdown; 'View Tasks or Defects or both' pointing to the 'Show' checkboxes; 'Click on icons to Add Task or Add Defect' pointing to icons on a card; and a list of instructions at the bottom: 'View Story cards on left most lane', 'Task and defect cards show in right lanes as per their status.', and 'Drag and drop cards to change the status.'.

Home Plan Track Analyze Setup

Iteration Release **Taskboard** User Story Tasks Defects

Taskboard

Iteration: Iteration 2.6.1 Task Owner: ---All--- Show:  Tasks  Defects Filter Reset

User Story (13) New (Tasks & Defects) (18) In Progress (Tasks & Defects) (9) To Verify (Tasks) (13)

26500 - Develop UserStory peter with Nancy Adam Est Hrs: 0 Est Pts: 0

26500 - Rewrite Iteration- Participant page Nancy Adam Est Hrs: 0

390 - Replace ExpandAll V.Krishnan Est Hrs: 0 Todo Hrs: 0

381 - Open the Case picker in a T.Carter Est Hrs: 2 Todo Hrs: 1

362 - Display story status next V.Krishnan Est Hrs: 0 Todo Hrs: 0

379 - Search does not work prop V.Krishnan Est Hrs: 0 Todo Hrs: 2

371 - Select Trepid potential N.Adam Est Hrs: 0 Todo Hrs: 0

364 - Refresh backlog V.Mohan Est Hrs: 0 Todo Hrs: 0

388 - E-Saving a Dio up V.Mohan Est Hrs: 4 Todo Hrs: 0

408 - Give error if project is V.Mohan Est Hrs: 0 Todo Hrs: 0

508 - Add burndown to taskboard A.Gorden Est Hrs: 2 Todo Hrs: 2

425 - Show desc on hovering on ---Not Assigned--- Est Hrs: 5 Todo Hrs: 0

453 - Defect/Task Detail Add 1 T.Carter Est Hrs: 6 Todo Hrs: 0

538 - show aging fields in case N.Adam Est Hrs: 0 Todo Hrs: 0

571 - Always display ---Not Assigned--- Est Hrs: 0 Todo Hrs: 0

View Story cards on left most lane

View cards by Owner

View Tasks or Defects or both

Click on icons to Add Task or Add Defect

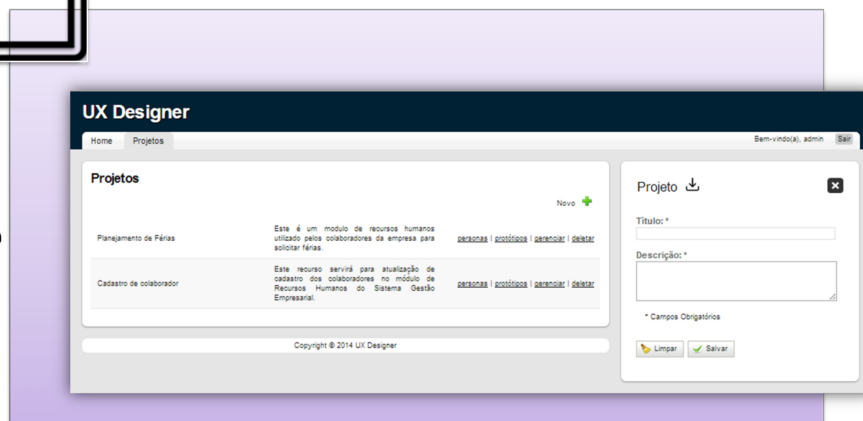
- Task and defect cards show in right lanes as per their status.
- Drag and drop cards to change the status.

## Tutorial: UX Designer

### F.1 Criar projeto



Projetos





## F.2 Inserir protótipos e criar cenários

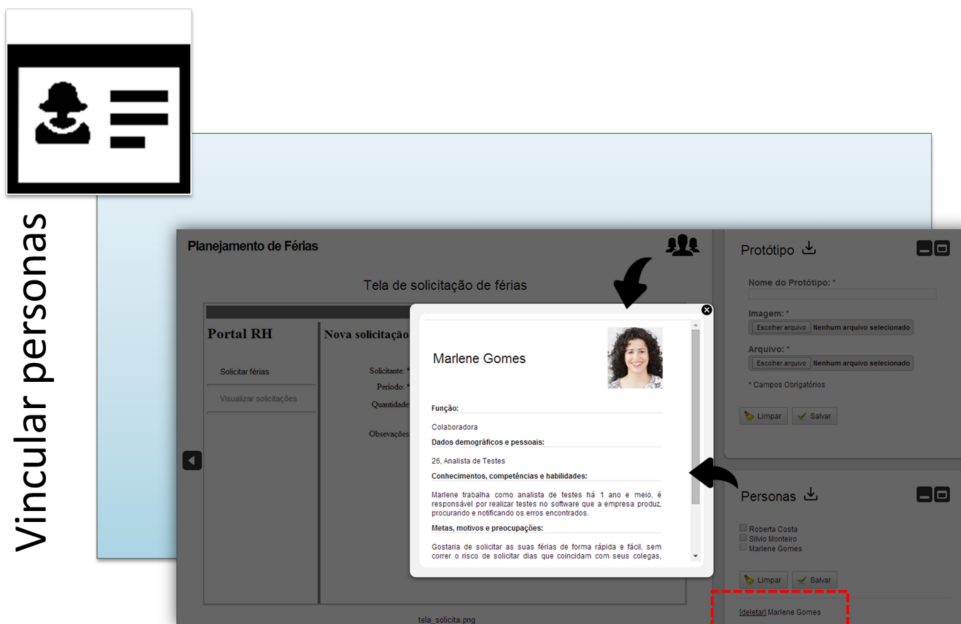
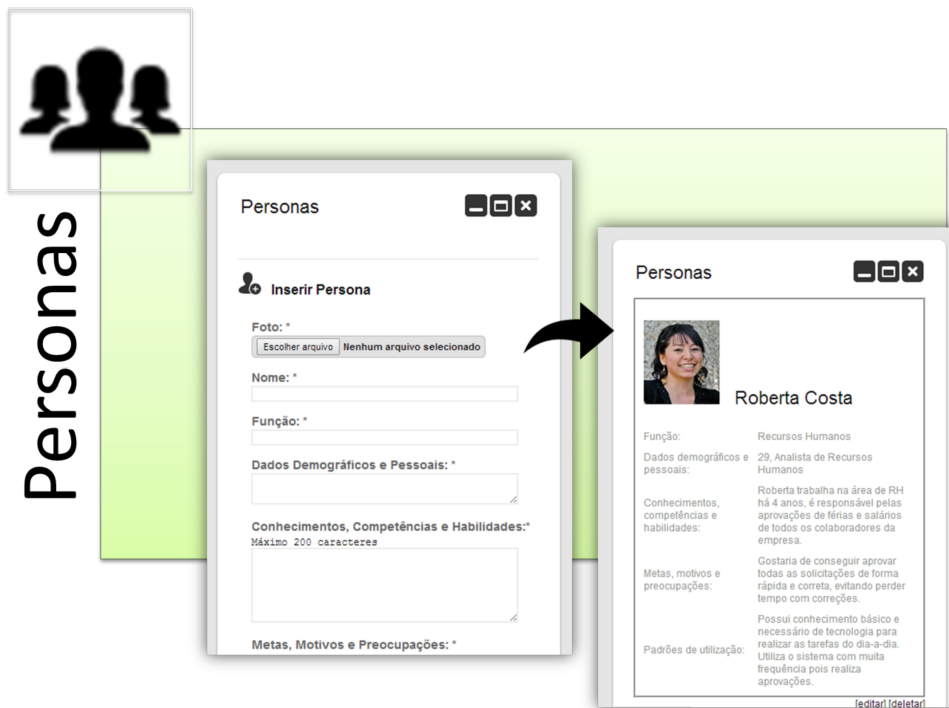
**Protótipos**

The image shows the 'Protótipos' section of the UX Designer tool. On the left, there is a large icon representing a grid of prototyping elements. The main area displays a 'Protótipo' form with the following fields: 'Nome do Protótipo:', 'Imagem:' (with 'Escolher arquivo' and 'Nenhum arquivo selecionado' options), and 'Arquivo:' (with 'Escolher arquivo' and 'Nenhum arquivo selecionado' options). Below these fields are 'Limpar' and 'Salvar' buttons. A 'Código' window is also visible, showing HTML code for a table with a border and cell spacing. The background shows a preview of a 'Portal RH' interface with a 'Nova solicitação de férias' form.

