ÉRICK DE CARVALHO PEREIRA GOMES

Design de Experiência de Usuário e Desenvolvimento de Software Ágil: uma proposta do Ciclo Zero apoiada em um estudo de caso

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para conclusão do curso de MBA em Tecnologia de Software.

São Paulo

2014

ÉRICK DE CARVALHO PEREIRA GOMES

Design de Experiência de Usuário e Desenvolvimento de Software Ágil: uma proposta do Ciclo Zero apoiada em um estudo de caso

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de Software

Orientadora: Prof. Dra. Lucia Vilela Leite Filgueiras

São Paulo

2014

FICHA CATALOGRÁFICA

Gomes, Érick de Carvalho Pereira.
Design de Experiência de Usuário e Desenvolvimento de Software Ágil: uma proposta do Ciclo Zero apoiada em um estudo de caso / E. C. P. Gomes. --
São Paulo, 2014.
102p.

Monografia (MBA em Tecnologia de Software) – Escola Politécnica da Universidade de São Paulo. Programa de Educação Continuada em Engenharia.

1.Métodos Ágeis 2.Ciclo Zero 3.Experiência de Usuário 4.Usabilidade 5.Projeto Centrado no Usuário I.Universidade de São Paulo. Escola Politécnica. Programa de Educação Continuada em Engenharia II.t.

DEDICATÓRIA

Dedico este trabalho aos meus pais, pois muito contribuiram para meu crescimento pessoal e intelectual. Dedico também a minha futura noiva que me aguentou e apoiou durante todos os finais de semana firmemente dedicados aos estudos.

AGRADECIMENTOS

À Universidade de São Paulo – USP que me ofereceu toda a infraestrutura para realização das pesquisas e aos excelentes docentes.

Ao PECE – Programa de Educação Continuada em Engenharia que me deu a oportunidade de estudar um tema tão interessante ao qual pessoalmente tenho grande prazer em conhecer.

Aos meus pais que tornaram esta MBA uma realidade, me apoiando sempre na continuidade dos estudos e na sua importância para a carreira profissional.

A minha noiva e família que me aguentaram nos finais de semana dedicados aos estudos, no qual deixei de acompanhá-las diversas vezes à passeios e visitas familiares.

RESUMO

Estudos demonstram as vantagens que a aplicação de métodos ágeis tem no processo de desenvolvimento e na qualidade do software como produto final. No entanto, estes métodos não oferecem tradicionalmente técnicas adaptadas ao projeto de design de interação e experiência de usuário. O trabalho propõe o estudo de práticas de XP e UCD combinadas de forma a atender às necessidades da etapa de iniciação do projeto em processos ágeis, ou seja, durante o Ciclo Zero. O trabalho procura estudar conceitos e práticas de UCD no âmbito de processos de desenvolvimento ágil, no que diz respeito a fase de iniciação, momento este que se conquista fundos e infraestrutura para o projeto, realiza o levantamento de requisitos chave, analisa e desenvolve a estrutura arquitetural do sistema e realiza a pesquisa de usuários e de seu contexto. Como meio indispensável para a aplicação das práticas de UCD, em todas estas etapas há sempre a participação do usuário final. Os conceitos estudados são exercitados no desenvolvimento de uma aplicação Web para uma indústria têxtil, possibilitando o ganho de usabilidade, qualidade da experiência do usuário e qualidade do produto final.

Palavras-chave: métodos ágeis, experiência de usuário, processos ágeis, Ciclo Zero, usabilidade, projeto centrado no usuário.

ABSTRACT

Studies show the advantages that the benefits of applying agile methods have in the development process and the software quality. However, these methods doesn't traditionally propose interaction design and user experience adaptive techniques. The paper proposes the study of XP and UCD practices combined to address the initiation phase needs of the project in agile processes, ie, during the Zero Cycle. In this work, concepts and practices of UCD within the agile development processes are reviewed, with regard to the initiation phase, In this phase, the project is funded, its infrastructure is defined and a survey of key requirements is performed in order to develop system architectural structure, based on user research. Application of UCD practices requires user involvement in all these steps. The studied concepts are applied in the development of a web application in a textile industry environment, which resulted in improvements in usability, quality of user experience and overall final product quality.

Keywords: Agile, User Experience, Agile Processes, Zero Cycle, Usability, User-Centered Design.

LISTA DE ILUSTRAÇÕES

Pág.

Figura 1 - Ciclo de vida do software no modelo Cascata e Ágil (extraído de [SY]) ...20
Figura 2 - Método de desenvolvimento em estágios utilizado no projeto Falcon (extraído de [NAJAFI e TOYOSHIBA, 2008]) 43
Figura 3 - Ciclo Zero e design à frente da codificação (extraído de [SY]) 48
Figura 4 - Tela de digitação de pedido do sistema legado 53
Figura 5 - Fluxo do XP sem adaptação 56
Figura 6 – Inclusão de abordagens UCD no Ciclo Zero (*Iteration Zero*) da metodologia XP (Adaptado de [XP]) 57
Figura 7 - Histórias de usuários cadastradas no EasyBacklog 71
Figura 8 - Ficha de requisito da Vitrine Virtual 72
Figura 9 - Ficha de requisito do cadastro de cliente "*prospect*" 72
Figura 10 - Ficha de critérios e aceitação para requisito de Vitrine Virtual 73
Figura 11 - Modelo arquitetural proposto 75
Figura 12 - Tela da ferramenta Pencil 77
Figura 13 - Protótipo P1 da tela de Vitrine Virtual 78
Figura 14 - Protótipo P2 da tela de emissão de pedido – Cesta do Pedido 78
Figura 15 - Protótipo P3 da tela de emissão de pedido para *Smartphone* 79
Figura 16 - Protótipo P2 com observações dos participantes 81
Figura 17 - Protótipo P2 ajustado 82
Figura 18 - Leiaute da tela *Home* 84
Figura 19 - Leiaute responsivo para visualização em *Smartphones* 84
Figura 20 - Protótipo funcional da Vitrine Virtual 86
Figura 21 - Protótipo funcional da Tela de "*prospect*" 86

LISTA DE TABELAS

Pág.

Tabela 1 - Combinação de abordagens de engenharia de usabilidade e métodos ágeis (Tradução livre de [SOHAIB e KHAN]) 46
Tabela 2 - Persona da usuária Juliana / F / 29 (P2) 64
Tabela 3 - Persona da usuária Mariana / F / 33 (P4) 64

LISTA DE ABREVIATURAS E SIGLAS

AJAX	*Asynchronous Javascript and XML*
CSS	*Cascading Style Sheets*
DAD	*Disciplined Agile Delivery*
ERP	*Enterprise Resource Planning*
HTML	*HyperText Markup Language*
JSON	*JavaScript Object Notation*
LDUF	*Little Design Up Front*
ROI	*Return Over Investment*
RUP	*Rational Unified Process*
SOAP	*Simple Object Access Protocol*
UCD	*User-Centered Design*
UDDI	*Universal Description, Discovery and Integration*
UI	*User Interface*
UX	*User Experience*
XML	*Extensible Markup Language*
XP	*Extreme Programming*

SUMÁRIO

Pág.

1. Introdução 13

1.1. Motivações 13

1.2. Objetivo 14

1.3. Metodologia do Trabalho 16

1.4. Estrutura do Trabalho 17

2. Revisão Bibliográfica 19

2.1. Desenvolvimento Ágil 19

2.2. Extreme Programming (XP) 21

2.3. Ciclo Zero 24

2.4. Usabilidade 26

2.5. Experiência do Usuário (UX) 28

2.6. Design Centrado no Usuário (UCD) 29

2.7. Métodos e Práticas de UCD 29

2.7.1. Investigação Contextual 30

2.7.2. Protótipos de Alta e Baixa Fidelidade 31

2.7.3. Ordenação de Cartões (*Card Sorting*) 31

2.7.4. Avaliação Heurística 32

2.7.5. Entrevistas e Grupos Focais 33

2.7.6. Testes de Usabilidade 34

2.7.7. Checklists 35

2.7.8. Personas 35

2.7.9. Cenários 36

3. Processo de Pesquisa Literária 37

4. Análise da Aplicação de Técnicas de Usabilidade em Métodos Ágeis 39

4.1. Introdução ao Capítulo 39

4.2. A Visão do Design a Frente do Desenvolvimento 39

4.3. A Visão dos Designers UX nos Projetos Ágeis 40

4.4. A Visão de Tarefas e Esforço 41

4.5. Estudos de Caso - UX Agil 42

4.6. A Visão das Técnicas e Práticas 44

4.7. A Visão da Integração 45

4.8. A Falta da “Grande Figura” 46

4.9. A Visão das Experiências na Adaptação UCD Ágil 47
4.10. Considerações do Capítulo 49
5. Objetivos e Expectativas 50
6. A Prática de UCD no Ciclo Zero 53
6.1. Estudo de Caso – *Background* 53
6.2. A Equipe 55
6.3. Atividades Propostas – Ciclo Zero 55
6.3.1. 1º Etapa – Reunião de Planejamento 58
6.3.2. 2ª Etapa – Entrevistas para Análise de Negócio 60
6.3.3. 3ª Etapa – Investigação Contextual 65
6.3.4. 4ª Etapa – Levantamento de Histórias de Usuários 68
6.3.5. 5ª Etapa – Reunião sobre Decisão Arquitetural 74
6.3.6. 6ª Etapa – Construção de Protótipos 76
6.3.7. 7ª Etapa – Design Participativo 80
6.3.8. 8ª Etapa – Implementação da Arquitetura e UI 83
6.3.9. 9ª Etapa – Priorização do *Backlog* e Definição de Futuras Iterações 87
7. Análise dos Resultados 89
8. Considerações Finais 93
8.1. Trabalhos Futuros 95
REFERÊNCIAS 96
GLOSSÁRIO 101

1. Introdução

O Extreme Programming (XP) é uma metodologia de desenvolvimento ágil com foco na entrega de software de forma rápida e com qualidade [PATTON, 2002]. Por sua vez o User-Centered Design ou Design Centrado no Usuário (UCD) é uma abordagem multidisciplinar baseada na participação ativa do usuário durante o desenvolvimento do produto de software e é amplamente considerada como ponto crucial para a usabilidade [MAO, VREDENBURG, SMITH e CAREY, 2005].

A proposta deste trabalho é combinar os dois conceitos acima mencionados a fim de fornecer instruções para a fase de iniciação do projeto, durante o Ciclo Zero previsto pelos processos ágeis. O estudo é exercitado no desenvolvimento do *design* de interação de um sistema Web para uma indústria do ramo têxtil.

1.1. Motivações

Conforme experiências anteriores da equipe formada para o projeto do sistema Web para uma indústria têxtil, foi constatado que houveram fracassos em atender as necessidades do usuários durante suas interações com o sistema Web no que se tratava de oferecer uma interface usável. A falta de processos organizados para o acompanhamento dos projetos de software resultou também em atrasos de entregas, ausência de *feedback* do usuário e retrabalho. Desta forma, manifestou-se a necessidade de estudar as práticas de XP e UCD e como elas podem ser combinadas para contornar estas deficiências.

Percebendo a carência das metodologias de desenvolvimento ágeis quando se trata de usabilidade [BAXTER, KELKAR, MILLER, PATTON, ROSENSTEIN, VILLAMOR e FEDEROFF, 2008], propõem-se este estudo para experimentar e analisar as práticas de XP e UCD em uma organização, a fim de entregar um produto melhor com o menor custo, no menor prazo e satisfazer as expectativas dos usuários finais durante sua utilização do sistema, para que estes alcancem os resultados desejados com rapidez, satisfação e sem cometer erros.

Um dos fatores indispensáveis para se conquistar isto, é o planejamento estruturado numa iteração do processo de desenvolvimento do software ágil, proposta por alguns autores como Ciclo Zero, Iteração Zero ou Sprint Zero, já que esta é uma das etapas que muitas vezes define o sucesso ou fracasso do projeto, por seu caráter de preparação do ambiente e infraestrutura.

Conforme sugestões de alguns estudiosos do assunto, algumas características podem motivar a utilização das técnicas de UCD dentro de um cenário ágil de desenvolvimento [WILLIAMS, FERGUSON, 2007], são elas:

- Redução da documentação e do tempo gasto para desenvolvê-las;
- Foco em um escopo reduzido de funcionalidades da interface para cada ciclo de projeto;
- Quanto ao controle de mudança, a participação do usuário alvo deixa de ser dificultada a logo prazo e passa a ser mais assertiva por conta do *feedback* e rápida adaptação às mudanças de mercado por conta das curtas iterações;
- Reuniões breves e focadas;
- O *feedback* do usuário quanto a aspectos de usabilidade pode ocorrer o mais tardar no final de cada iteração de curta duração;
- As tomadas de decisão têm menos impacto para o projeto já que poderá haver futuramente mudanças ou melhorias no que foi desenvolvido.

Todos estes aspectos devem ser considerados por se tratar de um projeto composto por poucos colaboradores e com curto prazo para entrega de artefatos que resultassem em maior visibilidade para o departamento de TI, focando na constante participação do usuário final para a então entrega de um software usável e com qualidade.

1.2. Objetivo

O objetivo deste trabalho é estudar a incorporação do design centrado no usuário (UCD) na metodologia de desenvolvimento ágil XP, durante o desenvolvimento de sistemas para a plataforma Web, exercitar essa abordagem em um projeto real de uma indústria têxtil e formatar uma proposta de adaptação do Ciclo Zero, previsto pelos métodos ágeis. Os objetivos de utilizar as duas abordagens nele são:

- Conduzir o projeto com completo apoio dos usuários finais e *stakeholders*;
- Experimentá-las em um ambiente corporativo e compreender como estas podem proporcionar a melhoria da qualidade do produto final;
- Aumentar a visibilidade da equipe de analistas/programadores perante a organização onde estão inseridos com a entrega de um software usável, de

navegação intuitiva, adaptado a mudanças, otimizado e que atendem boas práticas de mercado;

- Organizar os processos de desenvolvimento a fim de reduzir eventuais atrasos em entregas e retrabalhos;
- Reduzir a necessidade de intensa documentação;
- Projetar com maior foco no design de interação;
- Oferecer novas experiências e conhecimentos à equipe.

Com a conclusão do estudo de ambas abordagens, um conjunto de atividades identificadas, analisadas e propostas foram exercitadas em um estudo de caso de uma indústria têxtil, para então formatar uma nova proposta para o Ciclo Zero, com base nas experiências e resultados obtidos conforme descritos neste trabalho. Desta forma, o processo tradicional do XP foi adaptado em suas características de iniciação ou planejamento do projeto para considerar práticas de UCD, dando enfâse em uma possível sequência de atividades e em como implementá-las.

Com intuito de alcançar a máxima participação do usuário, viabilizar o projeto, e tornar a equipe visível a toda organização, foram aplicadas técnicas para a iniciação do projeto, como a identificação do ambiente e equipe, identificação dos usuários-chave, levantamento de requisitos e formação do *Product Backlog,* desenho e avaliação de protótipos, a fim de atender as propostas dos processos ágeis bem como as abordagens de UCD. A fase de iniciação é explicada em mais detalhes no capítulo 2.3. CICLO ZERO.

Para começar de forma organizada e estruturada, antes que se iniciassem as atividades de codificação e para entender como o usuário seria colocado como ativo participante do projeto, houve a necessidade de aplicar algumas práticas de UCD na etapa proposta pelas metodologias de desenvolvimento ágil, citada frequentemente como Sprint Zero por estudiosos da metodologia ágil SCRUM.

A equipe de desenvolvedores está alocada dentro da companhia que por sua vez é o cliente, e onde estão concentrados grande parte dos *stakeholders* e usuários do produto final analisados por este trabalho. A curta distância entre os membros da equipe, o cliente e usuário final também é um importante fator que motivou este estudo de caso, por seu caráter de rápido *feedback* entre os participantes como um todo.

Vale salientar um grande desafio para a equipe no que diz respeito ao desenvolvimento do sistema analisado neste trabalho. A equipe é composta unicamente por profissionais em análise de sistemas e programadores com habilidades em design de interface, e também formada por poucos integrantes. Não há especialistas na aplicação de práticas do UCD.

Com o ideal de futuros profissionais utilizarem o conhecimento apresentado neste trabalho para o benefício pessoal e da organização onde estão inseridos, este trabalho propõe uma análise das práticas de UCD por diversos autores, identificação daquelas que se adequam à etapa de iniciação do projeto, a combinação destas com as práticas e métodos de XP e o estudo de caso destas combinações em um projeto real.

O trabalho tem como bases conceituais os estudos do UCD e de como ele pode ser aplicado em uma etapa de projeto de uma organização que consome sistemas computacionais, o aprofundamento dos conhecimentos da metodologia de desenvolvimento ágil XP e o estudo das técnicas de avaliação de usabilidade e experiência do usuário.

Como o foco é analisar a combinação das práticas de métodos ágeis e UCD, será adotada a nomenclatura Ciclo Zero quando tratar da fase de iniciação do projeto, pois o termo é frequentemente adotado pelas equipes de desenvolvimento ágil.

1.3. Metodologia do Trabalho

Inicialmente o trabalho apoiou-se na pesquisa dos assuntos relacionados aos métodos ágeis e práticas de UCD aplicadas no Ciclo Zero, o capítulo 3. PROCESSO DE PESQUISA LITERÁRIA apresenta o processo de pesquisa acerca dos assuntos mencionados no capítulo anterior, a fim de localizar o máximo de artigos, teses e livros sobre os assuntos UCD, XP e Ciclo Zero. Em seguida estas abordagens foram observadas do ponto de vista de diversos autores. Diversas práticas relacionadas ao contexto de XP e UCD foram estudadas e priorizadas em conjunto com as práticas extraídas dos trabalhos de referência, para compor o processo de desenvolvimento do projeto aqui proposto, neste caso, do sistema corporativo para atender as necessidades da Indústria Têxtil.

Durante as fases de projeto foram identificados os perfis dos usuários-alvo, compreensão de suas capacidades técnicas, análise de suas tarefas (fluxo de trabalho) dentro do atual sistema de emissão de pedidos, levantamento e priorização de todas as funcionalidades a serem implementadas no novo sistema, design e construção da arquitetura e prototipação de baixa a alta fidelidade.

Em seguida, a análise crítica das experiências obtidas com a aplicação destas abordagens foram documentadas no capítulo 7. ANÁLISE DE RESULTADOS. Neste momento, foram comparadas as espectativas com os resultados atingidos.

1.4. Estrutura do Trabalho

O Capítulo 1 INTRODUÇÃO apresenta características motivacionais que levaram a escolher as abordagens de XP e UCD dentro do contexto da fase de iniciação do projeto em um estudo de caso.

O Capítulo 2 REVISÃO BIBLIOGRÁFICA apresenta o resumo do estudo realizado sobre metodologias de desenvolvimento ágil, bem como dos métodos e práticas de design centrado no usuário, com o objetivo de apresentar de forma clara os conceitos envolvidos nesta pesquisa.

O Capítulo 3 PROCESSO DE PESQUISA LITERÁRIA apresenta os mecanismos e características das pesquisas realizadas para se alcançar o máximo de artigos, livros e testes acerca do assunto aqui estudado, mais em específico para o Ciclo Zero, XP e UCD.

O Capítulo 4 TRABALHOS RELACIONADOS apresenta a análise sistemática das práticas aplicadas no mercado de trabalho, estudadas por diversos autores, quando se refere a integração entre métodos de desenvolvimento ágil e métodos de design centrado no usuário, aplicados a fase de iniciação do projeto.

O Capítulo 5 OBJETIVOS E EXPECTATIVAS apresenta uma introdução ao estudo de caso, uma melhor compreensão do problema e mais detalhes sobre os aspectos que motivaram a equipe a escolher o processo de desenvolvimento ágil combinado ao UCD.

O Capítulo 6 A PRÁTICA DE UCD E XP NO CICLO ZERO apresenta o estudo empírico das abordagens em um projeto real de uma indústria do ramo têxtil, de todas as possíveis práticas previstas e analisadas para o Ciclo Zero. Este capítulo, desta forma, formata uma proposta para o Ciclo Zero.

O Capítulo 7 ANÁLISE DOS RESULTADOS descreve como as abordagens renderam resultados satisfatórios ou não para o projeto real. Descreve como foi a integração entre as práticas e métodos, como a equipe se adaptou a estas mudanças, como o usuário participou ativamente do projeto, o que estes acharam das funcionalidades desenvolvidas na iteração um e como foi o reconhecimento da organização sobre os resultados obtidos.

O Capítulo 8 CONSIDERAÇÕES FINAIS descreve considerações sobre os resultados obtidos e propostas para trabalhos futuros.

REFERÊNCIAS relaciona todos os artigos e livros utilizados neste trabalho.

GLOSSÁRIO relaciona e descreve sucintamente termos técnicos utilizados neste trabalho.

2. Revisão Bibliográfica

Os temas necessários para o entendimento do projeto foram objeto de estudo da literatura e são apresentados neste capítulo. Todos os assuntos aqui mencionados têm papel fomentador no estudo do Ciclo Zero e na característica ágil do UCD.

2.1. Desenvolvimento Ágil

O sucesso da organização frequentemente é negligenciado pela equipe de software em favor do sucesso técnico e pessoal. A alta diretoria não se preocupa se a solução é elegante, fácil de manter ou intuitiva para os usuários, ela quer apenas resultados. Quando o gerente não está satisfeito com o resultado do time, um dos primeiros cortes é o custo. O processo ágil foca em alcançar o sucesso técnico, pessoal bem como o organizacional e auxilia no sucesso com a entrega de valor e redução de custos. Se o projeto não caminhar em direção ao sucesso, a organização descobrirá isto cedo o suficiente para interrompê-lo. Projetos ágeis entregam primeiramente as funcionalidades de maior valor, a equipe rapidamente se adapta a mudanças, desenvolve com o menor número de *bugs* possíveis, são rápidos e com precisão e dão progresso ao projeto mesmo que os indivíduos-chave não estejam disponíveis [SHORE e WARDEN, 2008].

Os métodos ágeis são caracterizados pelo desenvolvimento iterativo e incremental e promovem a entrega frequente do software funcional, de acordo com priorizações oriundas de consultas realizadas a partir da participação ativa dos clientes, com objetivo de agregar valor ao negócio. Os métodos mais comuns são SCRUM que abrange mais a gestão do projeto e XP que foca nas práticas de desenvolvimento [HODA, KRUCHTEN, NOBLE e MARSHALL, 2010]

SCRUM é uma metodologia ágil de gestão de projetos, defende a utilização somente de documentação necessária para apoiar o desenvolvimento. Nele, o ciclo de vida do desenvolvimento é subdividido em pequenos ciclos de 15 a 20 dias de duração, chamados de *Sprints*. Uma importante característica desta metodologia é encarar a mudança como parte natural do processo [MACHADO e MEDINA].

No tradicional modelo Cascata ou *Waterfall*, o cliente só recebe o produto pronto para uso e dá o *feedback* quando o mesmo é entregue, e isto ocorre apenas no final do projeto. No modelo Cascata o processo é subdividido em Análise de

Requisitos, Projeto, Codificação e Teste, e um estágio não se inicia até que o último estágio esteja completo [SOHAIB e KHAN].

No modelo ágil, é adotado um estilo iterativo e incremental que é a subdivisão do projeto em conjuntos de funcionalidades. As entregas são feitas em pequenos *releases* incrementais e é executado um ciclo de vida do software completo em cada iteração, para um subconjunto de requisitos [SANTOS, 2012].

A figura 1 ilustra a comparação entre os dois modelos, Cascata e Ágil. A figura destaca a implementação de todas funcionalidades do escopo em cada fase do ciclo de vida de desenvolvimento no modelo cascata, ante a implementação de um número menor de funcionalidades em um ciclo contendo todas as fases de um ciclo de vida de desenvolvimento de software.

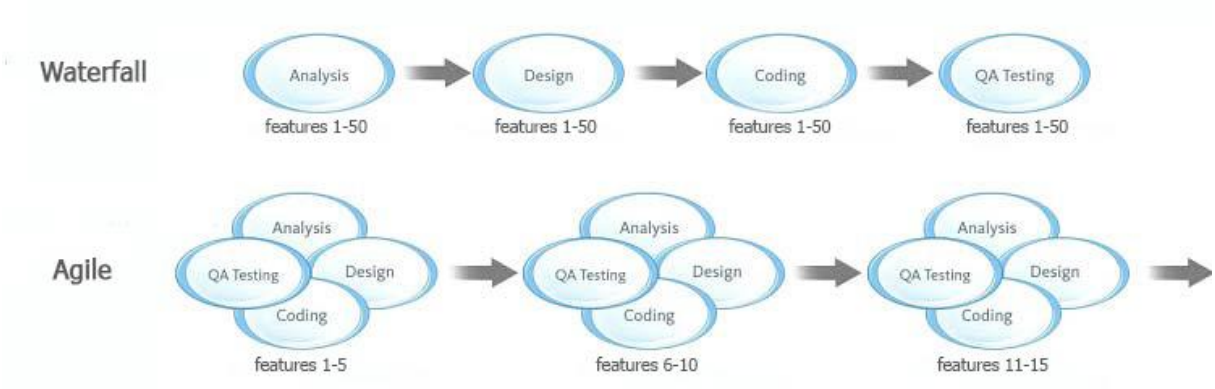


Figura 1 - Ciclo de vida do software no modelo Cascata e Ágil (extraído de [SY])

A referência filosófica para os métodos ágeis é resultado de reunião ocorrida em fevereiro de 2001, entre 17 especialistas de desenvolvimento de software que já vinham praticando metodologias rotuladas como “leves”. Nesta reunião, esses especialistas entraram em consenso para construir um documento de consulta global, que foi chamado de Manifesto do Desenvolvimento Ágil de Software, e possui quatro valores [MANIFESTO]:

- Indivíduos e Interações mais do que Processos e Ferramentas
- Software Funcionando mais do que Documentação Abrangente
- Colaboração do Cliente mais do que Negociação de Contratos
- Responder a Mudanças mais do que Seguir um Plano.

[SHORE e WARDEN, 2008] destacam alguns aspectos importantes das metodologias de desenvolvimento ágil no que diz respeito ao sucesso do projeto, da

equipe e da organização. Neste trabalho estes aspectos são utilizados para guiar a escolha da equipe e da organização.

- A organização poderá rapidamente saber se o projeto está seguindo em direção ao sucesso ou fracasso, e poderá cancelar seus investimentos no momento que achar necessário;
- Será possível focar nos objetivos mais valiosos logo no começo do projeto e fazer entregas rápidas, para mostrar um imediato retorno sobre o investimento (ROI);
- O importante será entregar no prazo, dentro do orçamento, de acordo com a especificação e atendendo as reais necessidades dos usuários;
- Quando o negócio precisar de mudanças, as equipes ágeis poderão mudar de direção rapidamente. O ambiente possivelmente estará propício a sofrer tais mudanças ao ser guiado por metodologias ágeis que já orientam a equipe sobre tais;
- Não só o projeto será ágil, como também a equipe. A equipe eliminará impedimentos, evitará o desperdício, eliminando trabalhos ruins e substituindo-os com práticas mais simples e baratas. A equipe se comunicará com mais rapidez e precisão, serão proativos, revisarão frequentemente seus trabalhos em busca de maior qualidade e trabalharão em conjunto com os usuários em todas etapas de projeto;
- O sucesso do projeto significa entregar valor à organização, conseguir diferenciação competitiva, projeção da marca, estratégia da informação, dentre outros.

2.2. Extreme Programming (XP)

Extreme Programming ou XP é método para desenvolvimento ágil de software criado por Kent Beck. Beck foi um dos colaboradores para a criação do Manifesto Ágil em 2001.

O XP foi criado para atender as necessidades de pequenas equipes de desenvolvimento, que possuem requisitos vagos e em constantes mudanças. É constituído por pequenas iterações de 1 a 4 semanas, em cada qual há uma etapa de planejamento, em que funcionalidades são agrupadas para compor o escopo da próxima iteração, de acordo com exigências do cliente [SOHAIB e KHAN]. No XP há

um forte trabalho colaborativo entre a equipe de programadores e o cliente, que participa da priorização de funcionalidades. Este trabalho colaborativo e de priorização é uma das tarefas consideradas em uma das etapas de projeto do XP, chamado de *Planning Game* [SHORE e WARDEN, 2008].

No início de cada iteração é realizado um *Planning Game*, como uma reunião para priorização das histórias do usuário para serem desenvolvidas e disponibilizadas no próximo *release*. Na já mencionada metodologia SCRUM, o *Planning Game* é chamado de *Pre-game Phase* onde é utilizado o conceito de *Product Backlog,* local este onde são documentados e organizados os requisitos. Nesta fase são definidos os integrantes da equipe, as ferramentas utilizadas e os provaveis riscos do projeto [MACHADO e MEDINA].

Abaixo, é brevemente comentado alguns valores do XP:

- Comunicação – muitos problemas no projeto se iniciam com a falta de comunicação entre os *stakeholders*;
- Simplicidade – o projeto do software se apoia na simplicidade do design, principalmente pelos requisitos estarem em constante mundança;
- Feedback – precisa haver um rápido *feedback* do cliente à equipe de desenvolvimento durante toda a iteração, para que a equipe rapidamente se adapte às mudanças;
- Coragem – coragem para realizar refatoração do código, dispensar documentação, aceitar mudanças e experimentar novas tecnologias.

Quando há o objetivo de aplicar o XP integralmente a um projeto, os criadores destacam 13 práticas [PAMPLONA e KUHN]:

De acordo com [BECK, 1999] todas práticas devem ser aplicadas, e apenas assim poderá descobrir quais são suas fraquezas e quais melhor se adaptam ao projeto em específico. O autor ainda comenta que os pontos fracos de algumas práticas são reforçadas por pontos fortes de outras, há um equilibrio entre elas.

- Planejamento – decide o que deve ser feito imediatamente e o que pode ser adiado;
- Entregas Frequentes – versões funcionais devem ser entregues o mais rápido possível;
- Metáforas – utilizar termos amigáveis entres os stakeholders;

- Projeto Simples – o projeto deve ser simples, legível e de fácil alteração;
- Testes Frequentes – para todo o software que for entregue, é necessário que haja um teste de unidade para garantir o perfeito funcionamento;
- Programação em Pares – enquanto um programador codifica, o outro observa o que está sendo feito, dá sugestões e testa;
- Refatoração – todo desenvolvedor que encontrar um código inadequado, deve melhorá-lo;
- Propriedade Coletiva – todo desenvolvedor tem acesso a todo sistema;
- Integração Contínua – toda alteração desenvolvida deve ser prontamente testada do ponto de vista de integração;
- Trabalho Semanal de 40 horas – não pode haver sobrecarga de trabalho;
- Cliente Presente – o cliente deve fazer parte do desenvolvimento do software;
- *Standup Meeting* – reunião que deve ser feita de pé para alinhar tudo que foi desenvolvido no dia anterior.

Beck [BECK, 1999] destaca algumas observações sobre experiências em projetos mal sucedidos em XP. Para Beck, a equipe não pode mostrar valor a organização através de longas horas de trabalho extra e em alta velocidade. A equipe não pode executar o XP cansado. Se isso estiver ocorrendo, possivelmente há algo errado e precisa ser corrigido. Outra observação é referente ao tamanho da equipe. O XP não pode ser executado com muitos programadores, 10 é um número factível mas 3 ou 4 já é o suficiente para executar algumas práticas. A tecnologia e ambiente também são fatores para o não uso do XP, se seu código leva muito tempo para compilar haverá um longo tempo de trabalho de integração e testes, dificuldado o rápido feedback, ou se sua tecnologia não é flexível o suficiente para suportar rápidas mudanças [BECK, 1999].

No estudo de caso do software e do processo deste trabalho, é contemplada uma pequena equipe de desenvolvedores, a tecnologia escolhida é flexível, moderna e adaptada a mudanças e todos envolvidos estão cientes dos aspectos motivadores da execução do XP.

2.3. Ciclo Zero

Este tópico do trabalho é apoiado nas características de diversos *frameworks*, metodologias ágeis e nas observações de diversos autores sobre o Ciclo Zero.

De acordo com [SUYDAM] um bom planejamento é o pré-requisito para alcançar um produto que funcione corretamente. É importante conhecer, em um ponto de vista empresarial, os sistemas legados antes de iniciar qualquer codificação, pois os requisitos podem mudar ao longo do projeto, uma vez que as mudanças são inevitáveis nos processos de negócios. Ou seja, como experiência da empresa e seus negócios, os sistemas legados podem ter sofrido mudanças frequentes e a equipe de desenvolvimento precisa estar ciente.

O Ciclo Zero é uma fase única antes do início da implementação. As equipes ágeis levam em média de 2 a 4 semanas para realizar este levantamento. A primeira iteração (Ciclo 1 ou Sprint 1) é realizada imediatamente após o Ciclo Zero. Pode ser definido como uma fase pré-implementação para definir a visão geral do produto, metas gerais, planejar as iterações futuras, definir princípios de design e estabelecer papéis para os membros da equipe [SANTOS, 2012].

Um *framework* que trata em detalhes o Ciclo Zero, ou fase referenciada pelos autores como *Inception Phase*, é o *Disciplined Agile Delivery* (DAD). DAD foi desenvolvido pela IBM em meados de 2011, tendo como autor deste e de outros *frameworks* Scott W. Ambler [AMBLER e LINES, 2012]. O *framework* provê orientação para que a equipe alavanque sua eficácia em projetos de TI, mas não prescreve procedimentos obrigatórios.

Na metodologia SCRUM o Ciclo Zero é chamado de Sprint Zero. A metodologia ágil SCRUM foi associada ao desenvolvimento pela primeira vez por Takeuchi e Nonaka [MACHADO e MEDINA] e formalizada em meados da década de 90 por Schwaber, Sutherland e Beedle, e muitas equipes vêem utilizando-o juntamente com o XP, por seu foco ser no gerenciamento de desenvolvimento de projetos e não no aspecto técnico.

Dentro do contexto do SCRUM e durante o Sprint Zero são produzidos um conjunto coerente de planos que visam melhorar a execução do *product backlog* e abrangem temas como gestão de *stakeholders*, marcos e prazos de entrega, estimativas de custo e qualidade [JACOBSEN E SUTHERLAND]. É uma etapa de

visão global e planejamento do projeto e também inclui tarefas de levantamento de requisitos bem como a própria priorização do *product backlog*.

SCRUM e XP são importantes bases para o DAD, tal que incorporam a este responsabilidades de um *Product Owner* em SCRUM, membro responsável por representar os *Stakeholders*, como também é responsável pela entrega de um software utilizável a cada iteração, e no caso do XP, práticas de Integração Contínua, Refatoramento, Desenvolvimento Direcionado a Testes (TDD) e muito mais.

O *framework* DAD sugere algumas estratégias para suportar o aprendizado sobre o domínio, tais como observação dos requisitos iniciais, entrega incremental de software funcional e ativa participação dos *stakeholders* através do ciclo de desenvolvimento da solução.

Os autores reconhecem a necessidade de um período para iniciação do projeto (*Inception Phase*), tipicamente de alguns dias a algumas semanas de duração [AMBLER e LINES, 2012]. Para o DAD esta fase de iniciação termina quando a equipe tem desenvolvida a visão da *release* ao qual os *stakeholders* concordam e também é obtido os fundos para necessários para o resto do projeto.
De acordo com o DAD a *Inception Phase* adota algumas práticas, são elas:

- Formar a equipe inicial;
- Identificar a visão do projeto;
- Conquistar a concordancia dos *Stakeholders* com base na visão proposta;
- Identificar uma estratégia técnica inicial, requisitos iniciais e planejamento das *releases;*
- Configurar o ambiente de trabalho e a equipe;
- Identificar riscos.

Ambler menciona sobre o termo considerado como inapropriado, o Sprint Zero, que por sua vez é adotado pelas equipes de SCRUM. Menciona também sobre o mesmo conceito de fase de iniciação no XP, o *Planning Game*. Vale destacar que a cada iteração também existe uma fase de planejamento com menor duração, onde ocorrem por exemplo, o levantamento e priorização de novos requisitos e construção do planejamento de *release*. Ainda no *Planning Game*, o autor comenta sobre o planejamento da primeira iteração que demanda mais tempo, normalmente com

algumas semanas de duração no inicio do projeto [SHORE e WARDEN, 2008]. De acordo com Ambler, a equipe pode levar de 3 a 9 semanas nestas atividades.

A mesma nomenclatura dada à fase de iniciação pelo DAD bem como suas atividades de planejamento, também é referenciada pelo método *Rational Unified Process* (RUP). RUP [POLLICE, 2003] é um *framework* para processo de desenvolvimento de software, desenvolvido e mantido pela IBM. Provê um guia de orientações e boas práticas para todos aspectos de um projeto, a fim de minimizar os riscos provenientes do desenvolvimento de um novo software.

As decisões iniciais, em processos de desenvolvimento ágil de um modo geral, podem ser tomadas com base nos artefatos de entrega resultantes do Ciclo Zero, são eles [SUYDAM]:

- Escrever as Histórias de Usuários;
- Formular uma definição para as Histórias;
- Estimar o esforço de desenvolvimento;
- Priorizar o *Product Backlog*;
- Desenhar a arquitetura suficiente para iniciar a codificação;
- Criar um plano para liberação de *releases*;
- Fornecer treinamento.

O artefato principal gerado nesta etapa, como já mencionado anteriormente conforme outros *frameworks* e metodologias, é o *Product Backlog*. Futuros ciclos não são iniciados até que este documento seja aprovado, e os riscos e objetivos gerais do projeto estejam em um plano aceitável pela gerência.

2.4. Usabilidade

Usabilidade é: “Grau que um produto ou sistema pode ser usado por usuários específicos para atingir metas específicas com eficácia, eficiência e satisfação num contexto de uso específico” de acordo com a norma ISO 25000 [ISO]. Para caracterizar a usabilidade, a ISO define seis sub-características:

- Grau de Adequação – Grau em que os usuários podem reconhecer se um produto ou sistema é adequado para suas necessidades;
- Capacidade de Aprendizagem – Grau em que um produto ou sistema pode ser usado por usuários específicos para atingir objetivos específicos de

aprendizagem na utilização de um produto ou sistema com eficácia, eficiência, liberdade de risco e satisfação em um contexto de uso específico;

- Operacionalidade – Grau em que um produto ou sistema tem atributos que os tornam fáceis de operar e controlar;
- Proteção de Erro – Grau em que um sistema protege os usuários contra cometer erros;
- Estética de Interface – Grau em que a interface permite uma interação agradável e satisfatória para o usuário;
- Acessibilidade - Grau em que um produto ou sistema pode ser usado por pessoas com as mais diversas características e capacidades a fim de alcançar um objetivo específico em um contexto de uso específico.

Na Web, a usabilidade é um atributo crucial para o sucesso. Se o usuário tem dificuldade em realizar alguma tarefa, ele abandona o a ferramenta Web e provavelmente irá procurar outro que atenda suas necessidades. A usabilidade é geralmente reconhecida como um fator de qualidade do sistema e representa a resposta para muitas interações frustrantes com a tecnologia [MATERA, RIZZO e TOFFETTI, 2006].

Dada a emergente necessidade de usabilidade, os processos tradicionais de desenvolvimento vêm se estendendo para atender os requisitos de usabilidade. Métodos de avaliação de usabilidade passaram a fazer parte de todas as fases do ciclo de vida de desenvolvimento do software [SOHAIB e KHAN]. Esta proposta foi chamada de Design de Interação que conforme os autores citados, é a única maneira de ter eficácia em algumas decisões de projeto durante sua construção e/ou avaliação, através do uso de protótipos [MATERA, RIZZO e TOFFETTI, 2006]. Isto resulta em produtos que são desenvolvidos para atender as necessidades dos clientes.

Ainda, de acordo com os autores, a usabilidade frequentemente é confundida com o UCD ou com a Experiência do Usuário (UX), os quais são campos muito mais complexos, que tem foco em conhecer e entender as necessidades dos usuários típicos do sistema. De acordo com o autor Osama Sohaib [SOHAIB e KHAN], a metodologia chave para a realização da usabilidade é chamada de Design Centrado no Usuário (UCD).

Quando a expressão “Problema de Usabilidade” vem à tona, ela frequentemente se refere a aspectos que fazem a aplicação ineficaz, ineficiente, difícil de aprender e de utilizar [MATERA, RIZZO e TOFFETTI, 2006].

2.5. Experiência do Usuário (UX)

De acordo com Jacob Nielsen, primeiramente, experiência do usuário é atender as necessidades específicas do usuário sem confusão ou incomodo. Em seguida, destaca-se a simplicidade e elegância, para gerar satisfação em usar o produto [NIELSEN].

A experiência do usuário envolve diversas disciplinas e não apenas design de interface. Deve haver uma perfeita harmonia entre seus serviços, sendo as disciplinas consideradas: engenharia, marketing, design industrial / gráfico e design de interface.

De acordo com o autor, a definição de experiência do usuário tem uma definição complexa, já que esta pode se estender a quase tudo onde há interação de alguém com algum produto, a partir do texto de um botão de pesquisa, um esquema de cores ou até uma linguagem utilizada para dar suporte ao cliente [KUNIAVSK, 2003].

É importante distinguir a usabilidade da experiência do usuário, a usabilidade é um atributo de qualidade da interface do usuário, enquanto experiência do usuário é um conceito mais amplo, no qual trata de todos aspectos sobre a interação de usuários finais com a companhia, serviços ou produtos.

Para minimizar a complexidade e entender a Experiência do Usuário, é necessário dividi-la em partes gerenciáveis. No desenvolvimento de sistemas Web há três categorias gerais de trabalho [KUNIAVSK, 2003]:

- Arquitetura de Informação é um processo de organização de informações ao qual está tentando entregar na forma de um produto;
- Design de Interação é a forma que a estrutura é apresentada aos usuários;
- Design de Identidade provê aumento da personalidade e a atração de um produto.

2.6. Design Centrado no Usuário (UCD)

A prática de criar uma eficiente Experiência do Usuário através do atendimento de suas necessidades é chamada de Design Centrado no Usuário. Embora muitos projetos ignorem as reais necessidades dos usuários, é possível que existam certas atividades que visam aumentar o valor destas necessidades no processo de desenvolvimento do software e um candidato para estas atividades é o UCD [KEINONEN, 2008].

UCD é um termo que foi lançado em 1986 por Donald A. Norman e Stephen W. Draper no título do livro Sistema de Design Centrado no Usuário – Nova Perspectiva na Interação Homem-Computador [NORMAN, 2002]. O livro trata das pessoas e como os sistemas podem auxiliar nos seus trabalhos mentais. Os autores enfatizam que o livro não é um “como fazer”, mas sim um guia para novas maneiras de pensar sobre o UCD no design do sistema.

A Associação dos Profissionais de Usabilidade (UPA) definem UCD como “Uma abordagem de design que fundamenta o processo de informações sobre as pessoas que irão utilizar o produto” [KEINONEN, 2008]

Muitos produtos de software ainda são entregues com falhas de usabilidade e isso se deve aos grandes projetos de TI envolvendo muitos *stakeholders* [SVANAES e GULLIKSEN, 2008]. A maioria destes problemas de usabilidade poderia ter sido evitada se houvesse um estágio inicial do processo de desenvolvimento que visasse a aplicação de técnicas e métodos de UCD.

2.7. Métodos e Práticas de UCD

Existem diversos métodos para a compreensão dos usuários e a geração de ideias nas fases iniciais de projeto, tais como: Projeto de Protótipos, Investigação Contextual, Entrevistas e Design Participativo [MATTILA, ROTO e HASSENZAHL] [GEORGE, 2008].

Em 1998 James Hom compilou um manual com diversos métodos de avaliação de usabilidade, dando sua definição, explicando como é e quando deve ser aplicado [HOM].

[LEAVITT e SHNEIDERMAN] desenvolveram um manual, o qual foi utilizado neste estudo, que contém diretrizes de usabilidade, onde obteve o apoio de diversos

especialistas assim assinados como “Forças de Evidência”. O manual foi disponibilizado no website [USABILITY].

Alguns destes métodos são particularmente úteis para o processo de pesquisa de usuário e levantamento de requisitos e por isso, são frequentemente aplicados à etapa de iniciação do projeto ou Ciclo Zero. Esses métodos foram descritos de forma resumida nos tópicos abaixo.

2.7.1. Investigação Contextual

A investigação contextual é definida como: “Um estudo que observa como os usuários interagem com os equipamentos e interfaces em seu próprio ambiente” [USABILITY].

De acordo com [SANTOS, 2012], a investigação contextual é vista como uma entrevista individual conduzida no local de trabalho dos usuários, que focam em observações sobre o trabalho em andamento. Em sua definição completa, é um conjunto de métodos e ferramentas poderosas que levam à compreensão dos usuários em suas vidas cotidianas [GARRET, 2011].

A única desvantagem do método é que ele pode ser caro e muito demorado quando se procura um profundo conhecimento dos usuários. Em outros casos, o método pode ser barato e rápido embora tenda a não produzir o conhecimento da pesquisa completa [GARRET, 2011].

A investigação contextual é um processo de pesquisa de usuário com base em entrevista estruturada, partindo de três princípios fundamentais:

- A compreensão do contexto em que o produto é utilizado é essencial para o bom design;
- O usuário é o parceiro no processo de design;
- O processo de design de usabilidade deve ser focado.

De forma simplificada, a Investigação Contextual é composta pela visita de membros da equipe do projeto ao local de trabalho do usuário, avaliando seus hábitos, atividades e rotinas de trabalho, ou realizando entrevistas onde os usuários contam suas histórias sobre suas experiências. Este é um método eficaz quando se precisa entender o contexto de trabalho dos usuários e é utilizado nos estágios de planejamento do projeto, já que muitas informações obtidas são subjetivas.