**Cenários**

The image shows the 'Cenários' section of the UX Designer tool. On the left, there is a large icon representing a list of scenarios. The main area displays a 'Cenários' form with the following fields: '+ Inserir Cenário', 'Título:' (with 'Cenário 1' entered), and 'Descrição:'. Below the description field is a rich text editor with a toolbar containing icons for undo, redo, bold, italic, underline, and text color. The description text reads: 'Marlene quer viajar com a família no mês de setembro, então, entra no sistema para fazer sua solicitação de férias no período que lhe é mais conveniente.' Below the form are 'Limpar' and 'Salvar' buttons. To the right, a 'Cenário 1' card is shown with the same text description and 'editar' and 'deletar' links.

### F.3 Criar personas



## F.4 Gerar Protocolo de Comunicação



Plano de validação

### Plano de Validação

Componente *	Heurísticas de Nielsen *	Recomendações *	Detalhes
<input type="text" value="Tag: input - Id: usuario_logado"/>	<input checked="" type="checkbox"/> H1 <input type="checkbox"/> H2 <input type="checkbox"/> H3 <input type="checkbox"/> H4 <input type="checkbox"/> H5 <input type="checkbox"/> H6 <input checked="" type="checkbox"/> H7 <input type="checkbox"/> H8 <input type="checkbox"/> H9 <input type="checkbox"/> H10 <input type="button" value="Reconhecer ao Invés de Lembrar"/>	<div style="border: 1px solid #ccc; padding: 2px;">                     Selezione uma Opção                      Selezione uma Opção                      Campo com comunicação                      Campo obrigatório                      Organização Alfabética                      Organização Smart Default                      Seleção Múltipla                      Recomendação                      Campo obrigatório - <del>usuario_logado</del> </div>	<div style="border: 1px solid #ccc; padding: 2px;">                     Atenção: Antes de salvar adicione as informações.                 </div>

\* Campos Obrigatórios



Protocolo de comunicação

### Protocolo de Comunicação

Componente	Heurísticas de Nielsen	Recomendações
<del>Tag: input   Id: usuario_logado</del>	H1 H7	Campo obrigatório - Recomendação - Este campo já deve vir preenchido com o nome do usuário logado.
<del>Tag: input   Id: de_data</del>	H5 H6 H7	Campo obrigatório - Recomendação - Não permitir data passada. Recomendação - Permitir que o usuário selecione período no calendário.
<del>Tag: input   Id: ate_data</del>	H5 H6 H7	Campo obrigatório - idem item Id: de_data
<del>Tag: input   Id: dias</del>	H1 H5	Campo obrigatório - Recomendação - Calcular número de dias.
<del>Tag: label   Id: lb_solicitante</del>	H4	Recomendação - Alinhamento à direita (idem demais rótulos)

Imprimir protocolo

## Planilhas de acompanhamento do grupos

### G.1 Projeto E1 - Experimental

Grupo 1	Sprint 1							Sprint 2						
	2/9	8/9	15/9	22/9	29/9	6/10	13/10	20/10	27/10	3/11	10/11	17/11	24/11	1/12
Cadastro na ferramenta	ok													
Histórias criadas		10	10	10	10	10	10	10	10	10	10	10	10	10
Tarefas criadas		1	1	16	20	20	20	20	25	25	25	25	25	25
Planejamento de sprint		não	sprint 1	sprint 1	sprint 1	sprint 1	sprint 1	sprint 1	sprint 2	sprint 2	sprint 2	sprint 2	sprint 2	sprint 2
Histórias - Pts estimados		60	58	58	58	58	58	58	74	74	74	74	74	74
Histórias - Hrs estimadas		1	0	56	56	56	56	56	148	163	163	163	163	163
Histórias - Prioridades		ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok
Histórias - new/defined/verify		0/9/1	0/8/1	0/6/1	0/6/1	0/6/3	0/4/1	0/2/1	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
Histórias - in progress			0	3	3	3	5	0	1	1	1	1	1	0
Histórias - Hrs todo			0	56	56	56	56	4	48	15	15	12	3	0
Tarefas - Hrs estimadas			0	91	91	91	91	89	148	163	163	163	163	163
Tarefas - Hrs todo			0	91	91	91	91	0	48	15	15	12	3	3
Tarefas - new/defined/verify			0	16/0/4	15/0/4	15/0/0	15/0/0	1/0/0	0/0/0	1/0/0	0/0/0	0/0/0	0/0/0	0/0/0
Tarefas - in progress			0	0	1	5	3	3	1	0	1	1	1	0
Tarefas - spent hrs				8	10	39	48	106	125,5	168,5	168,5	168,5	168,5	183,5
Tarefas - accepted				0	4	8	13	18	24	24	24	24	24	25

### G.2 Projeto E2 - Experimental

Grupo 2	Sprint 1							Sprint 2						
	2/9	8/9	15/9	22/9	29/9	6/10	13/10	20/10	27/10	3/11	10/11	17/11	24/11	1/12
Cadastro na ferramenta	ok													
Histórias criadas		10	10	10	10	10	10	10	10	10	10	10	10	10
Tarefas criadas		0	14	14	14	14	14	14	26	33	33	33	33	33
Planejamento de sprint		não	sprint 1	sprint 1	sprint 1	sprint 1	sprint 1	sprint 1	sprint 2	sprint 2	sprint 2	sprint 2	sprint 2	sprint 2
Histórias - Pts estimados		0	45	45	45	45	45	45	45	45	45	45	45	45
Histórias - Hrs estimadas		0	39	39	39	39	39	39	112	158	158	158	158	158
Histórias - Prioridades		0	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok
Histórias - new/defined/verify		9/0/0	0/10/0	0/10/0	0/7/0	0/6/0	0/5/0	0/4/0	0/1/0	0/0/0	0/0/0	0/0/0	0/0/0	0/0/0
Histórias - in progress			0	0	3	4	5	1	5	4	2	2	1	0
Histórias - Hrs todo			39	39	37	30	5	0	26	51	40	30	0	0
Tarefas - Hrs estimadas			81	81	81	81	81	81	112	158	158	158	158	158
Tarefas - Hrs todo			81	81	68	49	0	0	26	51	40	30	0	0
Tarefas - new/defined/verify			14/0/0	14/0/0	10/0/0	6/0/0	0/0/0	0/0/0	9/0/0	8/0/5	4/0/0	3/0/0	0/0/0	0/0/0
Tarefas - in progress			0	0	2	5	0	0	3	1	3	1	0	0
Tarefas - spent hrs				0	15	33	41	69	74	108	119	129	143	143
Tarefas - accepted				0	2	3	10	14	14	19	26	29	33	33