A Investigação Contextual inclui métodos de Análise de Tarefas como meio de examinar os passos precisos dos usuários ao realizar tarefas. As tarefas podem ser muito focadas (como a compra de ingressos para o cinema) ou mais amplas (como aprender sobre regras internacionais de comércio) [GARRET, 2011].

O Ciclo Zero é o momento ideal para se realizar a Investigação Contextual, para compreender o produto que se deseja construir, embora possa-se pensar que em todos os releases do produto podem ser acompanhados por investigações contextuais como forma de direcionamento dos releases seguintes.

2.7.2. Protótipos de Alta e Baixa Fidelidade

Prototipação é definida como “uma versão preliminar de um produto que lhe permite explorar suas ideias e mostrar a intenção por trás de uma funcionalidade ou um conceito de design global aos usuários antes de investir tempo e dinheiro no desenvolvimento” [USABILITY].

A fidelidade do protótipo está associada ao estágio do ciclo de vida do produto. Em estágios iniciais, protótipos de baixa fidelidade materializam o conceito do produto, ainda em alto nível, podendo ser construídos em papel ou outras ferramentas simples. Na medida em que o conceito evolui, o protótipo se aproxima do produto final, oferecendo ao usuário a possibilidade de experimentar a interação, parcial ou completamente, o que se chama de protótipos de alta fidelidade.

No Ciclo Zero, protótipos de baixa fidelidade, em geral, e em especial aqueles em papel são uma ótima maneira de obter o rápido *feedback* sobre a arquitetura preliminar, design e/ou conteúdo, além de serem muito fáceis de aplicar, pois exigem poucos materiais, como papéis, tesouras e etiquetas adesivas.

2.7.3. Ordenação de Cartões (*Card Sorting*)

A ordenação de cartões é definida como “um método usado para ajudar a projetar ou avaliar a arquitetura da informação de um site” [USABILITY]. Aplica-se, portanto, à organização e nomeação do conteúdo que se deseja apresentar ao usuário.

De forma geral, o método propõe que o usuário agrupe a informação a ele apresentada em cartões, de acordo com a sua similaridade, em categorias. Parte-se do princípio que a forma de organizar a informação está associada à forma como se busca a informação, posteriormente.

Para aplicar o método, é necessário começar selecionando todos os tópicos do site que se deseja classificar. Tópicos podem ser frases, palavras, imagens, telas, ícones e outros elementos. Cada tópico deve ser anotado em um cartão individual. Cada participante é convidado a organizar os cartões conforme seu modelo mental no contexto de informação ou de acordo com grupos e categorias que sentem mais naturais [GARRET, 2011].

A análise dos resultados da ordenação de cartões realizadas com diversos usuários pode ajudar a entender como um grupo populacional pensa sobre a organização de informações do website [GARRET, 2011].

Este método é utilizado nos estágios de planejamento do projeto. É útil para agrupar funcionalidades por usuários, construir a estrutura do website, decidir o que colocar em uma tela e classificar a navegação nos menus. Na medida em que a informação cresce, nos demais ciclos, é importante haver uma arquitetura de informação organizada e flexível para receber os novos elementos.

2.7.4. Avaliação Heurística

Na avaliação heurística os especialistas julgam cada elemento da interface contra princípios de usabilidade previamente estabelecidos.

Uma das principais fontes destas heurísticas foi desenvolvida por Jacob Nielsen em meados de 1994 que foi um refinamento da lista original desenvolvida por Rolf Molich em 1990 [NIELSEN].

A avaliação heurística pode ser aplicada em qualquer etapa do desenvolvimento, embora seja mais adequada nos estágios iniciais.

O método pode ser aplicado a protótipos de papéis, ou até a uma simples especificação do projeto. Com isto, uma boa quantidade de problemas de usabilidade pode ser descoberta, antes que qualquer etapa de desenvolvimento se inicie, economizando tempo e dinheiro.

As heurísticas de Nielsen incluem:

- Visibilidade do status do sistema;

- Concordância entre o sistema e o mundo real;
- Liberdade e controle do usuário;
- Consistência e padrão;
- Prevenção de erros;
- Reconhecimento em vez de recordação;
- Flexibilidade e eficiência de uso;
- Estética e design minimalista;
- Ajudar os usuários a reconhecer, diagnosticar e se recuperar de erros;
- Ajuda e documentação.

2.7.5. Entrevistas e Grupos Focais

As entrevistas envolvem frequentemente um entrevistador fazendo perguntas para um participante por vez. O método permite a consulta sobre as experiências e preferências sobre seu produto.

Tanto as entrevistas como os grupos focais são eventos formais e estruturados, os quais há interação com os usuários, questionando-os sobre suas opiniões e experiências sobre o produto. Também é possível aprender sobre as atitudes, crenças, desejos e reações dos usuários.

Este método depende de como você formula as perguntas, do quão claramente você descreve o que quer saber sobre os usuários, como meio de garantir a obtenção de informações corretas [GARRET, 2011].

Diferentes das entrevistas, nos grupos focais diversos usuários participam, normalmente de cinco a dez usuários e o entrevistador está presente para interagir e realizar discussões sobre questões levantadas em suas perguntas. Com isso, é possível identificar problemas comuns que muitas pessoas experimentam [USABILITY].

Este método pode ser aplicado em qualquer estágio de desenvolvimento, mas é particularmente útil em dois momentos. No Ciclo Zero, para que se identifiquem as reais necessidades de usuários, quando os requisitos não estão maduros o suficiente, desta forma, dos grupos focais e entrevistas podem-se extrair requisitos dos utilizadores antes que o projeto se inicie. Segundo, após um período de

experimentação dos usuários com *releases* do produto, para direcionar sua evolução.

2.7.6. Testes de Usabilidade

O teste de usabilidade é definido como “avaliação de um produto ou serviço por meio de testes com usuários representativos” [USABILITY].

Durante o processo, normalmente os participantes irão tentar completar algumas tarefas enquanto os observadores ficam assistindo, ouvindo e tomando notas.

O objetivo geral do método identificar problemas de usabilidade, coletar dados e determinar a satisfação e a facilidade com o qual o usuário tem ao usar seu produto. O observador deve coletar os dados sobre o que o participante está fazendo, quanto tempo demora para completar uma tarefa e quantos erros ele comete durante o uso.

A avaliação pode ser aplicada em diversos ambientes, como em laboratórios com sistema audiovisual, em salas com equipamentos de gravação portáteis, em salas sem nenhum sistema de gravação ou apenas por um observador tomando notas, ou até remotamente.

Para atingir o sucesso nos testes de usabilidade, é necessário desenhar um bom plano de testes, recrutar participantes chaves e analisar e relatar suas descobertas aos membros da equipe.

O observador pode pedir que o participante pense em voz alta, pedindo que ele verbalize sobre o que está fazendo e porque está fazendo aquilo.

Este método normalmente é aplicado nas fases finais do projeto, quando já existe um produto a ser testado, ou num pré-lançamento a fim de erradicar problemas de usabilidade [GARRET, 2011]. Em alguns casos, também é possível testar protótipos aproximados do produto acabado.

No entanto, no Ciclo Zero, testes de usabilidade podem ser conduzidos sobre produtos dos concorrentes, para avaliar sua qualidade e estabelecer parâmetros de comparação de desempenho.

2.7.7. Checklists

As *checklists* ajudam a garantir que os princípios de usabilidade sejam utilizados no projeto.

O método é aplicado inicialmente com a decisão de quais diretrizes de usabilidade serão utilizadas para julgar os atributos da interação do usuário com a interface do sistema. Estas diretrizes podem ser encontradas em diversos lugares e podem ser adaptadas para atender as necessidades específicas do projeto.

As *checklists* normalmente são extensos por tratar de cada atributo da interface. Este aspecto pode tornar o método não muito satisfatório para o aplicador.

No Ciclo Zero, checklists são interessantes para orientar uma discussão sobre quais elementos de interação e aspectos de usabilidade são prioritários no projeto.

2.7.8. Personas

Alguns especialistas acreditam que o designer terá mais sucesso se este projetar a interface que atenda os objetivos de um usuário específico em vez de muitos [WODTKE e GOVELLA].

A identificação de Personas se apoia em localizar os usuários que realmente devem utilizar seu sistema e que talvez utilizam o sistema de seus concorrentes, entrevistando-os para obter a matéria prima para seu trabalho [WODTKE e GOVELLA].

As personas se subdividem em três tipos: primárias, secundárias e complementares [USABILITY].

- Persona Primária – é um usuário comum, importante para o sucesso do negócio, mas é carente de um ponto de vista de design;
- Persona Secundária – é um usuário avançado ou novato e que não possui tanta importância para o negócio, mas que custa muito no caso de suporte ao cliente;
- Persona Complementar – é um usuário que tem diferentes necessidades sobre o sistema, pode necessitar de características específicas no sistema para atender suas expectativas.

O objetivo das personas é criar representações realistas e confiáveis de seu principal público de referência [USABILITY]. As personas:

- Representam um grande grupo de usuários do seu sistema;
- Fornecem uma estrutura de dados portátil que permite a todos os membros da equipe de desenvolvimento ter uma base de referência comum [BAKER e BROSCHINSKY, 2008];
- Expressam e concentram as principais necessidades e expectativas dos grupos mais importantes de usuários;
- Ajudam na descoberta de características universais e funcionalidades;
- Descrevem pessoas reais, suas origens, metas e valores.

Autores consideram a criação de três a cinco personas durante a etapa de planejamento do projeto, que inclua informação de nome, idade, foto, informações pessoais e informações de habilidades com o computador [LEAVITT e SHNEIDERMAN].

2.7.9. Cenários

Os cenários descrevem as histórias e contextos por trás de um determinado usuário, podem ajudar a definir como ele alcança um objetivo dentro do seu website [USABILITY].

Os cenários são fundamentais tanto para o design da interface como também para os testes de usabilidade.

Existem três tipos de cenários, são eles:

- Cenários baseados em tarefa: definem apenas o que o usuário quer fazer, mas não como ele faz.
- Cenários elaborados: dão mais detalhes sobre as histórias do usuário, ajudam a destacar características deles para orientar a equipe na escolha de funcionalidades, conteúdo e comportamento do website.
- Cenários baseados em tarefas com escala completa: incluem os passos de como realizar uma tarefa. Este tipo de cenário é muito parecido com os Casos de Uso, mas colocando os passos do ponto de vista do usuário.

Na prática, por conta da limitação de tempo, a análise dos cenários deve ser feita para dez a doze tarefas. Não há necessidade de incluir todos os passos para se alcançar um objetivo, pois o teste de usabilidade irá dizer se a interface está usável e desta forma, se é possível completar o cenário.

3. Processo de Pesquisa Literária

Parte da pesquisa de trabalhos relacionados foi realizada no acervo da IEEE (*IEEE Xplore*), pois dentro do ambiente da USP era possível acessar o conteúdo completo de todos os trabalhos publicados.

O *IEEE Xplore* é uma poderosa biblioteca digital para consulta de artigos científicos e técnicos que foram publicados pela IEEE ou parceiros e oferece mais de três milhões de documentos para leitura e *download*.

A busca foi realizada nos metadados e no conteúdo completo dos documentos disponibilizados na *IEEE*. Outra parte da pesquisa foi realizada na ferramenta de busca do Google `scholar.google.com`.

A etapa de revisão bibliográfica teve como foco, a leitura do resumo dos artigos selecionados a fim de encontrar informações relevantes à pesquisa, apoiados em uma rápida leitura e que combinassem os assuntos de interesse, tais como: UCD, UX, *Usability*, *Agile* UCD, XP, *Extreme Programming*, SCRUM, *Iteration Zero*, *Sprint Zero* ou *Cycle Zero*. Após a pesquisa, todos os resultados foram documentados em uma planilha (incluindo o título, autor(es), ano da publicação e resumo) e todos aqueles que não atendiam, eram repetidos ou com datas de publicação anteriores há seis anos, foram excluídos da análise.

Foram escolhidos artigos com datas superiores há seis anos, pois diversos documentos pesquisados para este estudo foram publicados entre 2007 e 2009. A fim de encontrar referências mais atualizadas sobre o assunto, foram escolhidos apenas os artigos mais atuais, contendo novas direções para as pesquisas de Usabilidade, UCD, UCD Ágil, XP e Ciclo Zero.

Alguns artigos, por ser frequentemente referenciados por outros autores, indicando possivelmente uma fonte completa e segura de informações, foram analisados independente da sua data de publicação.

Para extrair o máximo de documentos relevantes ao estudo, foram considerados acrônimos, sinônimos e/ou termos mais utilizados em estudos (acadêmicos ou do mercado de trabalho), dos termos acima mencionados, são eles:

- UX - User Experience
- UCD – User-Centered Design
- XP - Extreme Programming
- Sprint Zero - Cycle Zero - Iteration Zero

De acordo com a quantidade de documentos encontrados com os termos relacionados e por diversos autores de todo mundo utilizarem esta língua como padrão, toda busca foi realizada na língua inglesa.

Para as pesquisas no Google e na IEEE foram formatadas algumas *strings*, como exemplo:

(“sprint zero” AND (“UCD” OR “user centered design”)) OR
(“sprint zero” AND “usability methods”) OR
(“sprint zero” AND “user experience”) OR
(“sprint zero” AND “XP” AND “UX”) OR
((“XP” OR “extreme programming”) AND (“user centered design” OR “UCD”)) OR
(“agile UCD” OR “agile user centered design” OR “agile UX”).

Em todo processo de pesquisa literária vale destacar um trabalho em específico, “Aplicação de Práticas de Usabilidade Ágil em Software Livre” de Ana Paula Oliveira dos Santos. No trabalho, a autora aborda um atributo do software, a usabilidade, na qual é fundamental para a qualidade durante o uso do sistema.

A autora estuda as principais práticas de usabilidade no contexto de métodos ágeis e no contexto do software livre, classificando-as de acordo com as fases do UCD. Em alguns momentos, o texto contempla informações sobre a definição, características e atividades do Ciclo Zero na qual guiaram o desenvolvimento deste trabalho.

Muitos dos materiais utilizados nesta monografia foram referenciados na tese de mestrado da Ana Paula. Estes materiais foram estudados de forma mais aprofundada e forneceram instruções para o desenvolvimento da monografia e orientações para atividades aplicadas no estudo de caso.

4. Análise da Aplicação de Técnicas de Usabilidade em Métodos Ágeis

4.1. Introdução ao Capítulo

O Ciclo Zero é frequentemente mencionado pelos autores quando se trata do desenvolvimento centrado no usuário. Os especialistas de usabilidade utilizam esta etapa de planejamento para entender os usuários, explorar seus contextos e identificar suas metas objetivos no projeto como um todo [NAJAFI e TOYOSHIBA, 2008] [SY] [UNGAR e WHITE, 2008] [SILVA, MARTIN, MAURER e SILVEIRA, 2011] [XIONG e WANG, 2010] [KOLLMANN e SHARP, 2009] [SOHAIB e KHAN] [HODA, KRUCHTEN, NOBLE e MARSHALL, 2010] [SANTOS, 2012].

Este trabalho apóia-se no aspecto de interação entre as equipes ágeis e as práticas e métodos do UCD, através da fase de planejamento do projeto, o Ciclo Zero. Estas práticas e métodos são utilizados com a finalidade de incluir o usuário como ativo participante em todo ciclo de vida do projeto. O principal fruto deste trabalho será alcançar maior usabilidade para o sistema.

Neste capítulo são destacados diversos trabalhos que estudam a aplicação do Ciclo Zero dentro do contexto de UCD.

4.2. A Visão do Design a Frente do Desenvolvimento

Os autores Rashina Hoda, Philippe Kruchten, James Noble e Stuart Marshall em “Agilidade no Contexto” elaboraram um trabalho no qual entrevistaram 40 profissionais ágeis, em 16 organizações e quatro estudos de caso, acompanhando na prática suas adaptações dos processos ágeis para caber em seus contextos de projeto [HODA, KRUCHTEN, NOBLE e MARSHALL, 2010].

Conforme comentário feito por um designer, citado no trabalho, os desenvolvedores ficam constantemente focados em “como fazer” do que em “o que fazer”, então, se o problema não estiver bem definido, o desenvolvedor poderá fazer muitas otimizações mas sem aderir a aquilo que os usuários realmente querem. Ainda conforme comentário do participante, os processos ágeis não são muito receptivos a especificações, mas algumas vezes, como o próprio participante destaca “o tiro sai pela culatra”.

Para a prática nomeada como *Design Pipeline* [HODA, KRUCHTEN, NOBLE e MARSHALL, 2010], os autores definem o próprio termo para o projeto de interface todo a frente do desenvolvimento. No *Design Pipeline*, a criação do *front-end* ou design da interface domina o chamado Ciclo Zero, desta forma, o Ciclo Zero é visto como uma etapa de planejamento muito mais focada na equipe de design do que na de desenvolvimento. A prática *Design Pipeline* foi abordada no trabalho a partir de um estudo de caso realizado na Nova Zelândia no qual foi suportado por observações realizadas por um representante do cliente para projetos no ramo de entretenimento.

4.3. A Visão dos Designers UX nos Projetos Ágeis

No trabalho “A importância da Identidade e Visão para Designers UX em Projetos Ágeis”, Johanna Kollmann e Helen Sharp investigam os profissionais de UX que trabalham em diferentes papéis e contextos, a fim de entender a agilidade e adaptação ao ambiente de desenvolvimento ágil. Para tanto, as autoras entrevistaram dez profissionais de UX e observaram uma equipe ágil que trabalhava com SCRUM em uma grande corporação [KOLLMANN e SHARP, 2009].

O contexto ágil leva o especialista em UX a reconsiderar suas técnicas para se adaptar às necessidades do desenvolvimento ágil. As autoras destacam as dificuldades que especialistas UX têm em se adaptar ao contexto ágil e aos desenvolvedores, mas que de um modo geral, trouxe maior qualidade ao produto e satisfação para os envolvidos.

Uma técnica seria permitir os profissionais de UX trabalharem a frente do projeto de desenvolvimento, utilizando ferramentas ágeis e que facilitassem a comunicação.

Muitos participantes entrevistados pelas autoras se referem a uma “fase exploratória” como Sprint Zero, Iteração Zero ou Ciclo Zero, mas que dependendo da complexidade do projeto, uma ou duas semanas podem ser pouco tempo e sentem necessidade de mais recursos de UX nesta etapa de planejamento e visão do projeto. Outro participante comenta sobre um possível Sprint Zero a cada trimestre, a fim de manter a visão da participação do usuário em dia e garantir uma correta priorização dos requisitos.

As autoras ainda mencionam um trabalho de Ambler “Usabilidade Adaptativa em Projetos Ágeis de Desenvolvimento de Software”, no qual este sugere o uso de protótipos como ferramentas ágeis pela equipe UX e o uso de Personas e Cenários pela equipe ágil, a fim de auxiliar o levantamento de requisitos do sistema.

Continuando a tratar dos requisitos, o autor menciona que os projetos de UX se apoiam no tradicional modelo cascata (*Waterfall*), em que a documentação é a base para a comunicação entre os envolvidos. Diferente disto, nas equipes ágeis há mais interação entre envolvidos e por conta disto, há menos documentação. No entanto, as documentações frequentemente não aderem ao comportamento interativo do sistema e por conta disto, as equipes utilizam técnicas como personas, *wireframes* e protótipos para apoiar o design interativo.

Um membro de uma equipe com seis anos de experiência em desenvolvimento ágil, colaborador de uma empresa do contexto de consultoria em UX e apelidado no trabalho de Kollmann e Sharp como P9, comenta: “Nós concordamos em fazer uma pesquisa inicial que foi focada em Investigação Contextual com usuários. (...) O produto chave da entrega foi Personas (...) e Cenários de alto nível.”

4.4. A Visão de Tarefas e Esforço

Em “O Estúdio de Design: o design de interface para equipes ágeis”, Jim Ungar relata a experiência em relacionar UCD dentro de equipes ágeis, durante um dia do estúdio de design [UNGAR e WHITE, 2008].

De acordo com a experiência da equipe do estúdio de design, eles perceberam que algum trabalho de planejamento precisava ser feito para que um Sprint ágil no contexto UCD se iniciasse. O autor apresenta um plano de pesquisa UCD para o suposto Sprint Zero do contexto de SCRUM:

- Identificar os objetivos de negócio e grupos de usuários;
- Realizar entrevistas e Investigação Contextual;
- Analisar os resultados da pesquisa e apresentar aos *stakeholders*;
- Fazer recomendações de design;
- Criar protótipos;
- Testar os protótipos;
- Fazer melhorias nos protótipos baseadas nos testes de usabilidade.

Estas tarefas, por ocuparem muito tempo, não se adequam ao prazo estipulado para um Sprint, desta forma, o design do produto deve ficar à frente do desenvolvimento para que as equipes UCD tenham tempo suficiente para planejamento e design.

De acordo com o autor, com este esforço inicial, é possível investigar as necessidades dos usuários, reunir informações suficientes para os *designs* de interface subsequentes e realizar testes de usabilidade.

4.5. Estudos de Caso - UX Agil

Os autores Maryam Najafi e Len Toyoshiba [NAJAFI e TOYOSHIBA, 2008] comentam sobre 2 estudos de caso, um sobre o projeto Falcon e outro sobre o projeto Razor construídos pela empresa VeriSign `www.verisign.com`. O primeiro projeto trata do desenvolvimento de um sistema para os usuários adquirirem e gerenciarem dispositivos para proteger suas contas contra fraudes eletrônicas e outros tipos de roubos de identidade. O segundo projeto trata-se da reconstrução de um sistema já existente para compra e gerenciamento de serviços online. O objetivo neste segundo projeto era oferecer uma experiência de uso aprimorada para seus usuários, principalmente por conta do sistema anterior oferecer grandes dificuldades no processo de compra e a necessidade frequente de suporte.

No primeiro projeto a equipe UX realizou tarefas de pesquisa de usuários e testes, e considerou um Sprint de design a frente do desenvolvimento. A figura 2 ilustra estes estágios em um contexto completo.