### G.3 Projeto E3 - Controle

Grupo 3	Sprint 1							Sprint 2						
	2/9	8/9	15/9	22/9	29/9	6/10	13/10	20/10	27/10	3/11	10/11	17/11	24/11	1/12
Cadastro na ferramenta	ok													
Histórias criadas		5	15	15	15	15	15	15	14	14	14	14	14	14
Tarefas criadas		0	22	37	40	41	36	36	58	58	58	58	59	59
Planejamento de sprint		sprint 1	sprint 1	sprint 1	sprint 1	sprint 1	sprint 1	sprint 1	sprint 2	sprint 2	sprint 2	sprint 2	sprint 2	sprint 2
Histórias - Pts estimados		0	28	37	37	37	37	37	63	63	63	63	63	63
Histórias - Hrs estimadas		0	134	158	163	175	150	150	262	262	262	262	274	274
Histórias - Prioridades		0	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok
Histórias - new/defined/verify		0/0/5	6/0/9	6/0/9	6/6/3.	6/3/0.	6/1/0.	6/0/0.	3/0/0.	1/0/0.	0/0/0.	0/0/0.	0/0/0.	0/0/0.
Histórias - in progress			0	0	3	6	8	8	2	4	5	3	1	1
Histórias - Hrs todo			134	158	163	175	155	150	97	49	13,5	9	8	2,5
Tarefas - Hrs estimadas			134	158	163	175	150	150	262	262	262	262	274	274
Tarefas - Hrs todo			134	158	163	175	155	150	97	49	13,5	9	8	2,5
Tarefas - new/defined/verify			0/0/22	37/0/0	32/0/0	23/0/0	2/0/0	0/0/0	18/0/0	13/0/0	6/0/0	4/0/0	1/0/0	0/0/0
Tarefas - in progress			0	0	3	8	2	0	4	3	1	1	1	1
Tarefas - spent hrs				0	0	62,25	127,25	133,5	148,5	195,75	234,25	238,75	251,25	256,75
Tarefas - accepted				0	3	10	32	36	36	42	51	53	57	58

### G.4 Projeto E4 - Controle

Grupo 4	Sprint 1							Sprint 2						
	2/9	8/9	15/9	22/9	29/9	6/10	13/10	20/10	27/10	3/11	10/11	17/11	24/11	1/12
Cadastro na ferramenta	ok													
Histórias criadas		13	14	15	15	15	15	15	16	16	16	16	16	16
Tarefas criadas		15	36	36	36	36	36	38	35	35	35	32	32	32
Planejamento de sprint		sprint 1	sprint 1	sprint 1	sprint 1	sprint 1	sprint 1	sprint 1	sprint 2	sprint 2	sprint 2	sprint 2	sprint 2	sprint 2
Histórias - Pts estimados		13   15	35   46	35   46	35   46	98	98	100	100	100	100	100	100	100
Histórias - Hrs estimadas		44   58	63   82	63   82	63   82	145	145	148	152	172	172	173	173	173
Histórias - Prioridades		ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok
Histórias - new/defined/verify		13/0/0	14/0/0	12/0/1	10/0/1	9/0/1	6/0/1	7/0/1	3/0/1	3/0/1	3/0/1	1/0/1	1/0/1	0/0/1
Histórias - in progress		0	0	3	4	5	8	8	12	15	15	17	17	18
Histórias - Hrs todo			0	46   82	33   82	106	106	85	66	36	24	15	5	5
Tarefas - Hrs estimadas			221	221	221	221	221	224	152	172	172	173	173	173
Tarefas - Hrs todo			217	200	187	178	160	157	62	29	24	15	5	5
Tarefas - new/defined/verify			36/0/0	33/0/0	30/0/0	29/0/0	23/0/0	25/0/0	7/0/0	7/0/0	7/0/0	1/0/0	0/0/0	0/0/0
Tarefas - in progress			0	1	3	3	3	0	3	3	0	1	1	1
Tarefas - spent hrs				16	27	36	53	61	81	134	144	154	164	164
Tarefas - accepted				2	3	4	10	13	16	25	28	30	31	31

## G.5 Projeto E5 - Controle

Grupo 5	Sprint 1							Sprint 2						
	2/9	8/9	15/9	22/9	29/9	6/10	13/10	20/10	27/10	3/11	10/11	17/11	24/11	1/12
Cadastro na ferramenta	ok													
Histórias criadas		5	5	5	5	5	5	5	5	5	5	5	5	5
Tarefas criadas		0	19	23	23	23	23	23	23	23	23	23	23	23
Planejamento de sprint		não	release	Sprint 1	Sprint 1	Sprint 1	Sprint 1	Sprint 1	Sprint 2	Sprint 2	Sprint 2	Sprint 2	Sprint 2	Sprint 2
Histórias - Pts estimados		23	26	26	26	26	26	26	26	26	26	26	26	26
Histórias - Hrs estimadas		0	114	117	117	119	122	74	122	122	122	122	122	122
Histórias - Prioridades		ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok	ok
Histórias - new/defined/verify		5	5/0/0	3/0/0	3/0/0	3/0/0	2/0/0	2/0/0	1/0/0	1/0/0	1/0/0	0/0/0	0/0/0	0/0/0
Histórias - in progress			0	2	2	2	3	0	1	1	1	1	1	0
Histórias - Hrs todo			113	113	113	109	111	111	43,5	38	29,5	17,5	8,5	0
Tarefas - Hrs estimadas			106	117	117	119	122	122	122	122	122	122	122	122
Tarefas - Hrs todo			105	113	113	109	111	111	48	38	29,5	17,5	8,5	0
Tarefas - new/defined/verify			18/0/0	18/0/1	16/0/1	14/0/1	10/0/1	10/0/1	5/0/0	5/0/0	4/0/0	1/0/0	1/0/0	0/0/0
Tarefas - in progress			1	1	2	4	4	4	3	3	3	3	3	0
Tarefas - spent hrs				11	15	24	57	69,5	69,5	79,5	87,5	98,5	105,5	113
Tarefas - accepted				3	4	4	9	15	15	15	16	19	19	23

## Estudo experimental: personas

**Arthur Camargo**

**Função:**  
Vendedor

**Dados demográficos e pessoais:**  
27 anos, Vendedor

**Conhecimentos, competências e habilidades:**  
Arthur possui curso técnico em administração e possui 9 anos de experiência em na área de vendas.

**Metas, motivos e preocupações:**  
Gostaria de poder realizar a escolha manual do lote dos produtos que devem ser vendidos. Assim como ser alertado sobre os produtos que devem ser descartados.

**Padrões de utilização:**

Figura H.1: Lean Persona (Equipe 1)

### Nicola Tesla - Gerente de estoque



Nicola Tesla tem 33 anos, trabalha há 8 anos na área de gerência, atualmente gerencia o estoque de uma filial de uma rede de mercados e utiliza um sistema de controle de estoque fornecido pela matriz que no geral atende satisfatoriamente as suas necessidades, porém gostaria de pequenas melhorias para facilitar sua vida, principalmente na parte de preenchimento de fichas que podem ser extensas e ser recorrente durante o seu trabalho.

#### Objetivos Pessoais:

- Evitar erros na geração relatórios de vendas.

#### Objetivos Práticos:

- Preenchimento automático de endereço a partir do CEP.
- Preenchimento automático do relatório a partir do CPF/CNPJ.

Figura H.2: Persona (Equipe 3)



## Estudo experimental: cenários

### Cenário

#### Venda de medicamento

Arthur é vendedor de uma farmácia e comumente recebe clientes que desejam realizar a compra de um medicamento. Quando no caixa, Arthur entra no módulo de nova venda e busca pelo cliente que deseja realizar a compra. Em seguida, ele adiciona os produtos na venda informando o nome ou código do medicamento desejado, automaticamente o lote que entrou primeiro no estoque é selecionado e Arthur pergunta se o cliente deseja trocá-lo. Quando adicionado todos os produtos desejados pelo cliente, uma nova tela de pagamento é carregada e o vendedor pergunta qual será o tipo de pagamento, o cliente escolhe entre as opções e Arthur finaliza a venda.

Figura I.1: Cenário (Equipe 1)

#### Cenário : Gerar relatório

Ator: Nicola Tesla (Gerente de estoque)

Nicola Tesla é gerente há 8 anos, diariamente precisa gerar relatórios sobre todas as vendas efetuadas, para isso quando entra no módulo de vendas, seleciona o produto, a quantidade a ser vendida e o CPF ou CNPJ do cliente, após isso clica na opção de venda e gera o relatório, onde o sistema a partir do CPF ou CNPJ irá pegar todas as informações necessárias do banco de dados para gerar o relatório completo.

Figura I.2: Cenário (Equipe 3)

# Apêndice J

## Estudo experimental: protótipos

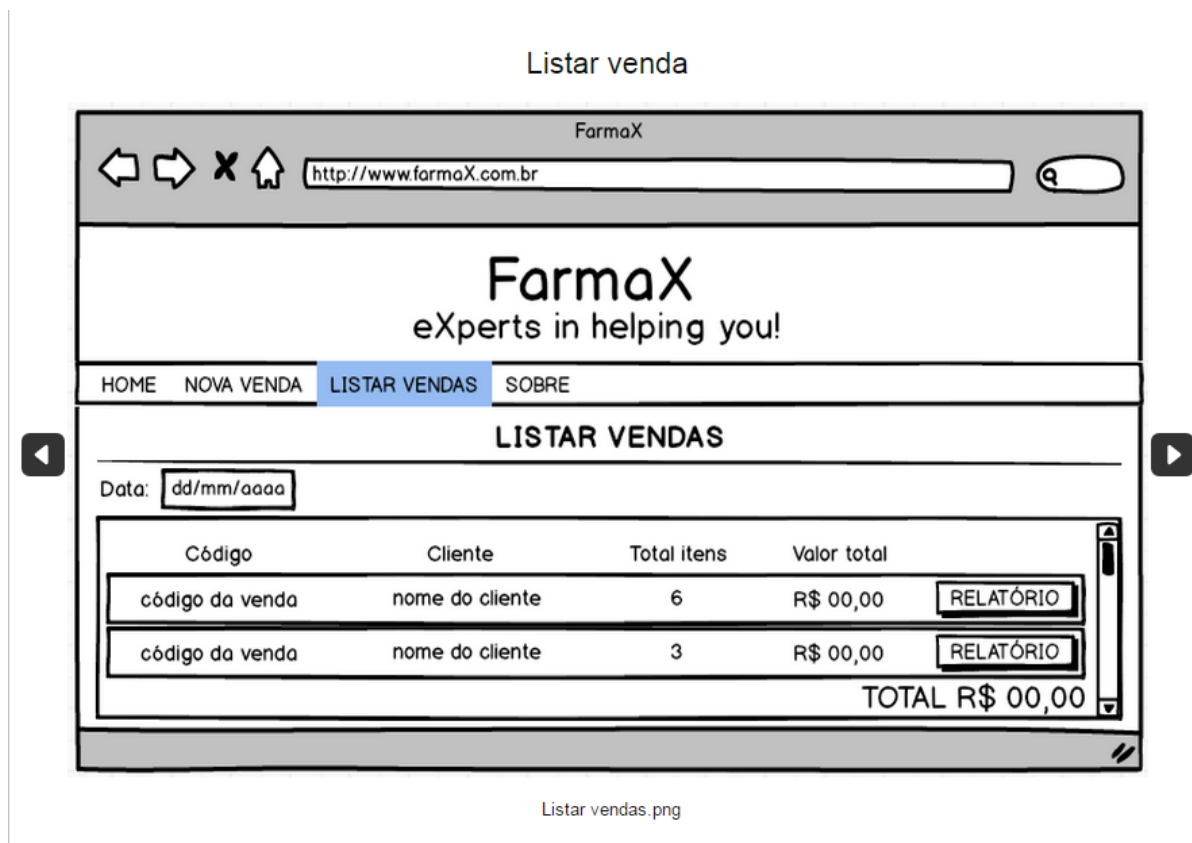


Figura J.1: Protótipo de baixa utilizando a ferramenta Balsamiq (Equipe 1)

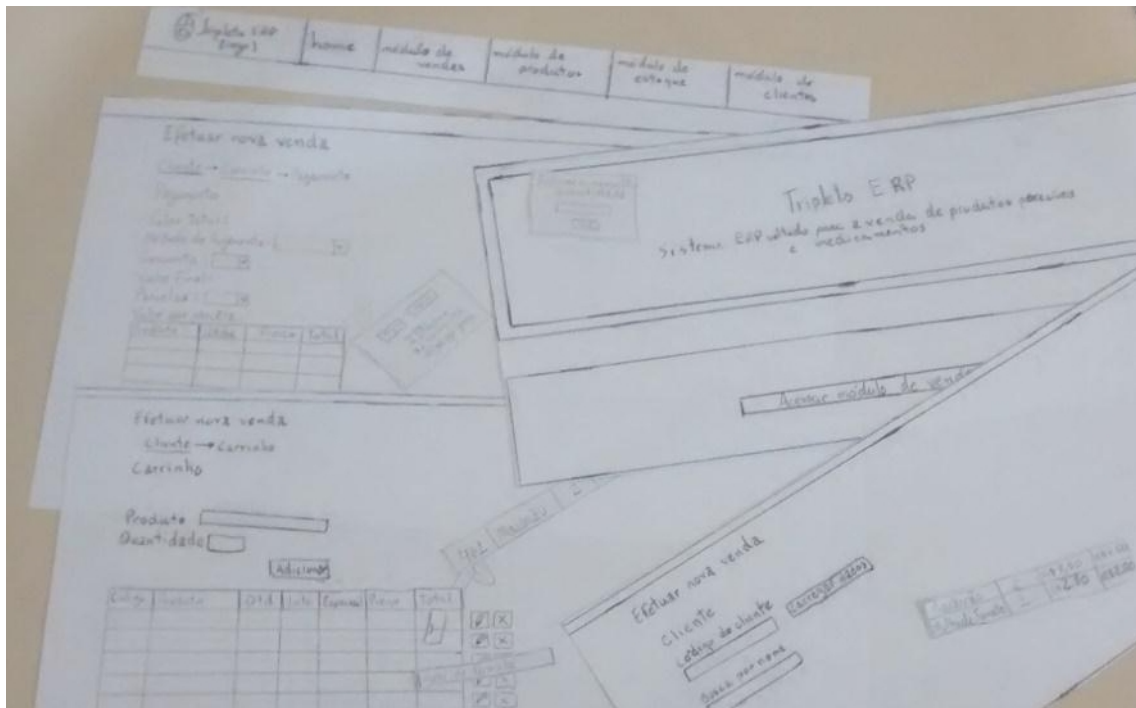


Figura J.2: Protótipo de baixa - sketches (Equipe 2)