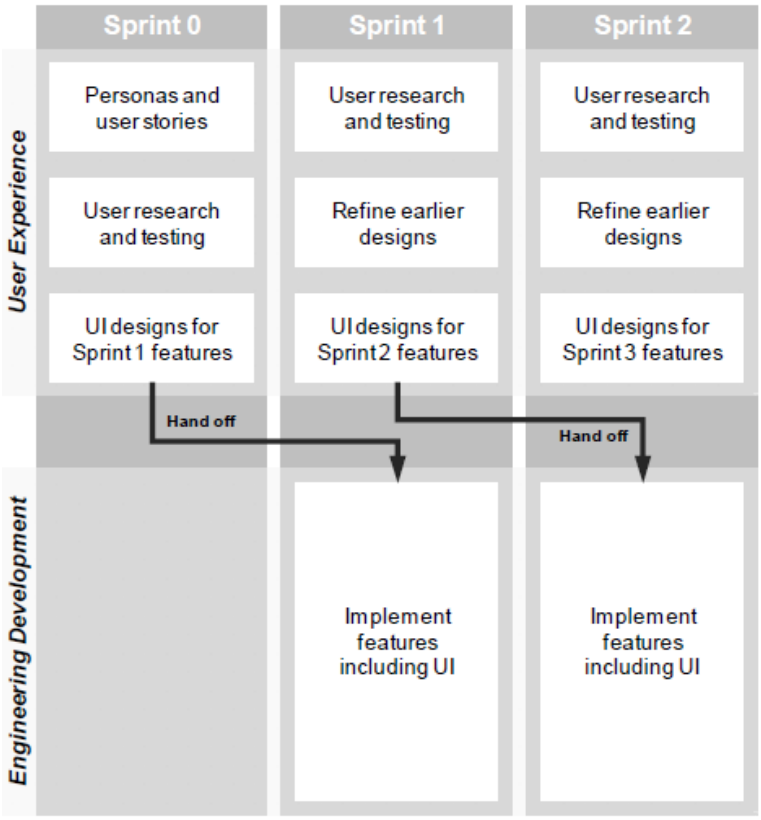


Figura 2 - Método de desenvolvimento em estágios utilizado no projeto Falcon (extraído de [NAJAFI e TOYOSHIBA, 2008])

Os autores comentam que sessões de testes de usuários ajudaram no envolvimento da equipe no processo UCD e promoveu o compartilhamento de ideias e base de informações suficientes para a equipe de desenvolvimento nos Sprints posteriores.

Na VeriSign as equipes UX usam o Sprint Zero para melhor entender os usuários, explorar o contexto e identificar os objetivos do projeto como um todo. Os artefatos gerados são utilizados para reportar as expectativas dos usuários à equipe e negociar a priorização de tarefas para o primeiro Sprint. Personas e Cenários são utilizados para levantar funcionalidades para os Sprints subsequentes.

Os autores concluem que todo o planejamento e teste necessários para garantir a máxima participação do usuário no processo, rendeu mudanças significativas na usabilidade das ferramentas e no sucesso da integração entre as equipes, no entanto, os prazos de entrega foram afetados.

4.6. A Visão das Técnicas e Práticas

Em “Design Centrado no Usuário e Métodos Ágeis: Uma Revisão Sistemática” [SILVA, MARTIN, MAURER e SILVEIRA, 2011], os autores realizam uma revisão sistemática sobre técnicas e práticas aplicadas no contexto de desenvolvimento ágil com o design centrado no usuário.

Existe uma considerável diferença entre os métodos ágeis e o UCD, no entanto, segundo teses, o uso de métodos ágeis podem resultar na melhoria da usabilidade. A diferença entre as abordagens é como elas encaram a alocação de recursos no projeto. Enquanto os métodos ágeis se esforçam em entregar pequenos pacotes de software utilizável para o cliente de forma rápida, o UCD gasta grande tempo com pesquisa e análise dos usuários antes que o desenvolvimento se inicie, gerando grandes volumes de documentação e demorando meses para serem desenvolvidos, o que seguem em sentido contrário aos preceitos de métodos ágeis.

Mesmo com suas diferenças, estas abordagens se convergem em alguns aspectos, como no levantamento de requisitos e que elas são focadas no usuário e cliente.

De acordo com a revisão sistemática, diversos artefatos, práticas e necessidades foram destacadas pela quantidade de ocorrências nos artigos, por exemplo: LDUF (Pouco design a frente), Estreita Colaboração (desenvolvedores entendendo o que os designers dizem), Protótipos de Baixa Fidelidade (desenhos de telas em papéis), Testes de Usuários (usar usuários no teste de usabilidade), Histórias de Usuários, Métodos de Inspeção (usabilidade) e Um Sprint a Frente (a equipe UCD trabalha um Sprint a frente da equipe de desenvolvimento).

Os autores concluem que no Sprint Zero a equipe de desenvolvimento pode realizar as seguintes atividades de UCD: Investigação Contextual, Análise de Tarefas, Entrevistas, Protótipos em Papéis, Cartões de Design, Histórias de Usuários com critérios de aceitação e aspectos de usabilidade e Cartões de Funcionalidades.

Também comentam que a partir da revisão sistemática, a maioria dos projetos têm um especialista em UCD ou até mesmo uma equipe envolvida.

Embora exista muitas pesquisas de integração de processos ágeis e UCD, nenhum deles é analisado com experimentos controlados, conforme evidências de lições aprendidas e experiências relatadas.

4.7. A Visão da Integração

Em “Integrando Engenharia de Usabilidade e Desenvolvimento Ágil de Software” [SOHAIB e KHAN], os autores apresentam uma revisão de literatura de como os processos ágeis vêm trabalhando em conjunto com as práticas de usabilidade com o objetivo de tornar o software fortemente usável e eficaz. O objetivo geral do trabalho foi identificar os pontos comuns entre as práticas ágeis e a engenharia de usabilidade.

No artigo, os autores respondem a duas perguntas: Quais são as dificuldades na integração entre as duas abordagens e Que práticas são sugeridas na integração da usabilidade com os métodos ágeis.

Os autores comentam em suas revisões literárias que o cliente deve participar ativamente do levantamento de requisitos juntamente com a equipe de desenvolvimento, e também testar o produto de software. Os métodos ágeis apresentam deficiências quando tratam de requisitos centrados no usuário.

A tabela 1 apresenta uma tradução livre da tabela de integração entre a engenharia de usabilidade e os métodos ágeis, ao qual foi ilustrada e comentada no artigo.

Com base na tabela de referência, para a iteração de planejamento (Sprint Zero), importantes aspectos são destacados, como as histórias de usuário em conjunto com a técnica de design baseado em cenários.

Conceitos de Métodos Ágeis	**Engenharia de Usabilidade**	**Abordagem Sugerida**
Entregar software funcional com frequência	Abordagem de software tradicional mas com iteração dentro das fases	Desenvolvimento Iterativo ao logo do projeto
Requer pessoas generalistas	Requer pessoas especialistas	Montar uma equipe multidisciplinar para garantir a experiência completa
Foco no cliente	Foco no usuário	Colaboração entre os *stakeholders* irá maximizar a experiência da equipe para

		um produto utilizável
Desenvolvimento dirigido a teste e integração contínua	Investigação Contextual, Pesquisas de Campo, Métodos de Inspeção de Usabilidade para os testes	Testes de Unidade, Testes de Aceitação do Usuário, Testes de Usabilidade ao logo de todo projeto
Usando o cliente no local, requisitos funcionais são encapsulados como histórias de usuários	Design baseado em cenários para a análise de requisitos	Integrar histórias de usuário com design baseado em cenário

Tabela 1 - Combinação de abordagens de engenharia de usabilidade e métodos ágeis (Tradução livre de [SOHAIB e KHAN])

Por fim os autores concluem que as abordagens ágeis e de engenharia de usabilidade trabalham bem em conjunto, ainda mais tratando-se da perspectiva do usuário. A engenharia de usabilidade pode ser adaptada ao contexto ágil com base na utilização de abordagem mais iterativa durante todo ciclo de vida do desenvolvimento, incluindo também especialidades em usabilidade no projeto.

4.8. A Falta da “Grande Figura”

De acordo com Ana Paula Oliveira Santos em “Aplicação de Práticas de Usabilidade Ágil em Software Livre” [SANTOS, 2012], o Sprint Zero é uma etapa crítica para se realizar algum trabalho de UX ou usabilidade, antes de iniciar a codificação. Na comunidade ágil esta etapa é realizada por uma equipe UX mas em alguns projetos outros membros podem ser envolvidos, como os desenvolvedores.

Pela falta da “grande figura” na iniciação e planejamento do projeto, as equipes ágeis normalmente realizam trabalho desnecessário, por não terem as prioridades claras. As equipes também relatam sobre a falta de tempo para trabalho de pouco design a frente (LDUF).

Por vezes, a “grande figura” é necessária no projeto quando a demanda de artefatos gerados fica difícil de ser gerenciada, quando a equipe começa a perder o foco sobre quais atividades estão por vir e quando perdem a visão do cenário geral do projeto. Esta “grande figura” normalmente é a visão apenas do Gerente de Projeto, Líder da Equipe, Analista de Negócio, Proprietário do Produto ou simplesmente o *SCRUM Master*.

Conforme descrito na tese, a falta da “grande figura” pode ser minimizada com a consideração do Sprint Zero. Dentro desta mesma fase de projeto, uma forma de garantir que você está construindo o produto certo, é realizar uma pesquisa geradora para definir os perfis de usuários através de personas e cenários.

4.9. A Visão das Experiências na Adaptação UCD Ágil

Em “Adaptação de Investigações de Usabilidade para Design Centrado no Usuário Ágil” [SY], elucida exemplos práticos da integração entre UCD e processos ágeis, ajustando assim o tempo gasto e a granularidade destas investigações. A autora comenta sobre o design à frente da codificação “*design up-front*, onde os designers trabalham sempre um ciclo a frente dos desenvolvedores”.

Para a autora, o Ciclo Zero pode conter atividades de investigação da usabilidade, como uma breve fase de coleta de requisitos. A figura 3 ilustra os ciclos de design e as atividades previstas por cada um destes ciclos. A coleta de requisitos dependerá de como o produto está quanto a seu lançamento, se é um produto já existente ou completamente novo. O Ciclo Zero pode incluir as seguintes atividades:

- Recolhimento dos dados e constituição de uma visão compartilhada entre os membros da equipe;
- Quando um produto é completamente novo, devem ser realizadas atividades de entrevistas ou investigações contextuais durante as visitas do cliente para a validação de mercado, também é necessário preparar e derivar designs de alto nível para guiar as decisões de características do produto;
- Quando um produto já existe, devem ser analisados e sumarizados as prioridades os resultados da investigação contextual e os testes de usabilidade;
- Quando é um mercado ou capacidade completamente novo, devem ser feitas breves descrições dos usuários alvo e seus fluxos de trabalho (personas e cenários).

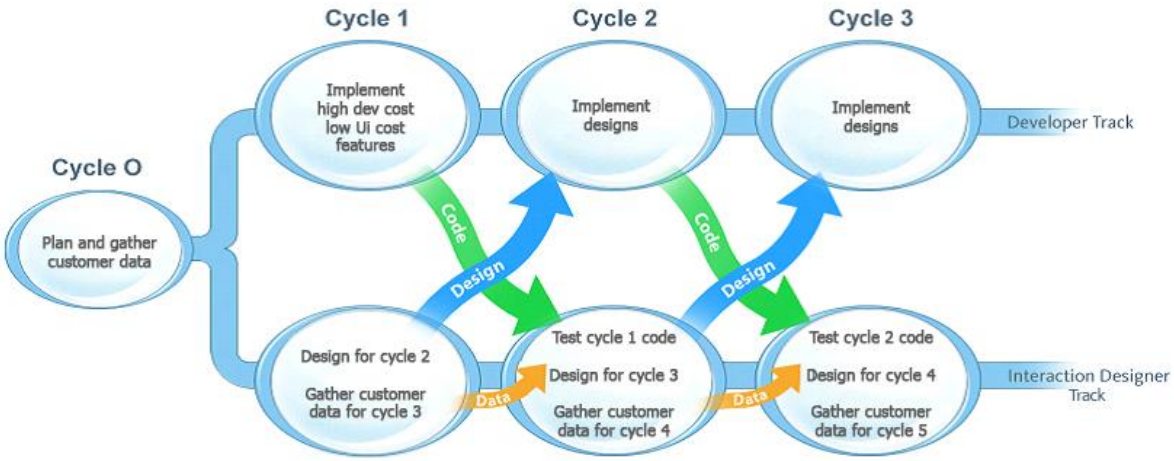


Figura 3 - Ciclo Zero e design à frente da codificação (extraído de [SY])

Ainda na figura 3, a autora apresenta 2 fluxos de criação, um de desenvolvimento, outro de design de interação. No segundo fluxo os especialistas em UX trabalham no design do ciclo posterior e recolhem dados do cliente para 2 ciclos a frente. Quando os desenvolvedores iniciam seus trabalhos, os dados de investigação e os designs de interface já estão disponíveis para então iniciar a codificação. Os especialistas em UX também testam com o usuário o que foi implementado no ciclo anterior.

Na fase de iniciação do projeto, onde os especialistas UX estão trabalhando, a equipe ágil está avaliando e construindo uma arquitetura preliminar do sistema, na qual nenhuma atividade de interação com o usuário é necessária.

A autora também comenta sobre a granularidade, tempo gasto do projeto, a maneira de relatar as descobertas de usabilidade e os caminhos paralelos que ocorrem entre os membros da equipe UX e ágil. Em processos ágeis, a equipe se foca apenas em um conjunto de requisitos por vez e a equipe UX não precisa trabalhar imediatamente em todos *designs* das interfaces e nem antes do início da codificação.

4.10. Considerações do Capítulo

Com base na análise da aplicação de técnicas de usabilidade em métodos ágeis, diversos aspectos da combinação das duas abordagens no contexto do Ciclo Zero foram identificados a fim de apoiar o processo de desenvolvimento do estudo de caso. Destacando-se pela orientação sobre atividades propostas para a fase de iniciação, como: realizar entrevistas com usuários e grupos focais, realizar a investigação contextual, criar personas, fazer o levantamento de cenários e desenhar protótipos.

Em resumo, nos trabalhos os autores comentam frequentemente sobre a ativa participação do cliente e usuário no projeto, e sobre o ganho de qualidade do produto final com a combinação das duas abordagens. É comum ter especialistas em UX dentro da equipe que frequentemente trabalham do design de interação à frente do desenvolvimento, ou seja, antes que a codificação se inicie.

Estes trabalhos forneceram considerações importantes sobre quais atividades incorporar no trabalho de desenvolvimento do estudo de caso.

5. Objetivos e Expectativas

Neste capítulo são tratados aspectos que motivaram a escolha do processo ágil juntamente com a inclusão do usuário no centro do desenvolvimento, com base em uma série de necessidades da empresa que adquiriria o produto, da equipe de TI e do perfil do sistema a ser desenvolvido.

Em princípio a empresa Corttex não contava com uma equipe de TI alocada e dedicada a construção de soluções de software para atender as necessidades internas. Há pouco mais de 25 anos, a empresa contratou e alocou sua demanda de sistemas em uma fábrica de software terceirizada que então construiu o ERP para gestão empresarial, incluindo módulos de Compras, Vendas, Financeiro, Fiscal, Contabilidade, Estoque e RH. No entanto, os 25 anos se passaram e o sistema continua o mesmo, desenvolvido em Cobol, no qual o usuário nem ao menos pode utilizar um mouse para navegação em suas telas. Desta forma, seria necessária uma radical evolução tecnológica e uma grande mudança de paradigma para os atuais usuários.

Pensando na segurança das informações da organização, propagação de conhecimento sobre o ERP em aspectos técnicos, continuidade do projeto, e principalmente, evolução tecnológica, a empresa alocou uma nova equipe de TI multidisciplinar composta de poucos colaboradores Analistas, Programadores, Designers e Gerente. A equipe ficou responsável por fazer a migração dos dados do antigo sistema, implantar o ERP da TOTVS (Protheus), desenvolver uma ferramenta para acesso externo de clientes, representantes de venda e gerentes, e desenvolver uma ferramenta de *back-office* para maior agilidade na aprovação de pedidos e emissão de relatórios gerenciais.

Este trabalho apoia-se no estudo e implantação das práticas de UCD no Ciclo Zero para a ferramenta Web, composta de recursos a atender as necessidades de acesso externo pelos clientes, representantes e gerentes de vendas. É importante salientar que a ferramenta Web forneceria à organização maior visibilidade de seus produtos e negócios, além de atingir uma fatia do mercado que a mesma não se enquadrava.

A primeira tarefa para o gestor de TI é de conseguir fundos e a equipe suficiente para implantação e desenvolvimento das soluções propostas. Ao mesmo

tempo, é necessário contemplar a visão geral dos projetos, analisar suas viabilidades e riscos.

Com a equipe multidisciplinar reunida, precisou-se de pequenos protótipos não-funcionais para entender as reais necessidades dos usuários e *stakeholders*, contemplando a entrada e manipulação dos dados conforme o sistema legado já disponibilizava, levantar algumas regras de negócio despercebidas pelos analistas durante a navegação na interface e pesquisar soluções para garantir satisfatória usabilidade já que o sistema legado, por mais que antigo e desatualizado, fornecia organização da informação e teclas de atalho que maximizavam a agilidade no processo. Precisava-se, também, de definir uma ou mais arquiteturas viáveis para este sistema e construir um pequeno conjunto de funcionalidades para testá-las.

A adoção de uma metodologia ágil tem diversos fatores motivadores. Primeiramente, a equipe é reduzida mas munida de conhecimento em diversas áreas disciplinares, como exemplo dos analistas/programadores que possuem algum conhecimento em desenvolvimento de interface gráfica e aplicação de técnicas de UCD. Outro fator motivador é a experiência e o contato de todos os membros da equipe com o departamento comercial de uma indústria do ramo têxtil, estes que absorvem a maioria das regras de negócio indispensáveis para o projeto da ferramenta Web. Vale salientar a necessidade de adaptações nos sistemas para atender as mudanças de negócio vindas do departamento comercial, como exigências da própria empresa, ou até por exigências legais.

Algumas características são exigidas dos participantes membros da equipe de TI quando feita a seleção para sua formação, como: ser comunicativo, auto gerenciável, multidisciplinar, disciplinado e autodidata.

Mesmo tendo uma pequena equipe alocada, é necessário o máximo de produtividade desta, pois muitas soluções em TI devem estar prontas em curto prazo. Algumas características dos processos ágeis, se adaptam bem ao perfil da equipe e ao projeto, como: simplicidade de código, desenvolvimento orientado a testes para minimizar a ocorrência de erros, programação pareada para maximizar a qualidade do código, reuniões diárias de pé para manter um conhecimento uniforme entre a equipe prevendo potenciais problemas que poderiam surgir e envolvimento do cliente e usuário para priorizarem os requisitos e serem participantes ativos em todo ciclo de desenvolvimento, contribuindo com a rápida visão dos riscos e experiências com sistemas anteriores.

Toda esta análise é apoiada em não uma metodologia ágil, mas sim nas combinações de diversas delas, como SCRUM, XP e DAD. Considerou-se que nenhuma metodologia deveria ser implantada integralmente (à risca), mas sim conduzir a fase de iniciação (Ciclo Zero) do projeto, considerando as melhores práticas de cada uma delas para atender as reais necessidades da equipe, dos projetos, dos usuários e do negócio, e acima de tudo, incluir na fase de iniciação os principais métodos de UCD.

Conforme os comentários acima sobre métodos ágeis e o desenvolvimento com a forte participação dos *stakeholders* e usuários, inclusive por todo estudo realizado nos artigos e teses de referência, percebe-se que os métodos ágeis e o UCD estão intimamente relacionados, um apoiando o outro. Para o projeto, base deste estudo, é necessário entender quais suas correlações e avaliar os resultados obtidos.

Como meta definida pelo gerente de projetos, esta fase de iniciação não pode ultrapassar 6 semanas. O capítulo 7. ANÁLISE DE RESULTADOS traz explicações sobre o sucesso ou fracasso das atividades propostas.

Em “A Arte do Desenvolvimento Ágil” [SHORE e WARDEN, 2008], os autores comentam sobre um perfil de projeto muito próximo da realidade da organização, o Desenvolvimento Customizado *In-House*, onde há construção de sistemas para atender as necessidades internas da empresa, como automatização dos processos de fábrica (no nosso caso, por exemplo, inclusão dos pedidos de venda) e produção de relatórios gerenciais. No nosso cenário há o grande envolvimento dos *stakeholders* e dos usuários finais, já que eles são facilmente acessíveis. Isso também pode ser considerado como um fator motivador para este projeto como também para este estudo.

Além de indispensável para o projeto, a ativa participação do usuário final também resultaria em um efeito moral: deixá-lo com o sentimento de que é o dono do produto por ter sido um participante ativo durante todo o processo de desenvolvimento. Assim, o usuário fica responsável pelo sucesso e crescimento da organização e seus produtos de software.

6. A Prática de UCD no Ciclo Zero

6.1. Estudo de Caso – *Background*

A Corttex é uma indústria brasileira que atua no mercado há mais de 37 anos e possui um nome de referência no mercado têxtil. Em 2006 uniu-se a Fatex Indústria especializada em artigos para o lar, e assim criou-se um novo segmento, Corttex Casa.

A empresa trabalha com diversos segmentos, como: tecidos planos e estampados, tapetes, cortinas e artigos de cama/mesa/banho. Parte de seus produtos é importada e outra parte significativa é produzida no Brasil, na cidade de Três Lagoas / MS, onde também se encontra seu centro de distribuição.

Atualmente, a empresa, incluindo todos seus departamentos administrativos e chão de fábrica, trabalham com um sistema desenvolvido a mais de 25 anos na linguagem Cobol. Na figura 4, é possível visualizar a tela de digitação de pedido, com interface “pouco” intuitiva e usável, e nada moderna, onde o usuário, por exemplo, fica impossibilitado de utilizar o *mouse* para navegação.

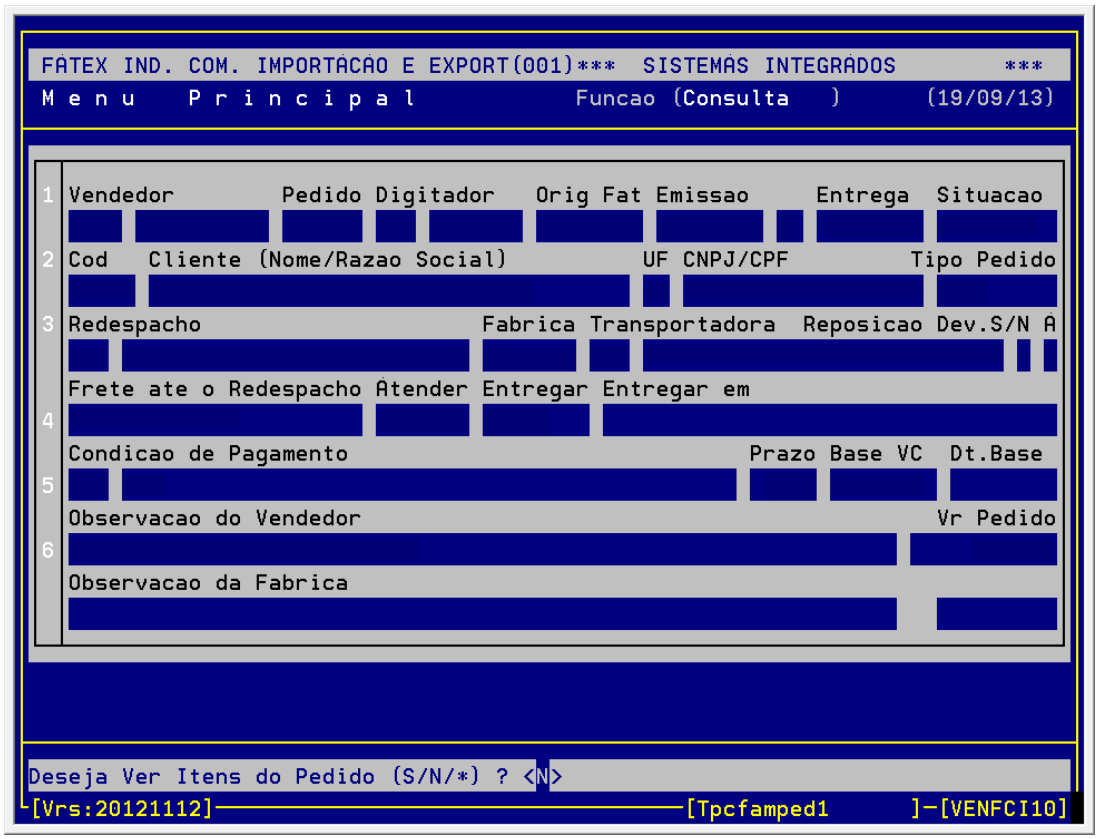


Figura 4 - Tela de digitação de pedido do sistema legado

Após diversas entrevistas informais, fica evidente que a interface do sistema não agrada o usuário, principalmente para aqueles que estão adaptados às novas tecnologias e conceitos da Web. Neste ponto, o estudo e aplicação das práticas de UCD foram necessários, o usuário teria que ser transportado, tecnologicamente, dos anos 80 para século XXI. A compreensão da usabilidade e experiência dos usuários foram os principais fatores para esta transformação. A usabilidade e experiência do usuário são determinantemente críticos para o sucesso em aplicações Web [CONSTANTINE e LOCKWOOD, 2001].

Para ganhar mais destaque no mercado, substituir em partes o sistema legado e organizar seus processos de negócio internos, a empresa comprou um sistema ERP de destaque, o Protheus da TOTVS (antiga Microsiga). A implantação do ERP percorreu uma trilha paralela ao desenvolvimento do sistema em plataforma Web, no qual diversos analistas de negócio estavam implantando cada módulo do sistema, juntamente com seus usuários chave. Este sistema foi em partes, a base de disponibilização dos dados consumidos pela aplicação Web.

O sistema legado não será completamente substituído, pois continuará com suas funcionalidades direcionadas ao chão de fábrica, funcionalidades responsáveis, por exemplo, pela alimentação e movimentação do estoque da empresa.

Para dar andamento ao projeto Web, foi necessário mapear os processos de negócio do departamento comercial, recuperar informações de clientes, vendedores, pedidos de venda, compra, produtos, notas fiscais e duplicatas, do sistema legado, para que fosse possível entender a semântica dos dados e construir as telas em paralelo a implantação do sistema TOTVS.

O sistema Web será construído do zero, tendo como previsto a criação de telas para emissão, alteração e consulta de pedidos de venda, consulta de artigos e produtos, consulta de notas fiscais e duplicadas. Os usuários finais típicos do sistema são clientes (lojas e magazines), representantes de venda, e colaboradores internos da empresa Corttex / Fatex (assistentes, coordenadores, gerentes e diretores).

Espera-se em um futuro não tão distante que todos os pedidos de venda da empresa sejam emitidos pelo ambiente Web, por este ser fácil de utilizar e estar disponível de qualquer lugar em qualquer dispositivo.

6.2. A Equipe

A equipe de desenvolvimento de sistemas da Corttex, durante o projeto, foi composta de dois analistas programadores Web, quatro analistas de sistemas TOTVS um arquiteto de informação e um gerente de projeto. Todos os membros precisaram ser multidisciplinares, já que as práticas de métodos ágeis e UCD foram aplicadas apenas por eles, sem a existência de nenhum especialista de ambas abordagens.

Os analistas programadores Web, além de suas atividades rotineiras, ficaram responsáveis pelo design *front-end* e aplicação de algumas práticas de UCD, sempre acompanhados pelo gerente de projeto (presencialmente, através de artefatos gerados ou reuniões diárias).

Os analistas da TOTVS ficaram responsáveis pela criação, gerenciamento e manutenção do banco de dados, que por sua vez é a base comum de integração entre todos os sistemas que estavam sendo desenvolvidos. Pelas boas práticas (conforme discussões realizadas entre a equipe), todo acesso a estes dados deveria ser realizado via *Webservices*. Os membros da equipe Web apenas criaram rotinas para consumir os *Webservices* disponibilizados pela TOTVS.

6.3. Atividades Propostas – Ciclo Zero

A elaboração de um plano de atividades executadas de forma sequencial foi uma proposta desenvolvida a fim de avaliar os métodos ágeis e de UCD em um estudo de caso, compreendendo suas viabilidades e seus resultados na melhoria do processo de desenvolvimento de software e na melhoria da usabilidade do produto final.

Este capítulo do trabalho trata como estas diversas práticas de métodos ágeis e UCD foram aplicadas sequencialmente na fase de iniciação do projeto ou Ciclo Zero. Por falta de tempo, não foi possível realizar experimentos com algumas delas e desta forma, foram utilizadas apenas aquelas que melhor se adequavam ao cenário atual do projeto, da equipe e da empresa.

Conforme as revisões bibliográficas, diversos autores propuseram algumas atividades que se adequavam à fase de iniciação do projeto. Abaixo são descritas algumas propostas e em seguida, são explicadas como estas combinações foram aplicadas no projeto do sistema Web da indústria têxtil.

Nos tópicos a seguir, a sequência de etapas realizadas durante a iniciação do projeto é explicada em mais detalhes, considerando: a forma como foram aplicadas, quais artefatos gerados e quais conclusões tiradas.

A metodologia XP prevê um fluxo de atividades que visam guiar o processo de desenvolvimento de software, a figura 5 ilustra a organização tradicional deste processo e ainda sem qualquer adaptação proposta por este trabalho. Para este estudo, foram adaptadas e criadas novas práticas executadas na fase de iniciação do projeto, e que no processo tradicional XP fazem parte do *Planning Game* conforme já mencionado na seção 2.3. CICLO ZERO. Conforme descrito pelo próprio guia da metodologia, o projeto pode se iniciar com a captura das *User Stories* (Histórias de Usuários) e com a condução do *Architectural Spike* (Visão da Arquitetura) [XP].

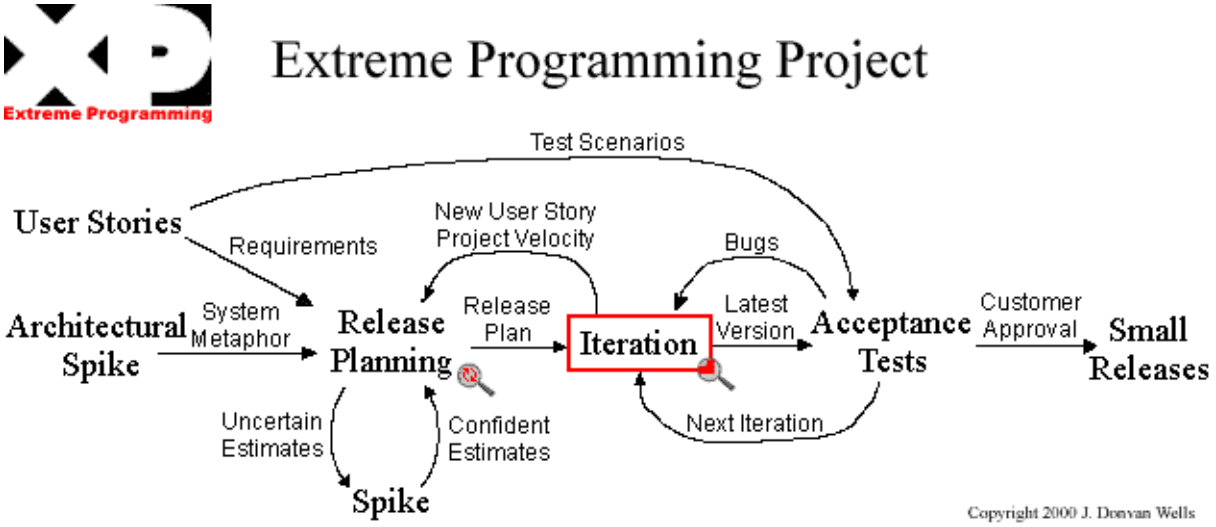


Figura 5 - Fluxo do XP sem adaptação

A figura 6 ilustra a adaptação do ciclo de vida do projeto XP com a inclusão de etapas de UCD no Ciclo Zero (*Iteration Zero*) no que diz respeito às então atividades para o *Planning Game*, como a exploração dos usuários e de suas atividades através da geração de personas e cenários, da construção de protótipos alimentados através das histórias e artefatos gerados na exploração dos usuários, construção e teste da arquitetura preliminar do sistema.

O quadro retangular posicionado sobre as seguintes atividades e artefatos (figura 6) demonstra a proposta para o Ciclo Zero, são elas: *User Exploration* (Exploração do Usuário), Personas, *Scenarios* (Cenários), *User Stories* (Histórias de

Usuários), UI *User Stories* (Histórias de Usuários p/ Interface), *Prototypes* (Protótipos), *Architectural Spike* (Visão da Arquitetura) e UI (Interface do Usuário).

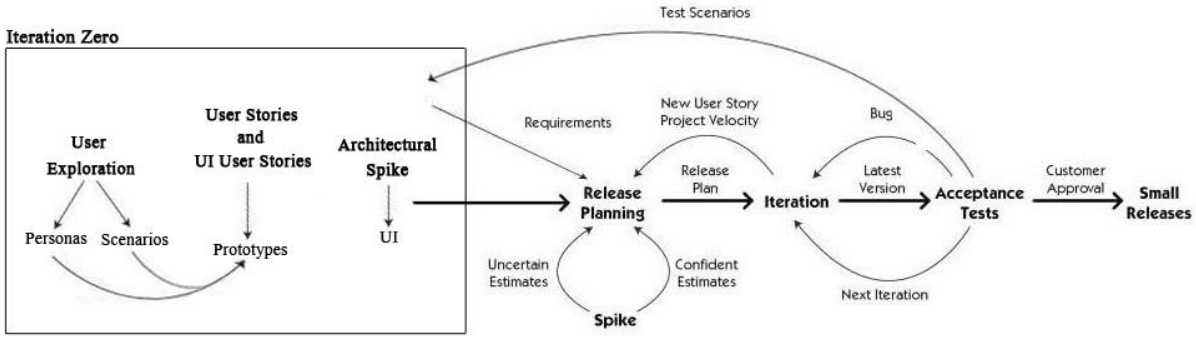


Figura 6 – Inclusão de abordagens UCD no Ciclo Zero (*Iteration Zero*) da metodologia XP (Adaptado de [XP])

A visão geral sobre o projeto, identificação de riscos e definição da equipe foram tratados na etapa 1 – Reunião de Planejamento.

As atividades de exploração dos usuários têm como objetivo gerar maior conhecimento aos membros da equipe de projeto sobre quem é o real usuário do sistema, sobre sua vida cotidiana e suas rotinas de trabalho. Estas atividades foram executadas e analisadas nas etapas 2 – Entrevistas para Análise de Negócio e 3 – Investigação Contextual. Conforme informações obtidas nestas fases, foram construídos os artefatos Personas e Cenários de Uso. As Personas descrevem representações realistas e confiáveis do usuário alvo. Os Cenários de Uso descrevem atividades passo-a-passo executadas pelo usuário alvo dentro de um sistema a fim de atingir um objetivo específico.

As histórias de usuários têm como objetivo extrair as reais necessidades dos usuários e clientes quanto as funcionalidades que o sistema deve ter. São usados em substituição aos extensos documentos de requisitos [XP]. Por sua vez, as histórias de usuários com critérios de interface dão mais detalhes à equipe sobre características do design de interface e design interativo. Esta atividade foi uma adaptação da tradicional atividade de levantamento de histórias de usuários prevista pela metodologia XP. A etapa 4 – Levantamento de Histórias de Usuários, descreve o processo de levantamento e participação do usuário nesta atividade.

Com o levantamento das histórias e com as experiências e resultados da exploração dos usuários, foram construídos os protótipos de baixa fidelidade que demonstram uma visão preliminar da interface do sistema e suas funcionalidades, na qual serão futuramente

implementadas e disponibilizadas em uma versão funcional do sistema. Seu processo foi descrito em detalhes na etapa 6 – Construção de Protótipos.

Por fim, foram realizadas reuniões entre os membros da equipe para discussão e definição da arquitetura do sistema. Na visão da arquitetura espera-se compreender o problema e explorar possíveis soluções [XP]. Uma versão preliminar desta arquitetura foi desenvolvida com a finalidade de compreender sua viabilidade. As etapas 5 – Reunião sobre Decisão Arquitetural e 8 – Implementação da Arquitetura e UI demonstram o processo por trás de aspectos que circundam a arquitetura do sistema em plataforma Web.

Os tópicos a seguir exploram mais detalhes sobre as atividades propostas para o Ciclo Zero como combinação dos métodos de XP com os de UCD.

6.3.1. 1º Etapa – Reunião de Planejamento

Para a iniciação do projeto, foi marcada uma reunião entre todos os membros da equipe de TI, constituída por gerente de projeto, arquiteto, analistas Web e TOVS, para decidir a direção e primeiros passos a serem realizados. Para tanto, houve uma predefinição das responsabilidades de cada membro da equipe, aqueles que ficariam responsáveis por gerir as rotinas ágeis e aplicar as práticas de UCD. A reunião teve duração de meio período comercial.

A fim de manter os servidores de aplicação e banco de dados funcionando, foi incluído à equipe um responsável pela infraestrutura.

Uma importante questão levantada na reunião foi “a equipe realmente está pronta para iniciar o projeto?”, esta pergunta não poderia ser respondida imediatamente. Cada membro da equipe deveria estar preparado para o seu impacto, já que a implantação de um sistema ERP de grande porte se iniciaria e o desenvolvimento de um sistema Web para atender todas as rotinas comerciais também se iniciaria em paralelo e com uma equipe enxuta. Os membros da equipe deveriam aprender a confiar um nos outros, estar suficientemente capacitada e composta por pessoas multidisciplinares ou dispostas a serem. O gerente de projetos se preocupou com isto ao escolhê-las.

Em geral, os analistas programadores Web ficaram responsáveis pelas práticas de UCD bem como as entrevistas e análise dos usuários, e criação de protótipos. Também ficaram responsáveis pela definição da arquitetura, implementação do software Web e criação e execução de testes unitários, conforme

as histórias fossem sendo implementadas. Por fim, realizaram testes de usabilidade que foram executados em iterações posteriores do processo ágil

A equipe TOTVS ficou responsável pela criação e manutenção de objetos e rotinas relacionadas ao Banco de Dados, a implantação, manutenção e customização do sistema ERP, a retenção e compartilhamento de conhecimento sobre as regras de negócio pertinentes a organização. Nenhuma prática de UCD foi aplicada ao sistema ERP da TOTVS, já que apenas se tratava de uma implantação de um sistema pronto, previamente modelado ao negócio e com alto grau de dificuldade em customizações conforme observações dos próprios especialistas no ERP. O gerente de projetos guiou a equipe quanto às práticas ágeis, mantendo reuniões diárias, acompanhando a interação entre a equipe e os usuários, e priorizando tarefas, por ser a pessoa que mais retém conhecimento sobre o negócio da organização. Para a equipe, o gerente de projetos ficou caracterizado como a “grande figura”.

Não foi definido um papel de treinador, conforme previsto pelo XP. Os participantes deveriam ser autossuficientes e independentes, com capacidade de perceber seus próprios desvios e fazer as correções adequadas.

Nesta mesma reunião, foram decididos detalhes prévios de implementação da arquitetura, como por exemplo, a escolha da linguagem C#, *Webservices* e JavaScript para desenvolvimento do sistema Web, de forma a unificar o conhecimento da equipe.

O gerente de projetos e a organização como um todo estavam dispostos a perder o tempo necessário para a fase exploratória. Mesmo assim, os diretores gostariam de receber algum *feedback* em curto prazo, sendo um conjunto de protótipos, ou uma pequena seção funcional do sistema. A primeira iteração não se iniciou até que toda equipe estivesse suficientemente confiante de que todos os artefatos gerados levariam a uma primeira e boa versão do sistema.

Para o acompanhamento das atividades diárias de cada membro da equipe e conforme previsto pelas práticas do SCRUM, foram agendadas reuniões diárias com duração de 15 minutos e às 13:30, que deveriam ser feitas de pé. Cada participante daria uma visão geral e resumida do que fez no dia anterior e o que fará/faz no dia corrente, para alinhamento com toda equipe.

Artefatos Gerados

Pela quantidade prevista de atividades para o Ciclo Zero a equipe aplicou a visão do *Burndown Chart,* conforme previsto por alguns processos de desenvolvimento ágil. Com isso, a TI entregaria uma visão abrangente de quais atividades estariam sendo realizadas pela equipe à alta diretoria, e estimaria o custo desta como um todo. Como já esperado, o Ciclo Zero deveria ser tratado como qualquer outra subsequente iteração.

O *Burndown Chart* nada mais é que um gráfico com todas as atividades de um ciclo ágil, incluindo uma linha diagonal como meta no intervalo de tempo x tarefa.

Para compor as atividades e prazos para o *Burndown Chart* durante a primeira reunião entre a equipe, o gerente de projetos solicitou uma estimativa do tempo gasto para realizar cada tarefa prevista para o Ciclo Zero.

Após a reunião de planejamento, o gerente de projetos juntamente com os analistas criou um *Burndown Chart* do Ciclo Zero.

Considerações

Durante todas as etapas do Ciclo Zero, o gerente de projetos, percebendo a necessidade de mais membros na equipe, tratou de encontrar novos analistas programadores TOTVS e Web, para minimizar gaps em conhecimentos técnicos dos membros existentes.

Durante esta primeira etapa, diversas reuniões foram agendadas entre a equipe, o que custou cerca de 1 semana de trabalho.

Após alguns meses, as reuniões diárias perderam forças e foram reestabelecidas com a ocorrência de 2 vezes por semana e com maior duração.

6.3.2. 2ª Etapa – Entrevistas para Análise de Negócio

Diversas entrevistas formais foram realizadas com os colaboradores e usuários da empresa, incluindo: Assistentes Comerciais, Coordenadore(a)s de Vendas, Gerentes de Vendas, Representantes de Vendas e Diretores. O principal objetivo foi ampliar o conhecimento da equipe quanto aos processos de negócio da empresa.

A entrevista foi realizada dentro de uma sala de reuniões, com quatro participantes, o entrevistador (gerente de projeto), dois analistas de sistemas e o

colaborador/usuário. O entrevistador seguiu com os questionamentos enquanto outro analista realizava as anotações em um computador. O segundo analista escutava e interrompia quando encontrava algum *gap* que desse abertura para maior detalhamento do negócio.

Todos os comentários foram documentados para futuramente disponibilizar a toda equipe, incluindo os analistas TOTVS que não participaram das reuniões.

Para adquirir mais conhecimento sobre os usuários e localizar aqueles que participariam das próximas etapas da análise, foi preparado um questionário com o objetivo de identificar o perfil, dados etários, *know-how*, conhecimento tecnológico, satisfações e frustrações de um modo geral. Este questionário foi formulado pela própria equipe durante uma rápida reunião antes que iniciasse esta etapa. Outro dado documentado foi a duração das entrevistas, com o objetivo de avaliar o custo total despendido durante todas as etapas para a fase de iniciação do projeto. Desta forma, as perguntas propostas para o questionário foram:

- Qual seu nome e idade? (a pergunta referente a idade foi “auto” respondida pela equipe de TI, para evitar possíveis constrangimentos)
- Qual seu tempo de relacionamento com a empresa?
- Qual seu cargo?
- À quanto tempo está no cargo atual?
- Qual sua formação acadêmica?
- Tem habilidades com computador?
- Tem habilidades com internet?
- Tem habilidades com manuseio celular e acesso à internet?
- Em sua função, utiliza computador e internet? Com que frequência?
- Quanto ao sistema atual da empresa, quais suas satisfações e frustrações? Se pudesse ser diferente, como seria?

A pesquisa dos usuários resultou em uma grande quantidade de informações, já que 30 usuários foram entrevistados. Foram realizadas entrevistas com colaboradores de diversos departamentos e sua distribuição seguiu o plano abaixo:

- Cerca de 80% do departamento comercial (assistentes, coordenadores e gerentes);
- Cerca de 5% do quadro de representantes de vendas;
- Cerca de 20% do departamento financeiro (gerentes);

- Cerca de 20% do departamento de cobrança (gerentes);
- Cerca de 60% do departamento de amostra (assistentes);
- Cerca de 20% do departamento de faturamento (assistentes e gerentes);
- Dois diretores (um do comercial e um de logística, estoque e chão de fábrica).

O foco foi dado ao departamento comercial, pois as rotinas implementadas no ambiente Web deveriam atender as necessidades de usuários que realizavam pedidos, consultavam disponibilidade para venda e consultavam produtos, ou seja, rotinas que fazem parte do contexto comercial de vendas. Alguns participantes do departamento comercial se colocaram no lugar de representantes de vendas, para aumentar as amostragens para a análise deste perfil de usuário. O representante de venda é um importante usuário do sistema, pois este o utilizará externamente, diferente dos demais usuários que estão próximos à equipe de TI, onde o suporte é mais rápido.

Por tratar apenas de questões de negócio, não foram incluídos no quadro de entrevistas de negócio os clientes, que também são usuários finais do sistema. No entanto, os questionários foram aplicados a três representantes que por sua vez responderam pelos seus principais clientes.

A partir de uma reunião entre os analistas e o gerente de projetos, as informações foram organizadas por idade e idade média, tempo de relacionamento com a empresa, classificação de *know-how* e *know-how* médio com valores que variam de 1 a 5, frase emblemática, frustrações / dificuldades, satisfações, melhorias emergentes, rotinas diárias, quais valores o usuário agrega como um usuário final para o sistema e duração total das entrevistas. Todas estas informações foram exemplificadas em personas, duas das quais podem ser vistas na seção de Artefatos Gerados.

Após a etapa de documentação, foram selecionados os usuários mais importantes para o negócio e aqueles que agregam maior valor como usuários chave. É importante salientar que as características dos personas foram escolhidas com base naquilo que mais importa para o negócio, ou seja, as perguntas foram motivadas pelo caráter profissional e não pessoal, já que o sistema é em boa parte direcionada aos colaboradores internos da empresa.

Artefatos Gerados

Com os resultados obtidos nos questionários foi possível construir cinco Personas, um para cada perfil: SAC, Coordenador de Vendas, Diretor, Representante de Vendas e Cliente. Os questionários e entrevistas foram arquivados em um documento de descobertas.