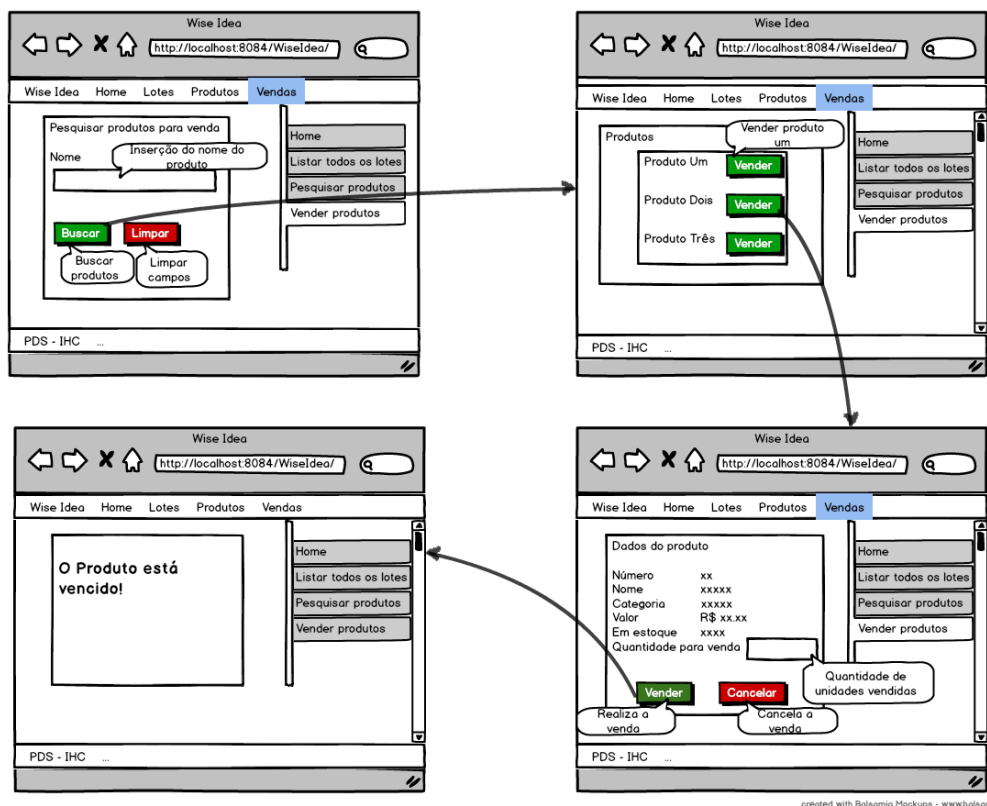


Figura J.3: Protótipos de baixa utilizando a ferramenta Balsamiq (Equipe 3)

## Estudo experimental: Protocolo de Comunicação

**Plano de validação**

---

**Tag: input | Id: autocompleteCli**  
Heurísticas: [ H1 ] [ H4 ] [ H5 ] [ H7 ] [ H9 ]

- Campo obrigatório - Uma venda só é permitida se estiver relacionada a um cliente.
- Organização Smart Default - O campo de cliente vai exibindo sugestões enquanto o usuário digita.
- Recomendação - Caso o usuário digite alguma coisa que não exista e/ou inválida, uma mensagem é exibida informando-o do ocorrido.

---

**Tag: li | Id: null**  
Heurísticas: [ H1 ] [ H4 ] [ H5 ] [ H7 ] [ H9 ]

- Campo obrigatório - Para se vender um produto, o seu código deve ser informado.
- Organização Smart Default - O campo de código vai exibindo sugestões enquanto o usuário digita.
- Recomendação - Caso o usuário digite alguma coisa que não exista e/ou inválida, uma mensagem é exibida informando-o do ocorrido.

---

**Tag: li | Id: null**  
Heurísticas: [ H1 ] [ H4 ] [ H5 ] [ H7 ] [ H9 ]

- Campo obrigatório - Para se vender um produto, o seu nome deve ser informado.
- Organização Smart Default - O campo produto vai exibindo sugestões enquanto o usuário digita.
- Recomendação - Caso o usuário digite alguma coisa que não exista e/ou inválida, uma mensagem é exibida informando-o do ocorrido.

---

**Tag: li | Id: null**  
Heurísticas: [ H1 ] [ H4 ] [ H5 ] [ H7 ]

- Campo obrigatório - Para se vender um produto, o seu lote deve ser informado.
- Organização Smart Default - O campo de lote é preenchido automaticamente de acordo com o produto informado (seja pelo seu código ou nome). A partir daí, o usuário simplesmente seleciona de qual lote deseja retirar o produto.
- Recomendação - Não é permitido entrar com um dado manualmente neste campo.

---

Figura K.1: Itens do protocolo gerados na ferramenta UX Designer (Equipe 1)

# Apêndice L

## Estudo experimental: UserX Story

 **User Story 13 – Dentro do prazo**  
Project: IHC-PDS

Name: Dentro do prazo  
Como gerente, eu quero que só seja permitida a venda de produtos que estão dentro do prazo de validade para que se venda apenas produtos próprios para o uso.

EXPANDIDA:  
Eu Viviane Oliveira, quero/preciso que só seja permitida a venda de produtos que estão dentro do prazo de validade, para isto o sistema deve realizar a venda de produtos sempre de acordo com a data

Description: de validade para que se venda apenas produtos próprios para o uso.

CRITÉRIOS DE ACEITAÇÃO:  
1. Teste veder um produto com data de validade vencido para atender A5, S5 de execução e A1, A9 de feedback,  
Heurísticas gerais: A1, A4, S7.

Tags: [prazo](#) [validade](#) [venda](#)

Priority:	High	Status:	Accepted
Points Est.:	2	User Story	Danilo Chenci
Est. Hours:	1,00	Author:	Danilo Chenci
Owner:		To Do Hours:	0,00
Release:	Versão 1,0	Parent User	Story:
Reported On:	04-09-2014 16:20:34 by Danilo Chenci (45 Days Old)	Sprint:	Sprint 1
		Last Changed On:	14-10-2014 03:29:11 by Lúgia Yoshida

**Tasks (1)**

 14 – Verificar prazo de validade

Status:	Accepted	Est. Hours:	1,00	Spent Hours:	1,00	To Do Hours:	0	Owner:	Lúgia Yoshida
---------	----------	-------------	------	--------------	------	--------------	---	--------	---------------

Figura L.1: UserX Story - US expandidas na ferramenta Agile Wrap (Equipe 1)

## Estudo experimental: histórias de usuário

Equipe 1 - Grupo experimental

Nº	UserX Story
13	<p>Eu Viviane Oliveira, quero/preciso que só seja permitida a venda de produtos que estão dentro do prazo de validade, para isto o sistema deve realizar a venda de produtos sempre de acordo com a data de validade para que se venda apenas produtos próprios para o uso.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste vender um produto com data de validade vencido para atender A5, S5 de execução e A1, A9 de <i>feedback</i>. Heurísticas gerais: A1, A4, S7.</li> </ol>
14	<p>Eu Arthur Camargo, quero/preciso poder buscar os lotes dos produtos, para isto escrevo o código do lote em um campo de busca para que facilite o descarte, a atualização e a visualização de informações dos mesmos. Avalio que meu objetivo foi atingido quando visualizo os resultados que a busca retornou.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste realizar uma busca com o campo em branco para atender A5, S5 de execução e A1, A9 de <i>feedback</i>.</li> <li>2. Teste realizar uma busca com um lote não existente para atender A1, A9 de <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4, S7.</p>
15	<p>Eu Arthur Camargo, quero/preciso poder vender um ou mais produtos, para isto eu adiciono um ou mais produtos dentro de uma mesma venda para que haja geração de lucros.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste a mensagem dos campos obrigatórios na tela de [Nova venda] para atender A5, S5 de execução e A1, A9 de <i>feedback</i>.</li> <li>2. Teste mensagem de sucesso ao efetuar pagamento na tela de [Efetuar pagamento] para atender A1 de <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4, S7.</p>
17	<p>Eu Arthur Camargo, quero/preciso que um relatório seja gerado ao final de cada venda, para isto finalizo a venda quando estiver efetuado o seu pagamento para que todas as informações relevantes dessa venda fiquem disponíveis para visualização. Avalio que meu objetivo foi atingido quando visualizo o relatório.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste geração de relatório através de uma venda efetuada para atender A1 de <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4, S7.</p>
18	<p>Eu Viviane Oliveira, eu quero/preciso que a venda dos produtos seja de acordo com a ordem de entrada no estoque, para isto o sistema deve implementar a venda FIFO para que os produtos mais antigos sejam vendidos primeiro.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste venda FIFO através da realização de uma venda para atender S7.</li> </ol> <p>Heurísticas gerais: A1, A4, S7.</p>

Nº	UserX Story
20	<p>Eu Arthur Camargo, quero/preciso poder escolher manualmente o lote do produto a ser vendido, para isto escolho o lote que desejo fazer a retirada na tela de venda para que os gostos do cliente sejam atendidos.</p> <p><u>Critérios de aceitação:</u></p> <p>1. Teste escolher o lote com o campo em branco para atender A5, S5 de execução e A1, A9 de <i>feedback</i>.</p> <p>Heurísticas gerais: A1, A4, S7.</p>
21	<p>Eu Arthur Camargo, quero/preciso ser alertado quando determinado lote estiver fora do limite de data de venda, para isto o sistema deve exibir uma mensagem de alerta para que a venda dos produtos desse lote não seja realizada.</p> <p><u>Critérios de aceitação:</u></p> <p>1. Teste mensagem de aviso através da realização de uma venda para atender A5, S5 de execução e A1, A9 de <i>feedback</i>.</p> <p>Heurísticas gerais: A1, A4, S7.</p>
22	<p>Eu Viviane Oliveira, quero/preciso poder determinar um limite de dias para se vender uma unidade próxima da data de validade, para isto busco o produto e configuro esta data limite para que ele não seja comercializado caso exceda essa quantidade de dias.</p> <p><u>Critérios de aceitação:</u></p> <p>1. Teste determinar um limite de dias com um número de dias grande o suficiente para ultrapassar a data de validade do produto para atender A5, S5 de execução e A1, A9 de <i>feedback</i>.</p> <p>Heurísticas gerais: A1, A4, S7.</p>
25	<p>Eu Arthur Camargo, quero/preciso poder buscar os produtos, para isto escrevo o nome ou o código do produto em um campo de busca para que facilite a venda e a visualização de informações dos mesmos. Avalio que meu objetivo foi atingido quando visualizo os resultados que a busca retornou.</p> <p><u>Critérios de aceitação:</u></p> <p>1. Teste realizar uma busca com o campo em branco para atender A5, S5 de execução e A1, A9 de <i>feedback</i>.</p> <p>2. Teste realizar uma busca com um produto não existente para atender A1, A9 de <i>feedback</i>.</p> <p>Heurísticas gerais: A1, A4, S7.</p>
28	<p>Eu Viviane Oliveira, quero/preciso visualizar o site através de dispositivos móveis, para isto o sistema deve ter uma interface mobile para que seja possível a busca de produtos e listagem de vendas.</p> <p><u>Critérios de aceitação:</u></p> <p>1. Teste navegação no site através de dispositivos móveis para atender A7, S7, A8 de execução.</p> <p>Heurísticas gerais: A7, S7, A8.</p>

## Equipe 2 - Grupo experimental

Nº	UserX Story
13	<p>Eu José, quero vender produtos para isto devo entrar no módulo de vendas, buscar e adicionar um cliente a venda, buscar e adicionar itens a venda, efetivar o pagamento e gerar relatório da venda através do meu <i>smartphone</i> ou navegador para que a venda seja efetivada. Avalio que meu objetivo foi atingido quando houver a confirmação da venda.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste vender produtos sem adicionar produtos para atender A5 e S7 de execução e A9 de <i>feedback</i>.</li> <li>2. Teste vender produtos sem adicionar cliente para atender A5 e S7 de execução e A9 de <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4.</p>
14	<p>Eu José, quero visualizar dados do cliente e dados dos produtos comprados. Para isto busco pelo cliente e em seguida busco pelos produtos e os adiciono ao carrinho, através do meu <i>smartphone</i> ou navegador, para que eu possa visualizar os dados da venda. Avalio que meu objetivo foi atingido quando visualizo os dados do cliente e dos produtos da venda.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste visualizar dados de clientes com cliente não cadastrados para atender A5 e S7 de execução e A9 de <i>feedback</i>.</li> <li>2. Teste visualizar dados de produtos com produtos não cadastrados para atender A5 e S7 de execução e A9 de <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4.</p>
15	<p>Eu José, quero delimitar o limite em dias para a venda de produtos próximos ao vencimento. Para isto, busco pelo produto no módulo de produtos, seleciono o produto desejado e altero o prazo de venda, pelo navegador ou <i>smartphone</i>, para evitar a venda de produtos que vencerão antes desse limite. Avalio que meu objetivo foi atingido quando o sistema não permite a venda de produtos após esse prazo.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste vender produtos após prazo estipulado para atender A5, S3 e S7 de execução e A9 e S9 de <i>feedback</i>.</li> <li>2. Teste colocar prazo com valores inválidos para atender A5, S3 e S7 de execução e A9 e S9 de <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4.</p>
17	<p>Eu José, quero vender primeiro as unidades que estão há mais tempo em estoque. Para isto, ao adicionar um produto à venda, o sistema automaticamente adiciona os itens que estão há mais tempo em estoque, pelo navegador ou <i>smartphone</i>, para evitar a perda de produtos em razão da validade. Avalio que meu objetivo foi atingido quando o sistema adiciona automaticamente os produtos corretos à venda.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste vender produtos sem especificar o lote para atender A5, S3, S5 e S7 de execução e A2, A9 e S10 de <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4.</p>
18	<p>Eu José, quero escolher o lote do produto que estou vendendo. Para isto, adiciono o produto desejado à venda e altero o lote, pelo navegador ou <i>smartphone</i>, para ter maior controle sobre a venda. Avalio que meu objetivo foi atingido quando consigo alterar o lote do produto sem ferir as restrições de prazo de venda, validade e quantidade em estoque.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste escolher outro lote para atender A5, S3 e S5 de execução e A2 <i>feedback</i>.</li> <li>2. Teste escolher lote com limite de venda vencido para atender A5, S3, S5 e S7 de execução e A9 e S9 de <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4.</p>
19	<p>Eu José, quero visualizar um relatório detalhado da venda. Para isto, após finalizar a venda, o sistema gerará esse relatório e apresentará ao usuário, pelo navegador ou <i>smartphone</i>, para ter um maior controle sobre o estoque. Avalio que meu objetivo foi atingido quando visualizo o relatório.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste finalizar a venda para atender A2 e S5 de execução e S10 <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4.</p>



Nº	UserX Story
20	<p>Eu José, quero adicionar itens à venda. Para isto devo buscar pelo item, inserir a quantidade e adicioná-lo ao carrinho através do meu <i>smartphone</i> ou navegador, para que eu consiga efetuar a venda. Avalio que meu objetivo foi atingido quando eu visualizo o item adicionado no carrinho.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste adicionar item ao carrinho com a quantidade inferior a do estoque para atender S3, A5 e S5 de execução e S9 <i>feedback</i>.</li> <li>2. Teste adicionar item ao carrinho com o campo "Especial" sendo "Sim" A2, A5, S3 e S5 de execução e A9 e S9 <i>feedback</i>.</li> <li>3. Teste adicionar item ao carrinho com a quantidade sendo um valor diferente de inteiro A2, A5 e S5 de execução e A9 e S9 <i>feedback</i>.</li> <li>4. Teste tentar adicionar nenhum item ao carrinho A5, S3 e S5 de execução e A9 e S9 <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4 e S7.</p>
21	<p>Eu José, quero buscar um produto. Para isto busco pelo nome do produto através do meu <i>smartphone</i> ou navegador para que consiga adicionar o produto na venda. Avalio que meu objetivo foi atingido quando consigo visualizar o dado do produto e consigo adicioná-lo ao carrinho.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste buscar o produto não cadastrado para atender A5, S3, S5 e S7 de execução e A9 e S10 de <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4.</p>
22	<p>Eu José, quero remover itens da venda. Para isto devo selecionar qual item desejo remover e clicar no botão referente a exclusão, através do meu <i>smartphone</i> ou navegador. Avalio que meu objetivo foi atingido quando recebo a mensagem de item removido do carrinho.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste cancelar a opção de remover o item do carrinho para atender A5, S3 e S5 de execução e A9 e S9 de <i>feedback</i>.</li> <li>2. Teste remover o item do carrinho para atender A5, S3 e S7 de execução e A9 e S9 de <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4.</p>
23	<p>Eu José, quero poder alterar a quantidade e/ou o lote de um produto previamente escolhido, para isto seleciono o produto que quero alterar a quantidade e/ou o lote através do meu <i>smartphone</i> ou navegador. Avalio que meu objetivo foi atingido quando recebo a mensagem produto alterado com sucesso.</p> <p><u>Critérios de aceitação:</u></p> <ol style="list-style-type: none"> <li>1. Teste cancelar a opção de alterar a quantidade/lote no carrinho para atender A5, S3 e S5 de execução e A9 e S9 <i>feedback</i>.</li> <li>2. Teste alterar a quantidade/lote do item no carrinho para atender A5, S3, S5 e S7 de execução e A9 e S9 de <i>feedback</i>.</li> <li>3. Teste alterar a quantidade/lote do item no carrinho com uma quantidade inferior a quantidade contida no estoque para atender A5, S3, S5 e S7 de execução e A9 e S9 de <i>feedback</i>.</li> </ol> <p>Heurísticas gerais: A1, A4.</p>