De acordo com as informações obtidas neste estudo, organizadas em um documento de descobertas, algumas personas foram identificadas. Dois exemplos delas são ilustradas logo abaixo:

PERSONA: PRIMÁRIA
NOME: JULIANA
IDADE: 29 ANOS
SEXO: FEMININO
TEMPO NA EMPRESA: 10 ANOS
CARGO: COORDENADORA DE VENDAS
***KNOW-HOW*: (4) ALTO**

FRASE EMBLEMÁTICA	"O atual sistema da empresa me deixa burra"
FORMAÇÃO	Pedagogia
PERFIL	Juliana tem facilidades em utilizar computador, mas utiliza pouco fora do ambiente empresarial. Gosta de acessar as redes sociais e é uma pessoa curiosa. Tem bons conhecimentos em internet, mas navega pouco, principalmente durante a noite e nos finais de semana. Tem bons conhecimentos em ferramentas como o Microsoft Outlook na sua versão Web, o website da prefeitura de SP e o website da Receita Federal, os quais acessa com muita frequência.
FRUSTRAÇÃO	Juliana gasta um tempo exagerado nas rotinas diárias, precisa trabalhar até tarde para atender toda demanda.
ROTINA	• Realiza o cadastro e dá suporte aos representantes; • Realiza o faturamento de pedidos da fábrica; • Digita pedidos de venda especiais; • Digita pedidos de amostra para os

	representantes; • Faz liberação de pedidos digitados pelos representantes; • Solicita limite e liberação de crédito ao gerente financeiro.

Tabela 2 - Persona da usuária Juliana / F / 29 (P2)

PERSONA: SECUNDÁRIA

NOME: MARIANA

IDADE: 33 ANOS

SEXO: FEMININO

TEMPO NA EMPRESA: 6 ANOS

CARGO: REPRESENTANTE DE VENDAS

***KNOW-HOW*: (3) MÉDIO**

FRASE EMBLEMÁTICA	"O maior problema para o representante é a consulta da disponibilidade e o acesso a um catálogo de produtos"
FORMAÇÃO	Jornalismo
PERFIL	Juliana utiliza a internet com bastante frequência, também utiliza *tablets* e *smartphones*, está por dentro de tudo que é novidade. Leva bastante trabalho pra casa e vive viajando.
FRUSTRAÇÃO	Juliana sente falta de um bom catálogo virtual para apresentar ao cliente os produtos com disponibilidade para venda, fotos e ficha técnica.
ROTINA	• Consulta disponibilidade do estoque; • Solicita a um auxiliar fotos de produtos quando não tem em um catálogo impresso; • Verifica o histórico de pedidos do cliente; • Baixa notas fiscais e duplicatas para o cliente.

Tabela 3 - Persona da usuária Mariana / F / 33 (P4)

Considerações

Durante as entrevistas, todos os comentários pertinentes ao contexto da entrevista foram anotados pelo analista que fazia o uso de um computador portátil. Perceptivelmente, alguns entrevistados se sentiram acuados com as anotações.

Com base em conversas informais pelos membros da equipe, algumas suposições foram feitas, por exemplo, (A1): “Talvez os usuários estejam amedrontados, achando que o novo sistema irá tirar o trabalho deles, ou talvez seus perfis e/ou rotinas diárias seriam utilizadas em algum momento contra eles mesmos”.

Foram levantadas muitas regras de negócio, que por sua vez gerariam inúmeros requisitos funcionais de sistema. A equipe prontamente percebeu o tempo e a dificuldade em implementar todas elas.

Faltou um pouco de *expertise* do entrevistador, os usuários muitas vezes se desviaram do foco da entrevista, o que as estendeu muito tempo. Em média, as entrevistas levaram de 30 a 40 minutos e todas as entrevistas levaram cinco dias para serem finalizadas.

Em geral, os usuários estavam satisfeitos com o sistema atual, no entanto, a usabilidade e o atraso tecnológico os levaram a solicitar melhorias, melhorias estas que não poderiam ser atendidas pela limitação tecnológica de uma linguagem de programação como o caso do Cobol.

Esta foi uma das etapas que mais custou tempo da equipe, foram pouco mais de duas semanas de entrevistas, análise, documentação e construção de personas.

6.3.3. 3ª Etapa – Investigação Contextual

Com o objetivo geral de coletar dados detalhados sobre como as pessoas trabalham e quais suas rotinas no atual sistema da empresa, foi aplicada a técnica da investigação contextual.

Para considerar um grupo de usuários focais para a investigação contextual, foram utilizados resultados de perguntas realizadas durante a 1ª etapa de entrevistas, como: “qual pessoa você considera estar diretamente relacionada a seu trabalho, ou seja, quem lhe completa em suas rotinas diárias?”, neste ponto, os entrevistados responderam sobre pessoas que fazem parte de suas rotinas diárias, como exemplo: “O fulano, a partir do recebimento de um telefonema ou e-mail, dá entrada a um pedido de devolução para uma mercadoria com defeito. Eu recebo e analiso este pedido, confiro os dados da nota fiscal, emito uma nota de devolução e encaminho para o ciclano das contas a pagar para ele efetue o estorno do valor”. Esta análise foi realizada, já que por exemplo, durante a emissão de um pedido de venda, é necessário o envolvimento de diversas pessoas, uma que inclui o pedido,

outra que aprova comercialmente, outra que aprova financeiramente e etc. Algumas destas rotinas também seriam realizadas na ferramenta Web, a qual era alvo do estudo de usabilidade.

Pela visível falta de tempo dos usuários que já executavam frequentemente tarefas de diversas áreas de negócio, foi aplicada esta técnica, pois boa parte da investigação é baseada na observação dos usuários, sem demandar muito tempo deles.

Durante a execução de suas rotinas diárias, dois entrevistadores multidisciplinares sentaram ao lado de alguns usuários chave, identificados após as entrevistas realizadas na 1ª etapa. Nesta mesma 1ª etapa de entrevistas, os entrevistadores comentaram sobre a necessidade de futuras entrevistas para recolher dados das rotinas diárias e ainda comentaram que para tanto eles seriam ocasionalmente interrompidos. Foi dito aos usuários que a entrevista forneceria ao projeto dados importantes para mapeamento das interfaces do sistema, conhecimento das suas rotinas de trabalho, inter-relacionamento e compartilhamento de tarefas com os colegas e auxílio no mapeamento das regras de negócio pertinentes ao cenário estudado.

Antes de iniciar, o entrevistador solicitou ao usuário a execução de algumas tarefas específicas que estavam pendentes (por exemplo, documentos que estavam em cima da mesa e precisavam ser baixados para o sistema), como meio de criar um foco para a observação. Algumas tarefas executadas especificamente foram: criar um pedido de venda, alterar os produtos e quantidades de um pedido, consultar um relatório em busca da cidade e transportadora de um cliente.

Abaixo é apresentado o resultado da entrevista e observação de uma assistente de vendas responsável por pedidos de uma das empresas do grupo, Estes artefatos foram nomeados como Cenários (1 .. N):

```
NOME DA ENTREVISTADA: Suzana
DURAÇÃO: 45 minutos
NOME DO CENÁRIO: Cenário 4 - Incluir Pedido de Venda

Descrição:
Quando se inicia o pedido, a primeira coisa feita pela Suzana é
identificar o tipo do pedido de venda. Ela utiliza a seguinte
nomenclatura: para pedido de pronta entrega “PE”, para pedidos de
programação “PR”, para duplicação de pedidos “DP”.
Em seguida, Suzana escolhe um cliente. Para evitar que o mesmo
cliente receba novamente o mesmo produto previamente devolvido, ela
```

define se o produto é de devolução. Caso o frete seja pago pelo cliente, é definida a transportadora do próprio cliente como FOB, se há transportadora de redespacho a Suzana também precisa defini-la na emissão do pedido.
Em seguida, ela escolhe o produto pelo código ou consulta seu nome pela tecla F5 pressionada sobre o próprio campo de código do produto, e depois define qual a quantidade, com base no QMM (quantidade por caixa), se a quantidade solicitada for inferior ao tamanho do QMM, este pedido é "especial" e a digitação deve ser feita diretamente para a fábrica, depois define o preço com desconto se for o caso. Ao pressionar F5 uma pequena caixa com campo de descrição surge na tela com cor de fundo vermelho, a usuária digita qualquer texto e pressiona *ENTER* para ver todos produtos disponíveis com aquele texto digitado.
Em seguida, Suzana digita o código da condição de pagamento. Logo após isto, ela tenta digitar a observação do pedido, tem problemas, pois, o campo é limitado a 30 caracteres, então ela começa a abreviar cada palavra digitada "FAT COM DT BASE 11-11-2013".
Em seguida, Suzana finaliza o pedido enviando-o para a análise.
Mesmo com as limitações técnicas que o sistema fornece Suzana não tem qualquer dificuldade na interação e com algumas dezenas de utilizações da tecla *TAB*, Suzana finaliza o pedido.

Artefatos Gerados

Durante as observações, o 2º entrevistador resumiu boa parte das rotinas executadas para aquela tarefa previamente estabelecida de forma a construir os documentos para os cenários.

Regras de negócio que não eram de conhecimento do entrevistador foram anotadas e compartilhadas com a equipe através de um documento nomeado como "Resultados da Investigação Contextual". Toda equipe conhecia a nomenclatura e sabia onde encontrar o documento, pois durante 20 minutos e na metade do expediente, reuniões diárias eram realizadas para alinhar o conhecimento da equipe. Diagramas de BPMN também foram construídos baseados nos cenários e interações entre outros membros da equipe comercial, a fim de fornecer melhores esclarecimentos sobre as regras de negócio e fluxos de processo da empresa.

Considerações

Durante uma das investigações, foi observado o papel do representante recebendo o treinamento de sua coordenadora de vendas, de como realizar pedidos no sistema atual. Esta investigação em específico rendeu bons resultados, pois além de observar as dificuldades do representante em realizar a tarefa, também identificou a visão da primeira impressão que o representante teve sobre a

compreensão das rotinas da coordenadora e sua assistente que frequentemente era incomodada para dar mais detalhes de negócio e apoio a identificação de recursos do sistema.

Em alguns momentos, o usuário foi interrompido pelo observador. Na maioria das vezes, quando este executava uma tarefa muito rapidamente, deixando o observador confuso com a ação.

A equipe sentiu falta da validação de alguns cenários pós-documentados com o próprio usuário. Durante a reunião para compartilhamento do conhecimento adquirido, o gerente de projetos, munido de muito conhecimento sobre o negócio, observou *gaps* nas anotações. Desta forma, foi percebida que durante a prática de investigação contextual, era imprescindível a participação de um membro com bastante conhecimento de negócio, pelo menos para aquele cenário específico observado. Por exemplo, um analista com grande conhecimento em contabilidade investigou o usuário durante a execução de uma tarefa pertinente ao âmbito vendas, deste modo, o analista observador ficou confuso com algumas tarefas como a falta de conhecimento em frete, transportadora e redespacho.

Nenhum entrevistado apresentou nervosismo e incomodo durante a investigação. Aparentemente isso se deve ao fato da interação que a equipe de investigadores teve com os usuários antes da investigação contextual.

A aplicação da técnica não pôde ser executada no ambiente do cliente, primeiro por este atualmente não fazer uso do sistema atual da empresa, segundo pela posição geográfica pouco favorável da maioria deles e pouco tempo para deslocar a equipe de entrevistadores até seus ambientes de trabalho.

Como a investigação foi aplicada a um grupo menor de usuários, a tarefa durou pouco mais de 1 semana para ser concluída. Um conjunto importante de observações sobre o negócio e as rotinas de trabalho apoiaram as fases posteriores.

6.3.4. 4ª Etapa – Levantamento de Histórias de Usuários

As histórias de usuários descrevem características ou funcionalidades que os usuários gostariam de encontrar no sistema. Conforme práticas estudadas neste trabalho, e para adequar o projeto às abordagens de UCD e processos ágeis, mais especificamente a metodologia XP, não só as histórias devem descrever características ou funcionalidades de negócio bem como aspectos de design. Para

tanto, a equipe orientou os participantes a pensar como designers e considerar nas histórias detalhes da interface, mesmo sendo difícil para eles pensar nestes aspectos, já que o sistema legado não atinge padrões satisfatórios de usabilidade e design interativo. Algumas perguntas e exemplos foram planejados para extrair o máximo de informações dos usuários e oferecer mais detalhes a estes sobre o que caracteriza o design de interface, são elas:

- “Caso ocorra um erro na execução de alguma rotina, como gostaria que a mensagem fosse exibida na tela?”;
- “Escreva como história, por exemplo: Quero poder utilizar o sistema em um celular ou *tablet*.”;
- “Escreva como história, por exemplo: Quando uma ação demorar em finalizar, quero que algo me diga que ela ainda está em execução”;
- “Escreva como história, por exemplo: Quero campos para entrada de textos largos, com fontes grandes e bastante espaço para digitação”;
- “Pensando no sistema como um todo, qual cor de fundo, botões e textos vocês particularmente gostam?”;
- “Se fosse pensar nas empresas Corttex e Fatex, para vocês qual cor representa a marca? E qual você gostaria de ver no website?”.

Um analista em algum momento disse para o participante: “Usem aqueles sites que vocês mais gostam como referência. Procurem no Google uma imagem ou um site que retrate aquilo que vocês querem. Como estamos tratando do acesso externo por representantes e clientes, e num cenário de vendas, sites de comércio eletrônico podem ser boas referências”. Estas perguntas podem parecer técnicas e dificilmente compreendidas por leigos, no entanto, de acordo com as entrevistas realizadas, definições de personas e investigação contextual com base em observações, os usuários eram bons na interação com computadores e internet e a equipe ficou segura em fazer estes tipos de questionamentos.

A fim de garantir a participação de vários usuários e gastar o mínimo de tempo dos analistas e gerente com orientações, foi agendada uma reunião com um grupo de usuários. Nesta reunião, foi apresentada a ferramenta que seria utilizada para administrar as histórias, a ferramenta EasyBacklog [EASYBACKLOG].

EasyBacklog é uma ferramenta que foi desenvolvida por Mattew O’Riordan em Janeiro de 2011, com objetivo de auxiliar na gestão de *backlog* por equipes

ágeis. Oferece capacidades para versionamento, relatórios sobre mudanças durante todo processo de entrega e gerenciamento diferenciado de pequenas entregas [EASYBACKLOG Blog]. De acordo com o próprio fundador, a falta de ferramentas para auxiliar a equipe na fase inicial de escopo e estimativa foi um dos motivos do desenvolvimento do sistema que está disponível totalmente através da Web e gratuitamente. A ferramenta garante maior contato com o usuário, pois os incentiva a alterar o escopo à medida que o projeto avança, permitindo também a gestão de custo e tempo de forma transparente.

Após sete dias de uso da ferramenta, a equipe conseguiu capturar 72 histórias, sendo que muitas delas precisariam passar por análise dos programadores. Abaixo são apresentadas algumas observações quanto aos textos digitados pelos usuários:

- Necessária à divisão de grandes histórias em histórias menores, os usuários colocaram diversos detalhes de negócio e sugestões de funcionalidades em uma mesma ficha;
- A equipe não conseguiu capturar tantos detalhes de interface quanto gostariam;
- Alguns usuários não estavam 100% comprometidos com o preenchimento das histórias, alguns separaram tempo hábil para se dedicar a tarefa e outros só acessaram a ferramenta, mas não preencheram nenhuma história durante o prazo estabelecido;

Na figura 6 é ilustrada a tela do EasyBacklog com algumas histórias já preenchidas para o levantamento do escopo do sistema em projeto CorttexNet.

A fim de facilitar a inclusão dos usuários e apoiar um *brainstorm* à distância, foi liberado o acesso à visualização de todas as histórias digitadas por outros usuários, desta forma, estes liam os textos digitados por outros e assim melhoravam ou geravam novas ideias;

Todos os usuários sabiam em tempo real quais outros usuários estavam utilizando a ferramenta, aumentando a interação entre eles, e muitas vezes um chamava o colega ao lado para discutir uma história incluída.

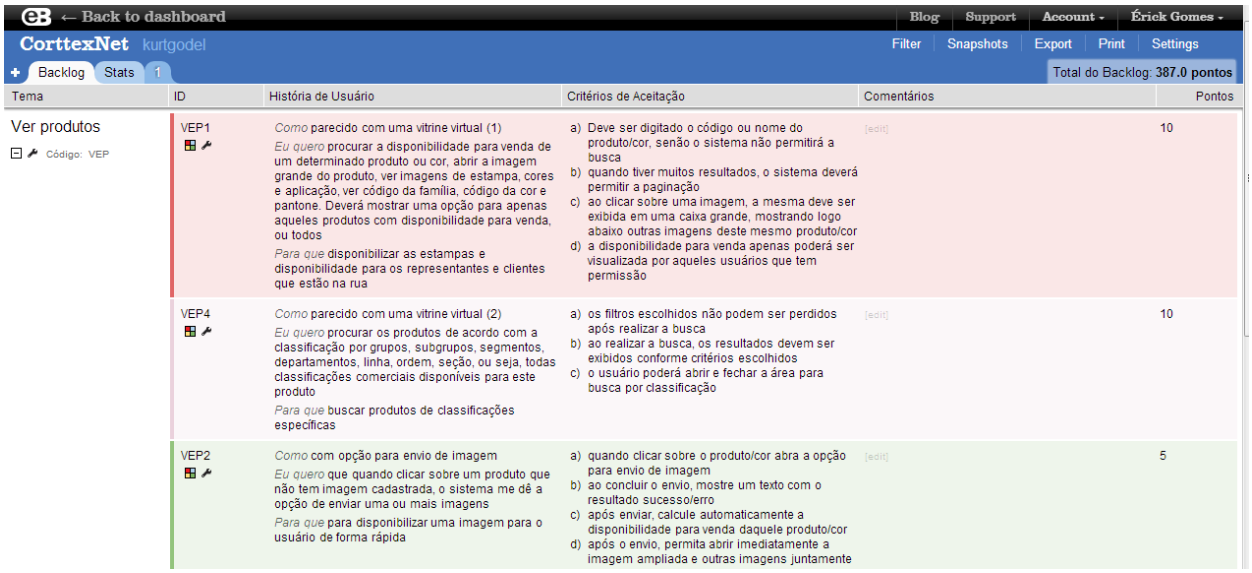


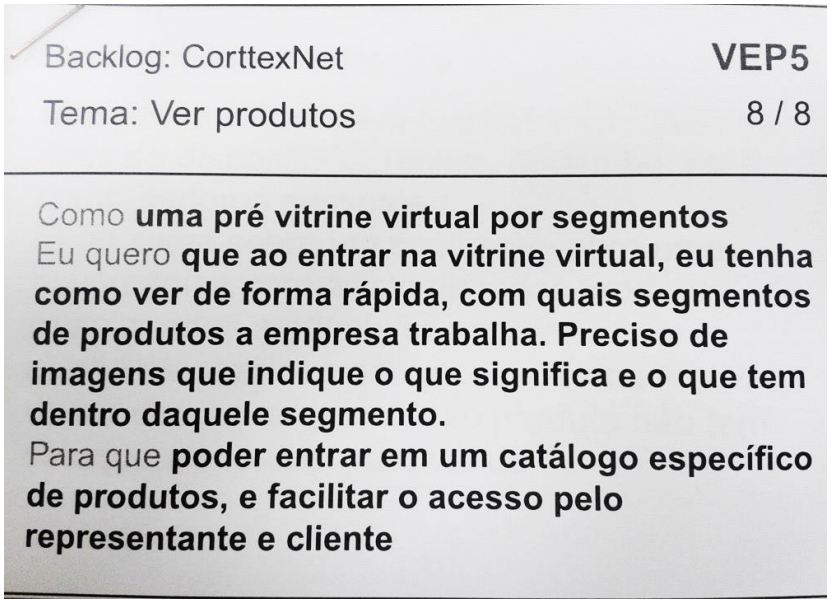
Figura 7 - Histórias de usuários cadastradas no EasyBacklog

Artefatos Gerados

Além de disponibilizar a todos os membros da equipe a coleção de histórias preenchidas através do sistema EasyBacklog, também foram criadas pequenas fichas impressas, incluindo os critérios de aceitação, para que as utilizassem como artefatos de referência para reuniões. Exemplos de fichas podem ser observados nas figuras 7, 8 e 9. A figura 7 ilustra uma ficha de requisito exportado pela ferramenta EasyBacklog que apresenta a necessidade de desenvolvimento de uma vitrine virtual para visualização dos produtos da empresa. A figura 6 ilustra a tela de cadastro de um novo cliente chamado de "*prospect*". Ambos requisitos foram solicitados pelos usuários do sistema legado. A figura 9 ilustra alguns critérios de aceitação para o requisito da vitrine virtual que foram definidos em um trabalho conjunto entre o analista e o usuário.

Os critérios de aceitação nada mais são do que testes aplicados ao sistema ou aos protótipos para estabelecer que um requisito foi implementado corretamente.

Todas as histórias aprovadas na análise foram reclassificadas na granularidade de requisitos funcionais e não-funcionais e documentadas em um *product backlog*. Por sua vez, o *product backlog* serviu como referência para fases posteriores.