## Equipe 3 - Grupo de controle

Nº	User Story
18	Como um usuário eu posso escolher um dos lotes disponíveis para realizar uma venda.
19	Como um usuário eu posso visualizar detalhes de um produto para verificar se o mesmo está dentro do prazo de validade.
20	Como um usuário eu posso visualizar todos os lotes disponíveis para escolher os lotes provenientes de cada produto.
21	Como um usuário eu posso pesquisar produtos para visualizar as informações que necessito.
22	Como um usuário eu posso vender os produtos mais próximos do vencimento para evitar o possível desperdício de produtos.
23	Como um usuário eu posso ordenar os lotes por data de fabricação para visualizar os lotes que foram fabricados primeiros.
24	Como um usuário eu posso ordenar os lotes por validade para visualizar os lotes que vencem primeiro.
25	Como um usuário eu posso realizar uma venda para satisfazer as necessidades do meu cliente.
26	Como um usuário eu posso consultar os dados do cliente para realizar um melhor atendimento.
27	Como um usuário eu posso visualizar os relatórios gerados para obter informações mais detalhadas sobre as vendas.
28	Como um usuário eu posso escolher o lote proveniente de cada produto para realizar uma compra.
29	Como um usuário eu posso evitar a venda de produtos com data de validade próxima do vencimento para evitar o consumo de produtos vencidos pelos meus clientes.
31	Como um usuário eu posso configurar o limite de dias para vender uma unidade para deixar de fora das vendas unidades vencidas ou próximas da validade.
32	Como um usuário eu posso impedir a venda de produtos vencidos para não caracterizar uma ação ilegal.

## Equipe 4 - Grupo de controle

Nº	User Story
14	Como gerente eu quero estabelecer um limite de validade aceitável de um produto para poder determinar quais ainda podem ser vendidos.
16	Como gerente eu quero visualizar o relatório de uma venda para eu ter um maior controle das operações.
19	Como distribuidor eu quero que sempre o lote mais perto da data de validade seja colocado na venda, para que os lotes mais próximos do vencimento sejam vendidos antes.
20	Como vendedor eu quero que visualizar a quantidade do produto em estoque para saber se a venda pode ser realizada.
25	Como vendedor eu quero um cadastramento das vendas realizadas para manter uma base de dados dos produtos vendidos.
27	Como vendedor eu quero visualizar todos os produtos para que eu possa escolher o produto a ser vendido.
28	Como gerente eu quero que um relatório da venda seja gerado após sua finalização para poder verificar o lote de onde o produto deve ser retirado.
29	Como vendedor eu quero escolher manualmente um lote para realizar a venda.
30	Como vendedor eu quero que exista um formulário para que possam ser cadastradas as vendas.
32	Como vendedor eu quero colocar mais de um tipo de produto em uma venda para otimizar o processo e cadastramento da compra de um cliente.
33	Como gerente eu quero que o sistema verifique a quantidade em estoque e a requerida antes de realizar uma venda para que não sejam realizadas vendas sem que haja a quantidade em estoque.
34	Como vendedor eu quero escolher um cliente para poder adicioná-lo a venda.
35	Como vendedor eu quero poder armazenar a data de uma venda para que as vendas fiquem melhor organizadas e mais fáceis de recuperar suas informações posteriormente.
36	Como distribuidora eu quero que os dados sejam armazenados em uma base de dados para poder recuperá-los quando necessário.
37	Como vendedor eu quero que tenha um formulário para que eu possa cadastrar os dados relacionados a venda.

## Equipe 5 - Grupo de controle

Nº	User Story
13	Como product owner, desejo que o sistema guarde as informações sobre os clientes, lotes, produtos e vendas, para que seja possível obter um controle efetivo sobre os dados presentes na aplicação.
14	Como product owner, quero que os produtos sejam vendidos de acordo com a data de validade, para que os produtos com vencimento mais próximo sejam vendidos primeiro.
15	Como product owner, quero definir a quantidade de dias limite próximos a validade de um produto, os produtos só poderão ser vendidos caso o dia atual da venda não tenha ultrapassado este limite imposto até a data de validade dos produtos.
16	Como product owner, desejo a opção de poder controlar de qual lote um produto será vendido, para caso seja necessário vender produtos iguais de lotes diferentes.
17	Como product owner, quero receber um relatório no final de cada venda me indicando de qual lote as unidades dos produtos foram vendidos, bem como a quantidade de cada um.

# Apêndice **N**

## Cálculo do Índice de Usabilidade

### N.1 Índice de usabilidade: Perspectiva Apresentação

Equipe 1:

HN	Nº	Score (S)	Cosmético (Peso 1)	Pequeno (Peso 2)	Grande (Peso 3)	Catastrófico (Peso 4)	(W * S)
H1	A1.1	76,35	0	0	2	0	458.10
	A1.2	73,65	0	0	0	1	294.60
H2	A2.1	85,81	0	0	0	0	0,00
	A2.2	75,00	0	1	0	0	150,00
H4	A4.1	75,00	4	0	3	0	975,00
H5	A5.1	87,84	0	1	0	0	175,68
	A5.2	86,49	0	0	0	0	0,00
	A5.3	90,54	0	0	0	2	724,32
H6	A6.1	66,89	0	0	0	0	0,00
H7	A7.1	70,95	0	0	2	1	709,50
H8	A8.1	61,49	1	0	0	0	61,49
H9	A9.1	91,22	1	0	0	0	91,22
						$\sum (W * S) =$	3639,91
						$\sum S =$	941,23
						$UI_{\text{apres}} =$	3,87

Equipe 2:

HN	Nº	Score (S)	Cosmético (Peso 1)	Pequeno (Peso 2)	Grande (Peso 3)	Catastrófico (Peso 4)	(W * S)
H1	A1.1	76,35	0	1	0	0	152,70
	A1.2	73,65	0	3	0	2	1031,10
H2	A2.1	85,81	0	0	0	0	0,00
	A2.2	75,00	0	0	0	0	0,00
H4	A4.1	75,00	1	0	0	0	75,00
H5	A5.1	87,84	0	0	0	1	351,36
	A5.2	86,49	0	1	0	0	172,98
	A5.3	90,54	0	5	1	8	4074,30
H6	A6.1	66,89	0	0	0	0	0
H7	A7.1	70,95	0	2	0	0	283,80
H8	A8.1	61,49	0	0	0	0	0,00
H9	A9.1	91,22	0	4	0	3	1824,40
						$\sum (W * S) =$	7965,64
						$\sum S =$	941,23
						UI <sub>apres</sub> =	8,46

Equipe 3:

HN	Nº	Score (S)	Cosmético (Peso 1)	Pequeno (Peso 2)	Grande (Peso 3)	Catastrófico (Peso 4)	(W * S)
H1	A1.1	76,35	1	4	2	0	1145,25
	A1.2	73,65	0	5	3	1	1693,95
H2	A2.1	85,81	1	0	1	0	343,24
	A2.2	75,00	0	2	1	0	525,00
H4	A4.1	75,00	2	4	0	1	1050,00
H5	A5.1	87,84	0	2	0	0	351,36
	A5.2	86,49	0	0	0	0	0,00
	A5.3	90,54	0	6	2	0	1629,72
H6	A6.1	66,89	1	0	3	0	668,90
H7	A7.1	70,95	0	0	1	0	212,85
H8	A8.1	61,49	1	0	0	0	61,49
H9	A9.1	91,22	1	10	0	1	2280,50
						$\sum (W * S) =$	9962,26
						$\sum S =$	941,23
						UI <sub>apres</sub> =	10,58

Equipe 4:

HN	Nº	Score (S)	Cosmético (Peso 1)	Pequeno (Peso 2)	Grande (Peso 3)	Catastrófico (Peso 4)	(W * S)
H1	A1.1	76,35	0	0	0	0	0,00
	A1.2	73,65	2	1	0	0	294,60
H2	A2.1	85,81	0	0	0	0	0,00
	A2.2	75,00	2	4	1	0	975,00
H4	A4.1	75,00	5	2	0	0	675,00
H5	A5.1	87,84	0	3	1	0	790,56
	A5.2	86,49	0	0	0	1	345,96
	A5.3	90,54	4	5	3	1	2444,58
H6	A6.1	66,89	0	0	0	0	0,00
H7	A7.1	70,95	0	0	0	0	0,00
H8	A8.1	61,49	0	0	0	0	0,00
H9	A9.1	91,22	1	1	3	0	1094,64
						$\sum (W * S) =$	6620,34
						$\sum S =$	941,23
						UI <sub>apres</sub> =	7,03

Equipe 5:

HN	Nº	Score (S)	Cosmético (Peso 1)	Pequeno (Peso 2)	Grande (Peso 3)	Catastrófico (Peso 4)	(W * S)
H1	A1.1	76,35	0	4	5	0	1756,05
	A1.2	73,65	0	1	0	0	147,30
H2	A2.1	85,81	0	0	0	0	0,00
	A2.2	75,00	2	9	0	0	1500,00
H4	A4.1	75,00	3	5	0	0	975,00
H5	A5.1	87,84	0	0	0	0	0,00
	A5.2	86,49	0	2	0	0	345,96
	A5.3	90,54	0	0	0	4	1448,64
H6	A6.1	66,89	0	7	0	0	936,46
H7	A7.1	70,95	0	3	0	0	425,70
H8	A8.1	61,49	0	0	0	0	0,00
H9	A9.1	91,22	0	4	0	4	2189,28
						$\sum (W * S) =$	9724,39
						$\sum S =$	941,23
						UI <sub>apres</sub> =	10,33

## N.2 Índice de usabilidade: Perspectiva Suporte à Tarefas

Equipe 1:

HN	Nº	Score (S)	Cosmético (Peso 1)	Pequeno (Peso 2)	Grande (Peso 3)	Catastrófico (Peso 4)	(W * S)
H3	S3.1	67,57	0	1	4	0	945,98
H5	S5.1	72,30	0	0	1	1	506,10
	S5.2	77,03	0	0	2	1	770,30
	S5.3	73,65	0	1	2	1	883,80
H7	S7.1	62,84	0	0	0	0	0,00
	S7.2	72,97	0	1	0	0	145,94
	S7.3	71,62	0	0	1	0	214,86
H9	S9.1	73,65	0	0	2	2	1031,10
	S9.2	77,70	1	0	0	0	77,70
H10	S10.1	80,41	0	0	0	0	0,00
						$\sum (W * S) =$	4575,78
						$\sum S =$	729,74
						$UI_{\text{suporte}} =$	6,27

Equipe 2:

HN	Nº	Score (S)	Cosmético (Peso 1)	Pequeno (Peso 2)	Grande (Peso 3)	Catastrófico (Peso 4)	(W * S)
H3	S3.1	67,57	0	3	0	0	405,42
H5	S5.1	72,30	0	4	0	7	2602,80
	S5.2	77,03	0	1	1	2	1001,39
	S5.3	73,65	0	0	0	0	0,00
H7	S7.1	62,84	0	0	0	0	0,00
	S7.2	72,97	0	2	0	0	291,88
	S7.3	71,62	0	0	0	0	0,00
H9	S9.1	73,65	0	2	0	0	294,60
	S9.2	77,70	0	3	1	2	1320,90
H10	S10.1	80,41	0	0	0	0	0,00
						$\sum (W * S) =$	5916,99
						$\sum S =$	729,74
						$UI_{\text{suporte}} =$	8,10

Equipe 3:

HN	Nº	Score (S)	Cosmético (Peso 1)	Pequeno (Peso 2)	Grande (Peso 3)	Catastrófico (Peso 4)	(W * S)
H3	S3.1	67,57	0	2	2	0	675,70
H5	S5.1	72,30	0	8	2	0	1590,60
	S5.2	77,03	0	0	0	0	0,00
	S5.3	73,65	0	0	0	0	0,00
H7	S7.1	62,84	1	0	0	0	62,84
	S7.2	72,97	0	0	1	0	218,91
	S7.3	71,62	0	0	0	0	0,00
H9	S9.1	73,65	0	0	1	0	220,95
	S9.2	77,70	1	10	0	1	1942,50
H10	S10.1	80,41	0	5	0	0	804,10
						$\sum (W * S) =$	5515,60
						$\sum S =$	729,74
						$UI_{\text{suporte}} =$	7,56

Equipe 4:

HN	Nº	Score (S)	Cosmético (Peso 1)	Pequeno (Peso 2)	Grande (Peso 3)	Catastrófico (Peso 4)	(W * S)
H3	S3.1	67,57	2	3	1	0	743,27
H5	S5.1	72,30	3	6	3	2	2313,60
	S5.2	77,03	1	3	1	0	770,30
	S5.3	73,65	0	1	1	0	368,25
H7	S7.1	62,84	0	1	1	0	314,20
	S7.2	72,97	0	3	2	0	875,64
	S7.3	71,62	0	0	0	0	0,00
H9	S9.1	73,65	0	2	2	0	736,50
	S9.2	77,70	0	0	1	0	233,10
H10	S10.1	80,41	0	0	1	0	241,23
						$\sum (W * S) =$	6596,09
						$\sum S =$	729,74
						$UI_{\text{suporte}} =$	9,04



Equipe 5:

HN	Nº	Score (S)	Cosmético (Peso 1)	Pequeno (Peso 2)	Grande (Peso 3)	Catastrófico (Peso 4)	(W * S)
H3	S3.1	67,57	0	0	7	0	1418,97
H5	S5.1	72,30	0	4	0	2	1156,80
	S5.2	77,03	0	5	0	0	770,30
	S5.3	73,65	0	0	0	2	589,20
H7	S7.1	62,84	0	3	0	0	377,04
	S7.2	72,97	0	0	2	0	437,82
	S7.3	71,62	0	2	0	0	286,48
H9	S9.1	73,65	0	0	2	4	1620,30
	S9.2	77,70	0	2	2	4	2020,20
H10	S10.1	80,41	0	1	2	0	643,28
						$\sum (W * S) =$	9320,39
						$\sum S =$	729,74
						$UI_{\text{suporte}} =$	12,77