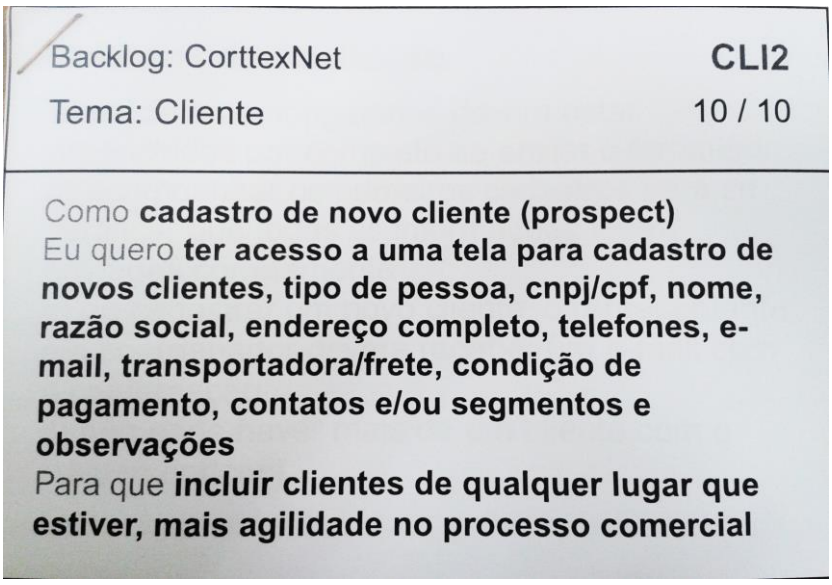


Backlog: CorttexNet VEP5

Tema: Ver produtos 8 / 8

Como **uma pré vitrine virtual por segmentos**
Eu quero **que ao entrar na vitrine virtual, eu tenha como ver de forma rápida, com quais segmentos de produtos a empresa trabalha. Preciso de imagens que indique o que significa e o que tem dentro daquele segmento.**
Para que **poder entrar em um catálogo específico de produtos, e facilitar o acesso pelo representante e cliente**

Figura 8 - Ficha de requisito da Vitrine Virtual

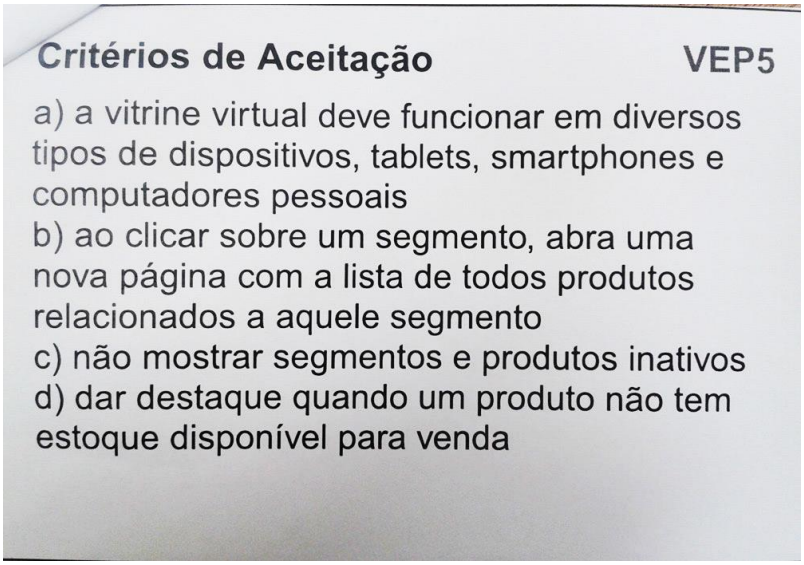


Backlog: CorttexNet CLI2

Tema: Cliente 10 / 10

Como **cadastro de novo cliente (prospect)**
Eu quero **ter acesso a uma tela para cadastro de novos clientes, tipo de pessoa, cnpj/cpf, nome, razão social, endereço completo, telefones, e-mail, transportadora/frete, condição de pagamento, contatos e/ou segmentos e observações**
Para que **incluir clientes de qualquer lugar que estiver, mais agilidade no processo comercial**

Figura 9 - Ficha de requisito do cadastro de cliente "*prospect*"



Critérios de Aceitação **VEP5**

a) a vitrine virtual deve funcionar em diversos tipos de dispositivos, tablets, smartphones e computadores pessoais
b) ao clicar sobre um segmento, abra uma nova página com a lista de todos produtos relacionados a aquele segmento
c) não mostrar segmentos e produtos inativos
d) dar destaque quando um produto não tem estoque disponível para venda

Figura 10 - Ficha de critérios e aceitação para requisito de Vitrine Virtual

Considerações

Durante a realização de perguntas técnicas sobre interface, design e navegação em websites, percebeu-se que nenhum usuário ficou confuso, todos eles pareciam saber do que a equipe estava falando e participaram ativamente com sugestões de melhorias, preferências pessoais, sistemas conhecidos e websites de referência.

O EasyBacklog demonstrou ser uma ferramenta com grande potencial, pois atendeu o objetivo geral de levantamento de requisitos e os usuários não apresentaram grandes dificuldades em usá-lo. A ferramenta também conseguiu gerenciar os acessos simultâneos por diversos usuários, porém, a falta de suporte completo à língua portuguesa, dificultou a tarefa de navegação intuitiva de alguns usuários, já que o objetivo da implantação foi capturar o máximo de histórias sem precisar oferecer demasiados treinamentos, o que demandaria muito tempo da equipe de TI.

A falta de comprometimento dos usuários, de acordo com o gerente e analistas, pode se dar ao fato destes não terem o apoio e intervenção da alta diretoria durante a apresentação da ferramenta, definição da tarefa e estabelecimento do prazo.

De acordo com a dificuldade de contato com os clientes e uma amostragem maior de representantes, os colaboradores internos fizeram o papel destes usuários, colocando-se no lugar deles para definição das histórias.

Após a criação de cada protótipo, foi anotado nas fichas das histórias o código do protótipo construído diretamente relacionado a ela. Uma história pode ter relação com um ou mais protótipos.

Esta etapa durou aproximadamente 2 semanas para orientação dos participantes, definição de um conjunto chave de histórias e documentação, no entanto, pode-se dizer que a processo continuou em paralelo com outros, já que muitas histórias foram escritas pelos participantes com o decorrer do projeto.

6.3.5. 5ª Etapa – Reunião sobre Decisão Arquitetural

Mais uma atividade proposta para o Ciclo Zero, agora com foco nas práticas de XP, é a reunião sobre decisões arquiteturais. Nesta reunião participaram dois analistas Web, um analista TOTVS e um gerente de projetos.

O objetivo foi discutir e alinhar com a equipe qual modelo arquitetural seria utilizado para a implementação do sistema de plataforma Web. Na figura 8 é ilustrado o modelo arquitetural adotado pela equipe.

Conforme já esperado, este sistema se comunicaria diretamente com o ERP Protheus da TOTVS, portanto, a arquitetura também deveria se refletir nesta integração. Toda a integração com o ERP e os módulos do sistema foi baseada em requisições via *Webservices*, que utiliza o protocolo SOAP para a comunicação.

Os pedidos de venda seriam incluídos diretamente pela ferramenta Web, desta forma, um dos fatores que preocupou a equipe foi a definição de um modelo de segurança que impedisse o acesso externo, e este também deveria se refletir na arquitetura desenhada e implementada pelos analistas.

A equipe se preocupou em desenhar uma solução simples, segura, utilizando novas tecnologias e aplicando as boas práticas do mercado, como por exemplo, a autenticação de URL com OAUTH 2.0 (Especificação na RFC 6749) para acesso aos *Webservices*, e disponibilização destes serviços para sistemas de externos que no futuro possam se comunicar com o sistema de vendas. O modelo arquitetural desenvolvido pode ser compreendido com base na figura 6.

E qual a usabilidade no contexto de arquitetura da solução e requisições de serviços via web? A arquitetura planejada para as páginas web foi baseada nas requisições via AJAX e utilizando o framework jQuery, com isso, foi possível fornecer ao usuário uma experiência de uso aprimorada já que as requisições HTTP são

feitas de forma assíncrona, mostrando sempre o *status* do carregamento, sem *refresh* de página (aquela tela branca), com animações e efeitos de redimensionamento e reposicionamento de elementos da página.

Esta composição baseada em *Webservices* e requisições HTTP via AJAX e jQuery é o que define o conceito da Web 2.0. De acordo com os autores Clovis Carneiro e Rida Al Barazi, o AJAX pode aumentar drasticamente a usabilidade da interface do usuário [CARNEIRO e BARAZI].

Figura 11 - Modelo arquitetural proposto

De acordo com a figura 10, o ambiente do usuário foi desenvolvido com as últimas tecnologias disponíveis no mercado, como HTML 5, jQuery, JSON e CSS3. A aplicação foi desenvolvida em ASP.Net e C#, plataforma de desenvolvimento Web proprietária da Microsoft, em composição com JavaScript.

O ERP funciona paralelamente a sistema web, acessando diretamente o banco de dados que é o mesmo acessado pelos *Webservices*, o qual provê dados para os recursos disponibilizados na web.

Considerações

Durante a reunião, a comunicação entre todos analistas foi de extrema importância, pois até detalhes de boas práticas de integração com o ERP foram discutidas, no caso, com a utilização de *WebServices* para acesso ao banco de dados. Pelas boas práticas da TOTVS, aplicações externas não podem acessar diretamente o banco de dados do ERP e para resolver isto, a TOTVS disponibiliza uma série de *WebServices* previamente implementados e manuais de como desenvolvê-los.

Apenas duas reuniões foram suficientes para definição da arquitetura, uma para discussão e outra para apresentação do esboço, o que por fim, custou cerca de 3 dias de trabalho da equipe.

Vale salientar que esta foi a única etapa do proposto Ciclo Zero onde não houve qualquer participação do usuário final, já que trata apenas de detalhes técnicos implícitos a implementação do software. No entanto, a escolha de tecnologias emergentes da Web 2.0 teve o objetivo de garantir a máxima interatividade e usabilidade, como é o caso do leiaute responsivo (auto ajustável) tornando a interface adaptada a diversos dispositivos, ou com o uso também do JQuery para o processamento assíncrono de requisições HTTP. As requisições assíncronas implicam na redução de recarregamentos de página, melhorando assim o tempo de resposta e o *feedback* para o usuário final.

6.3.6. 6ª Etapa – Construção de Protótipos

Nesta fase, diversos protótipos de baixa fidelidade referentes aos principais recursos do sistema foram construídos, como exemplo a tela de emissão de pedidos, a vitrine virtual de produtos e a ficha cadastral do cliente. Para tanto, foi utilizada a ferramenta Pencil da Evolus. Pencil é ferramenta *open source* multiplataforma fácil de instalar e usar, e é direcionada à fácil construção de protótipos e *mockups*. Tem uma interface bastante intuitiva e deixa o resultado final com aparência de desenhos feitos a mão (traços imprecisos), que é um aspecto muito importante, já que a equipe não queria passar neste momento a impressão de um produto já finalizado.

A figura 11 ilustra uma tela do Pencil com uma prévia construção de protótipo da tela de Vitrine Virtual onde são expostos os produtos no formato de uma ferramenta *e-commerce*.

Nas figuras 12, 13 e 14 são ilustrados protótipos de baixa fidelidade construídos com o Pencil, nas quais é possível compreender alguns poucos recursos disponíveis na ferramenta para design de protótipos ou *mockups*.

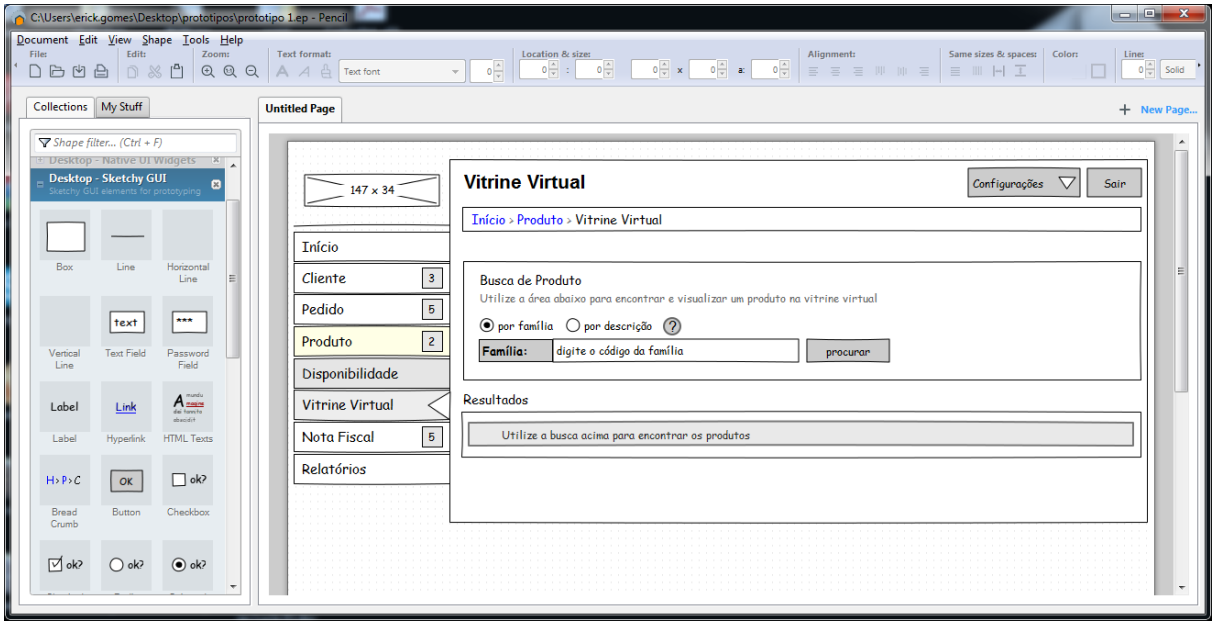


Figura 12 - Tela da ferramenta Pencil

Em todas as telas de construção do protótipo foram incluídos blocos de texto com descritivos das Heurísticas de Nielsen de forma resumida, como meio de fixação e para que o designer sempre se focasse na construção dos protótipos atendendo as características das heurísticas.

Nestes protótipos também foram considerados aspectos de design responsivo, ou seja, que este leiaute atendesse o maior número possível de dispositivos, como: *pcs*, *tablets* e *smartphones*. Na seção de artefatos gerados, a figura 14 ilustra um protótipo da tela de pedido de venda reajustado para um *smartphone*.

Todos os elementos visuais dos protótipos foram criados com base no levantamento de requisitos, sugestões de interface, cenários de uso e compreensão de experiências dos usuários com sistemas da plataforma Web.

Artefatos Gerados

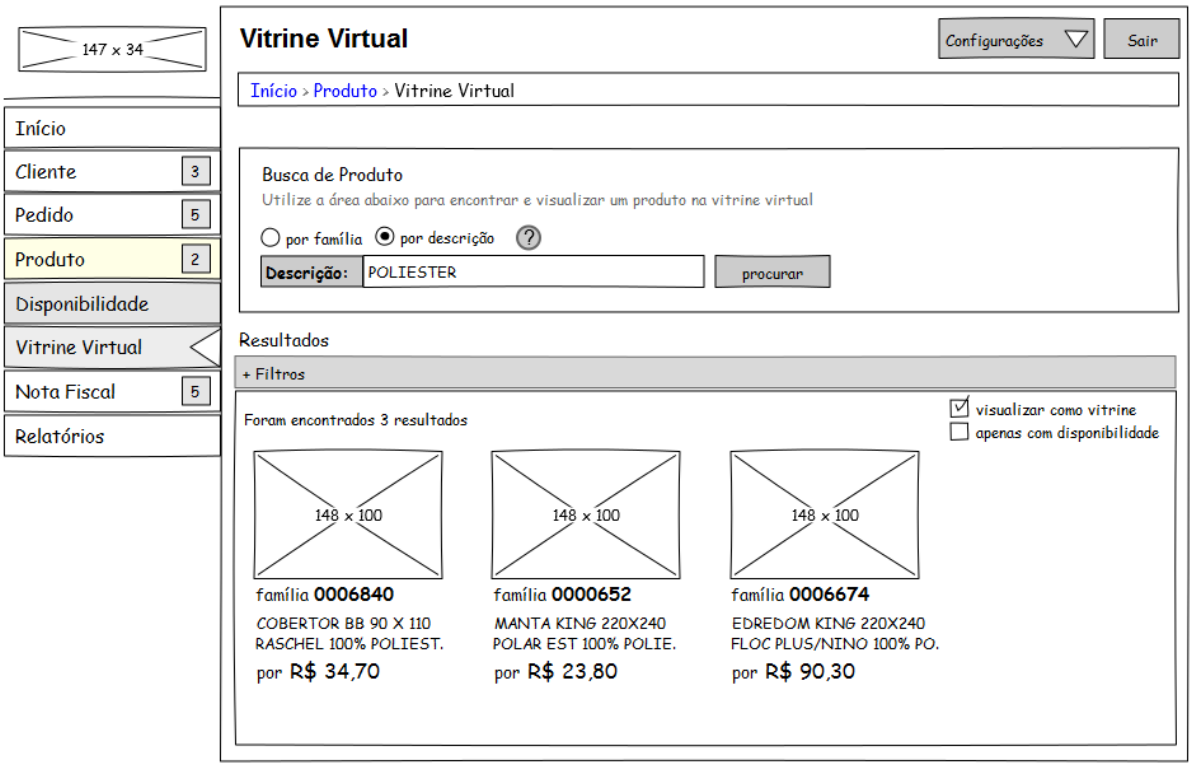


Figura 13 - Protótipo P1 da tela de Vitrine Virtual

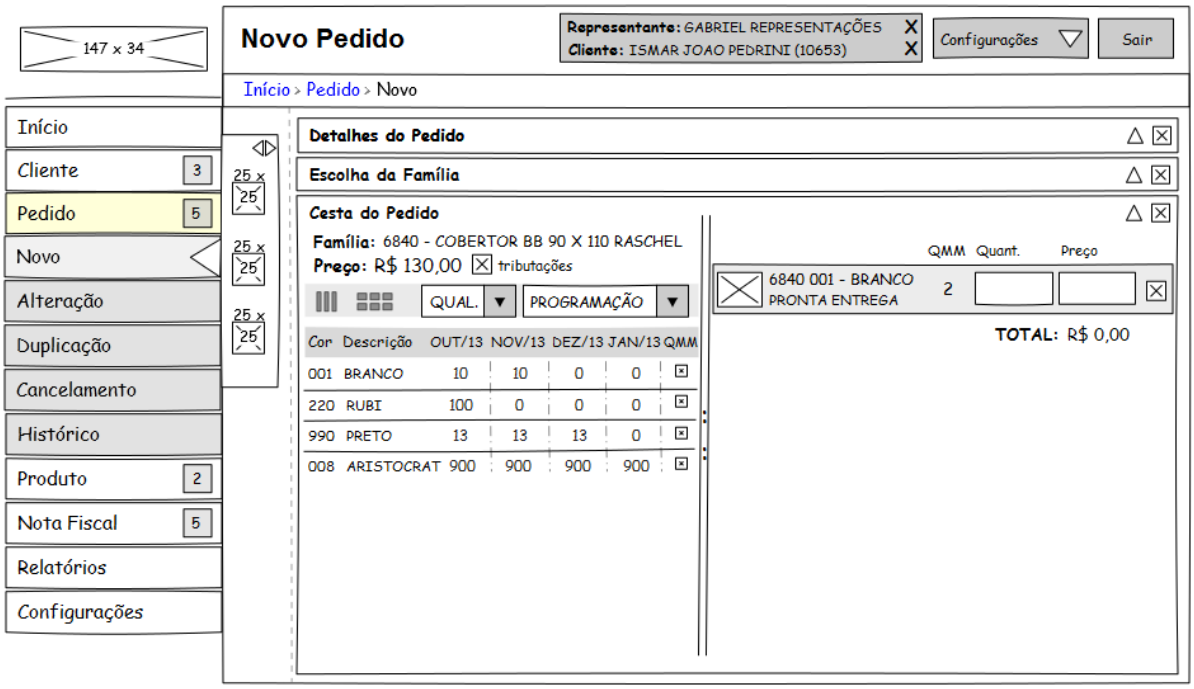


Figura 14 - Protótipo P2 da tela de emissão de pedido – Cesta do Pedido

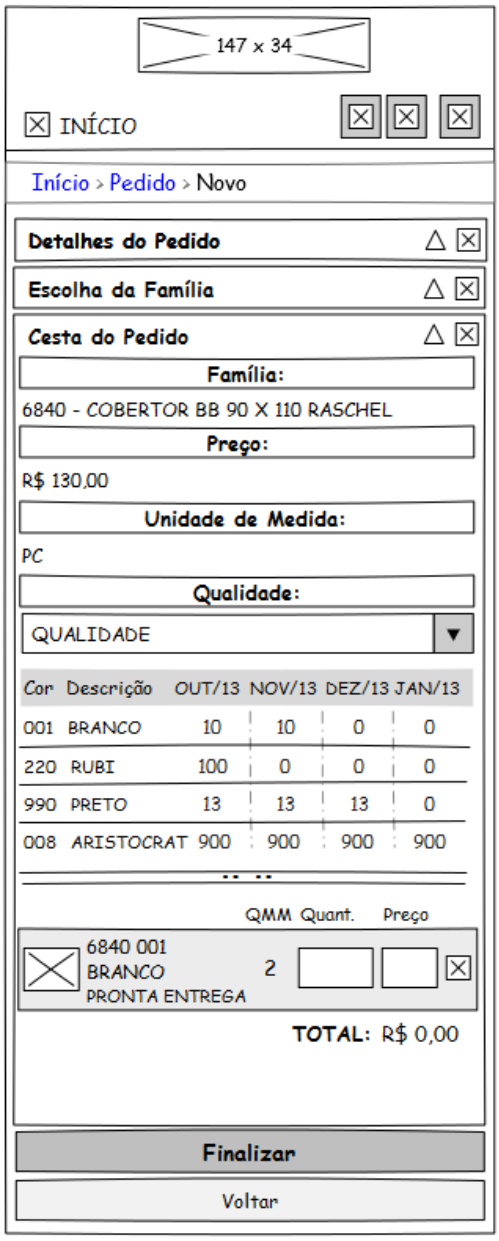


Figura 15 - Protótipo P3 da tela de emissão de pedido para *Smartphone*

Considerações

Não foram construídos os protótipos de todas as telas do sistema, apenas aquelas com maior número de recursos visuais, complexidade de leiaute, e com maior distinção das demais telas quanto a elementos visuais e interatividade. A escolha dos protótipos também foi baseada na priorização dos requisitos provenientes do *product backlog*.

Antes da etapa de Design Participativo a equipe se reuniu para discutir e fazer o primeiro refinamento dos protótipos. Muitas observações foram previamente anotadas e reaplicadas através da refatoração dos protótipos. A refatoração é uma

atividade prevista pela metodologia XP, mas aplicada sobre código fonte, no qual o desenvolvedor analisa o código escrito e faz correções para otimização, inclusão de boas práticas ou até o reescreve completamente caso seja necessário.

Na figura 13 é ilustrada uma importante característica que foi solicitada pelo usuário. Ao lado do *menu* esquerdo, pode ser visualizada uma lista de ícones dispostos verticalmente, que disponibilizam o acesso rápido a todos os relatórios pertinentes à tela em questão, ou seja, na tela de pedido de venda poderiam ficar disponíveis a ficha técnica de um produto, a ficha cadastral de um cliente e/ou a disponibilidade em estoque completa de um determinado artigo. Estes, por exemplo, são recursos que frequentemente um vendedor ou um assistente comercial utilizam durante o processo de emissão de pedido e desta forma, eles não precisariam sair da tela para encontrar estas informações, otimizando seus trabalhos através do ganho de tempo.

Como não foram construídas todas as telas, esta etapa custou da equipe um período menor, de aproximadamente 1 semana.

Para a equipe esta foi uma das principais tarefas e consecutivamente gerando os principais artefatos para o projeto. Houve muita vantagem com a redução de retrabalho e satisfação dos usuários e *stakeholders* pelo que foi gerado posteriormente a partir destes protótipos.

6.3.7. 7ª Etapa – Design Participativo

Neste momento do projeto, foram aplicadas técnicas de UCD mais em específico o Design Participativo para entender as reais necessidades dos usuários e incluí-los como donos do produto. O Design Participativo ocorreu através de reuniões entre os analistas/programadores/designers com os usuários para cooperação e análise conjunta sobre o que foi entendido pela equipe de TI e sobre o que foi desenhado em protótipos.

As abordagens de UCD tiveram como alvo a análise e entendimento dos cenários desenhados em etapas anteriores pela equipe de analistas e designers a fim de apoiar na construção de protótipos de baixa fidelidade. A partir da construção os analistas se reuniram com os usuários para assistirem a navegação destes nos protótipos. Todas as observações foram anotadas no próprio papel, incluindo comentários ou redesenhos de algumas seções do sistema.

Considerando sempre as práticas do XP, mesmo naquelas não intimamente relacionadas a ele, os protótipos após utilização e havendo necessidade foram alterados até atenderem as mudanças propostas por todos participantes.

Artefatos Gerados

A figura 15 ilustra as observações dos participantes e a figura 16 ilustra a correção do protótipo P2 contemplando novos detalhes.

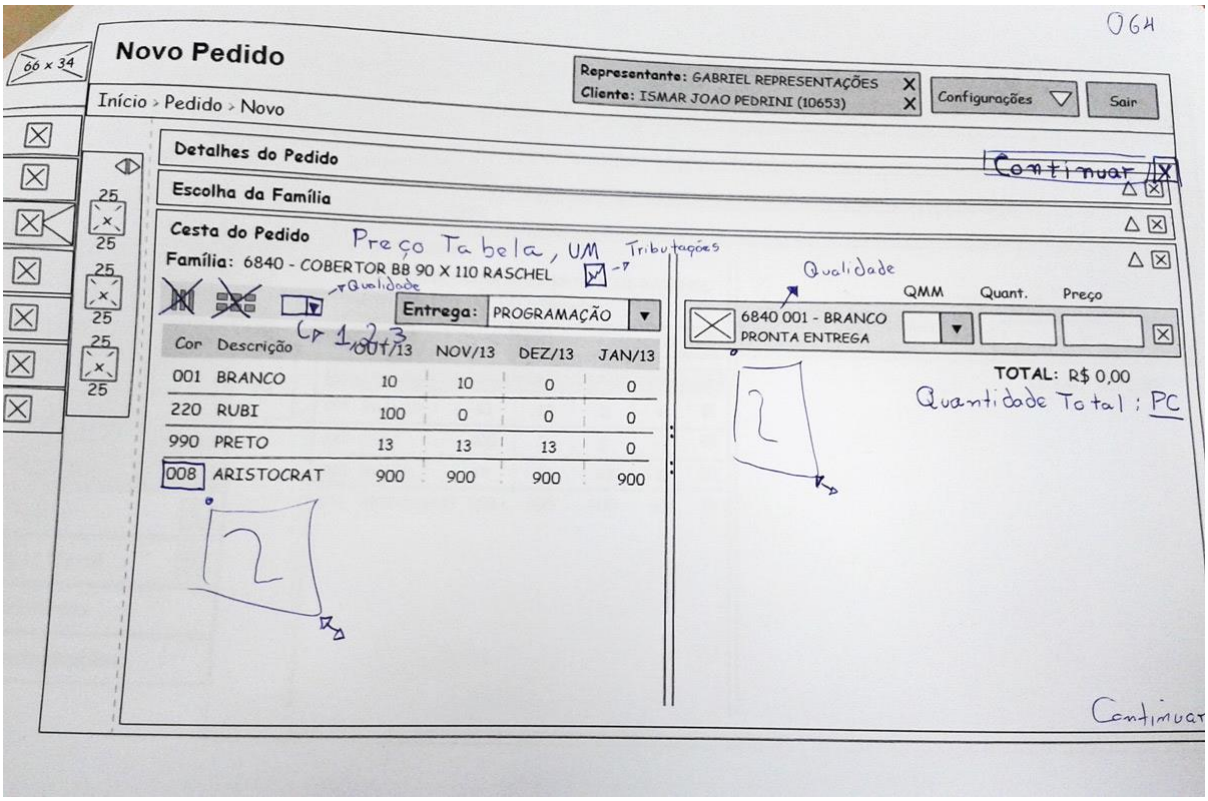


Figura 16 - Protótipo P2 com observações dos participantes

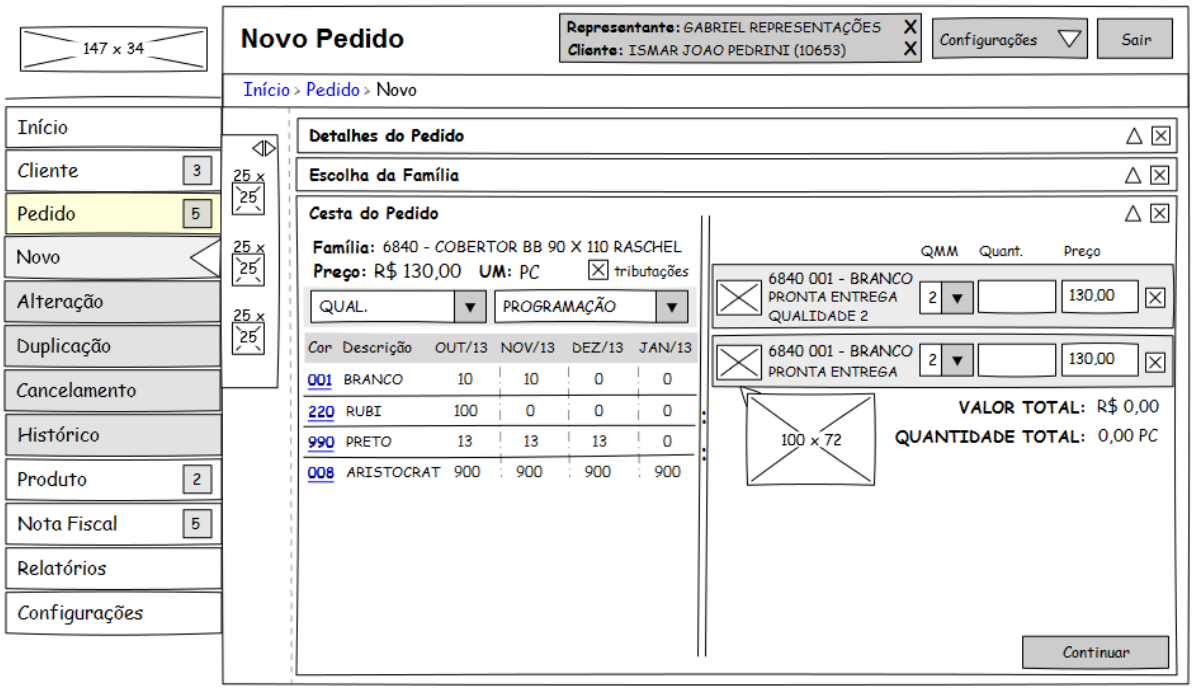


Figura 17 - Protótipo P2 ajustado

Considerações

Durante a utilização dos protótipos em reuniões, percebeu-se a importância de ter pelo menos um designer de interface e um programador presentes. Durante estas reuniões houve detalhamentos dos requisitos, surgindo novas necessidades e os demais participantes seriam necessários para entender e discutir a viabilidade dos novos recursos propostos pelos usuários.

Mesmo com a existência de ajustes pertinentes ao negócio e às solicitações de novas funcionalidades, a participação do usuário neste momento do projeto teve mais ênfase nos detalhes de interface do que em detalhes de negócio. Não se sabe ao certo se foi pela orientação que demos durante a reunião, ou se muitos detalhes de negócio já haviam sido pré-definidos corretamente, não havendo necessidade de refinamentos.

Notavelmente a aplicação desta prática de UCD ao projeto, especialmente nos protótipos resultou na melhoria da qualidade do sistema, diminuindo o retrabalho de implementação e redesign da interface, e aumentando a satisfação do usuário e *stakeholders*.

6.3.8. 8ª Etapa – Implementação da Arquitetura e UI

Durante 2 semanas, 2 analistas Web dedicaram 50% do seus períodos para a implementação da arquitetura previamente estabelecida. Após a implementação da arquitetura e consideração das alterações sugeridas durante o Design Participativo, foi construído um protótipo de alta fidelidade com características de navegação, paineis, menus, botões e etc (figura 17).

Para alcançar um leiaute adaptado à diversos tipos de dispositivos, o qual é uma necessidade emergente quando trata-se de aplicações em plataforma web, foi utilizado o framework Bootstrap. O Bootstrap é uma coleção de convenções de CSS e HTML construído em meados de 2011 por desenvolvedores do Twitter como resposta para manter e organizar requisitos de *front-end*. Possibilita o desenvolvimento de aplicativos responsivos de forma rápida, ou seja, que se adaptam a quaisquer tipos de dispositivos, como *smartphones*, *tablets* e *laptops* e PCs. Com o uso deste, o usuário navega em um website que possui tipologia, botões, formulários e tabelas elegantes e amigáveis [BOOTSTRAP].

Alguns usuários chave, que podiam ser identificados com as personas, foram convocados para conhecer e navegar no protótipo construído. O *feedback* dos usuários foi capturado neste momento e algumas dúvidas, sugestões e erros foram identificados e anotados.

A figura 17 ilustra um protótipo funcional com uma breve apresentação da tela *home* (*dashboard*), a navegação dos *menus* foi definida a esquerda. Os analistas também se preocuparam em desenhar botões e campos grandes com boa visibilidade.

Na figura 17 pode ser compreendido o funcionamento do leiaute responsivo adaptado à tela e resolução de um *smartphone*. Neste novo leiaute, nota-se que o *menu* foi redirecionado para o topo da página e agora abrirá verticalmente. As caixas de recursos como Caixa de Mensagens, Acesso Rápido, Metas e Preços também são redistribuídas verticalmente. Os botões de acesso rápido são redimensionados e ficam dispostos na primeira seção, ou seja, quando abrir a página web em um celular *smartphone* está será a primeira seção visível do website ao usuário, com botões em destaque para fácil *click*.

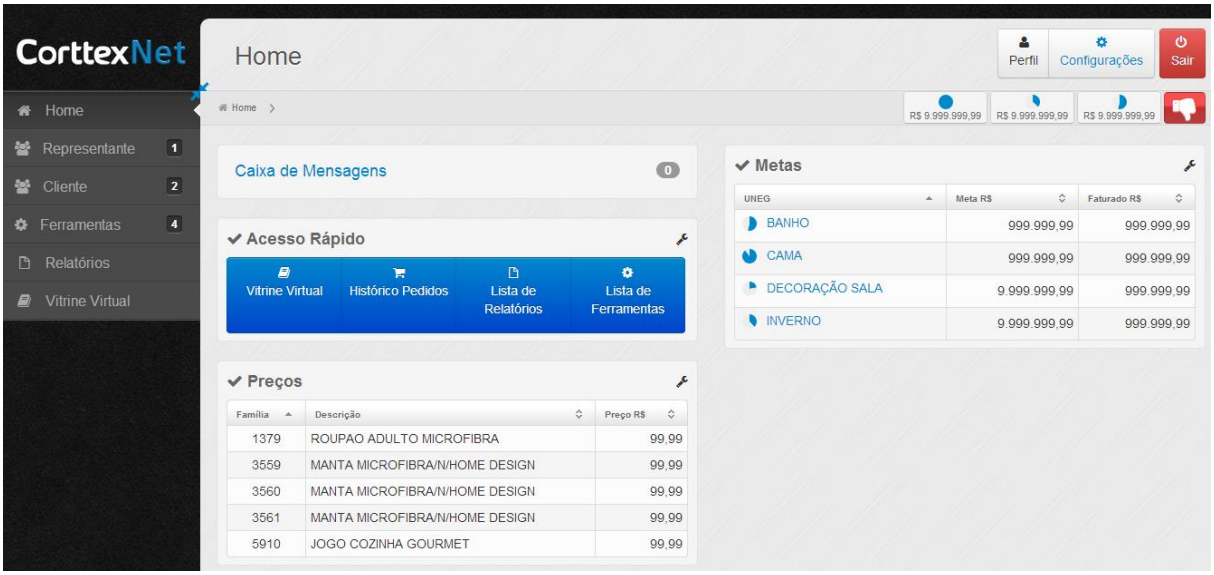


Figura 18 - Leiaute da tela *Home*

Figura 19 - Leiaute responsivo para visualização em *Smartphones*

Conforme prática do processo de desenvolvimento ágil XP e com a construção de alguns módulos para aceitação da arquitetura, diversos testes de unidade foram escritos pelos desenvolvedores. Durante a construção dos *WebServices*, construção de recursos para autenticação e autorização de usuários e contrução de módulos javascript utilizando jQuery, os testes de unidade foram validados.

Artefatos Gerados

A equipe concluiu neste período o protótipo funcional para uma visão geral do sistema. Para testar o funcionamento da comunicação com o ERP e consumo de *Webservices*, foi criado um protótipo uma tela de Vitrine Virtual (figura 19) como meio de disponibilizar a visualização de imagens e ficha técnica dos produtos da empresa, e também a tela de inclusão de clientes referenciados como "*prospect*" ou clientes em potencial (figura 20). A tela de cadastro do cliente "*prospect*" é feita no momento que o mesmo realiza um pedido de venda e há necessidade de análise comercial e de crédito. Neste último protótipo foi possível testar a leitura e gravação de dados no ERP através dos *WebServices*.

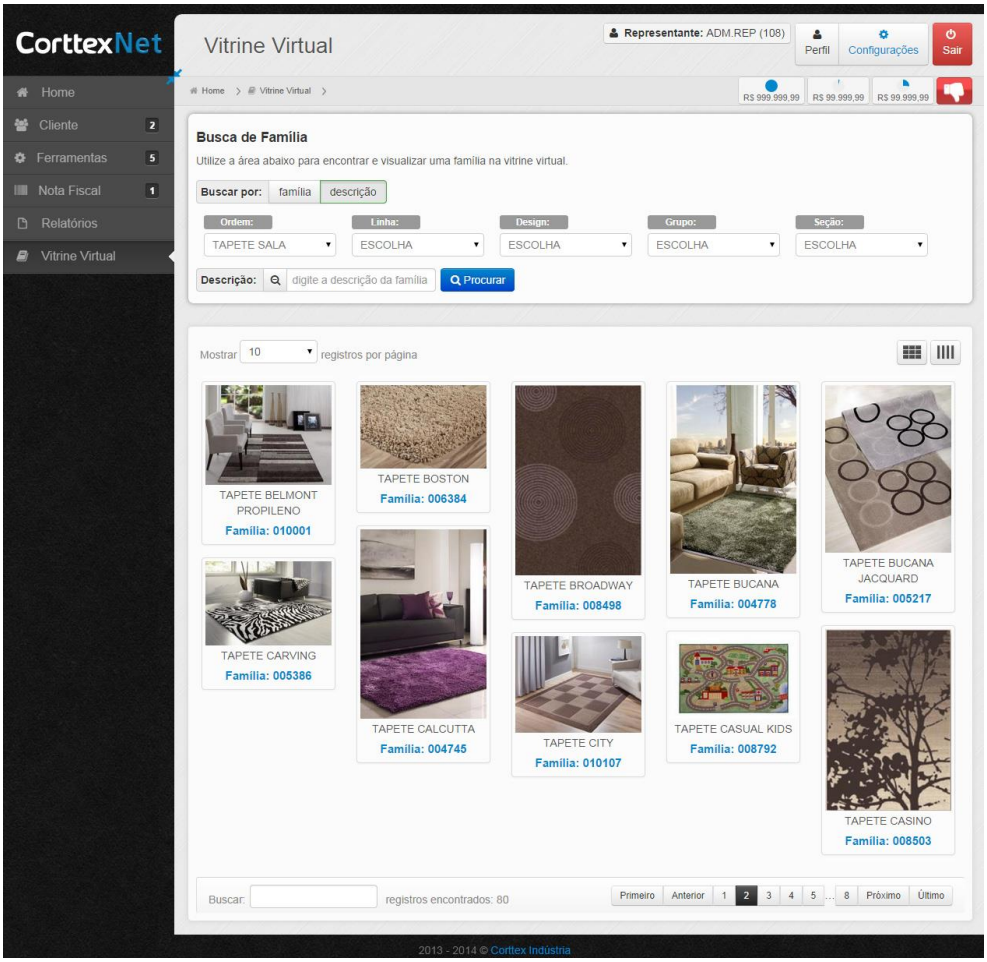


Figura 20 - Protótipo funcional da Vitrine Virtual

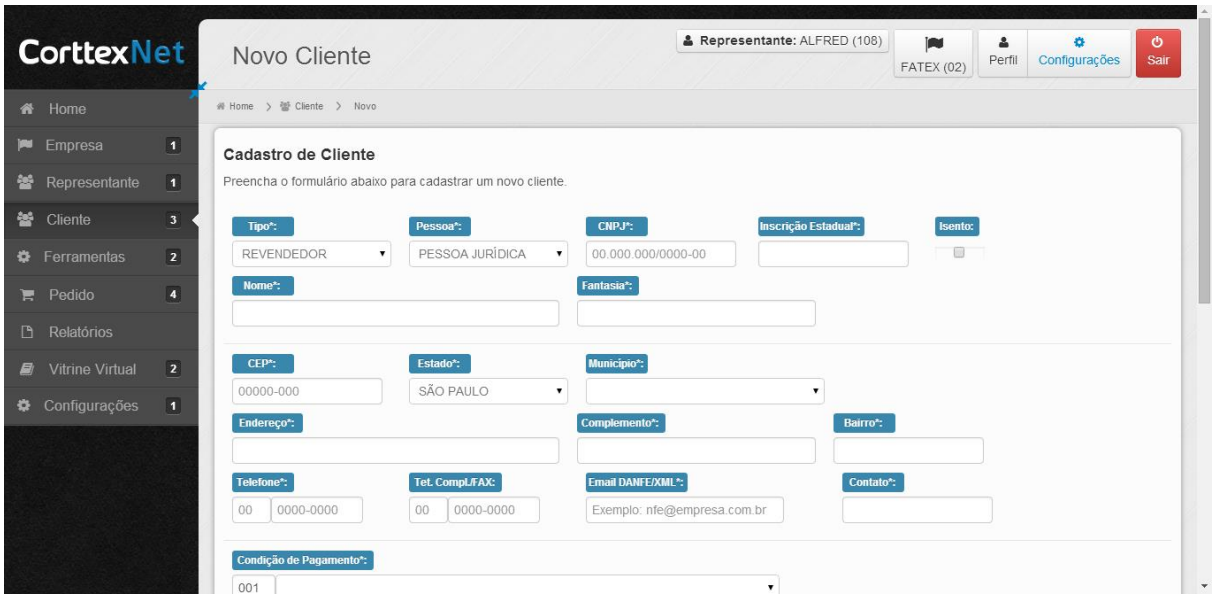


Figura 21 - Protótipo funcional da Tela de "*prospect*"

Considerações

Muitos são os comentários de diversos autores sobre aspectos teóricos de usabilidade e experiência do usuário, no entanto, pouco sabemos sobre estas abordagens na prática. De acordo com o resultado obtido no protótipo funcional e no *feedback* recebido pelos usuários, uma nova tecnologia como o Bootstrap pode agregar valor sobre a usabilidade do sistema Web. Aqueles usuários que já estavam habituados a navegar em sistemas de plataforma Web não tiveram quaisquer dificuldades em navegar nos protótipos construídos, inclusive daqueles que testaram a ferramenta em *tablets* e *smartphones*.

Como forma de capturar o *feedback*, antes que o participante iniciasse a navegação, o membro responsável da equipe disse a este: “encontre a Vitrine Virtual, encontre um artigo de interesse dentre uma lista pré-estabelecida e veja fotos dos produtos disponíveis”. Surgiram alguns questionamentos sobre a localização da ficha técnica do produto e relatórios que não estavam no escopo do protótipo, e assim, o gerente de projetos tratou de explicar qual a finalidade daquela observação.

Esta etapa, por envolver a construção dos protótipos funcionais, a construção e testes da arquitetura, o desenho de HTML e CSS das páginas e documentação, custou da equipe pouco mais de 4 semanas de trabalho.

6.3.9. 9ª Etapa – Priorização do *Backlog* e Definição de Futuras Iterações

Por fim, a última etapa prevista para o Ciclo Zero foi a priorização do *product backlog* e a definição das futuras iterações. Nesta fase, a equipe agendou algumas reuniões para discutir sobre os protótipos não funcionais chaves a serem completamente implementados e entregues no final de cada iteração. Lembrando que a escolha dos protótipos deveria ocorrer com alguma orquestração para o desenho de um fluxo contínuo, com via de se tornar um produto utilizável ao final de cada iteração do processo ágil.

Juntamente com os protótipos não funcionais, também foram utilizados os cenários de uso desenhados e as histórias, a fim de auxiliar no entendimento do fluxo de trabalho dos usuários.

Foi acordado entre a equipe que cada iteração duraria 15 dias, incluindo uma reunião para abertura da iteração e uma reunião para conclusão e análise dos resultados, sempre havendo a entrega de algo funcional ao termino de cada uma.

Artefatos Gerados

Para esta fase foram entregues cronogramas (inclusive para a alta diretoria), também foram entregues documentações com a definição de requisitos e características das próximas iterações.

A documentação da priorização incluiu o nome e descrição do requisito com base nas histórias, os critérios de aceitação, o grau de prioridade e a iteração onde este requisito seria implementado.

Considerações

Esta fase custou da equipe cerca de 2 dias, um para a reunião de priorização e definição, e outro para análise dos artefatos gerados pelos analistas.

Na priorização do *product backlog*, os requisitos e histórias mandatórias foram colocadas em destaque, como por exemplo, os requisitos não funcionais de segurança, performance e usabilidade. No entanto, a equipe deixou um espaço para a participação e priorização do usuário.

Ao final da definição das prioridades, alguns usuários chave, conforme identificados pelas personas, foram convidados para analisar os requisitos a serem implementados em cada iteração. Houve a participação ativa de todos eles e alguns requisitos foram reposicionados para garantir a entrega de primeiras versões funcionais do sistema com maior riqueza em recursos e interface.

7. Análise dos Resultados

Neste capítulo é descrita a análise crítica do capítulo anterior acerca dos resultados obtidos com a aplicação das técnicas de UCD e XP no Ciclo Zero.

Para começar, diferente da opinião de alguns autores sobre haver entregas de software funcional ao fim de cada iteração do XP (ou contra a própria proposta de pequenos *releases* do XP), houve entrega de ferramentas funcionais e um pacote completo de artefatos, incluindo documentações do sistema e da arquitetura, um *product backlog* contendo um pacote de requisitos chave definidos como escopo inicial, documentações de análise de negócio e dos usuários, protótipos de baixa fidelidade e protótipos funcionais, além de uma ferramenta navegável, conceituando desde cedo a usabilidade, a interatividade e os recursos chave priorizados, e que posteriormente serão implementados integralmente na ferramenta Web.

A alta diretoria, após o encerramento da fase de iniciação, pôde acessar a ferramenta e analisar o trabalho empenhado pela equipe de analistas/desenvolvedores alocada para este projeto.

Antes de compor considerações sobre entregas de software funcional em cada *release* das iterações do processo ágil, algumas perguntas precisam ser respondidas, como: Quem é seu cliente? O que ele considera como software funcional? O que ele considera como uma entrega válida numa fase de iniciação de projeto? Para nosso projeto, obtivemos sucesso neste aspecto pois o cliente ficou satisfeito com as entregas propostas e executadas. Ao final do Ciclo Zero, o cliente recebeu uma demonstração do sistema através de um protótipo funcional, agregando aspectos de organização da informação, interação e design da interface.

Fundamentando a análise de resultados, a equipe realizou entrevistas e apresentações do sistema Web para os diretores, gerentes e representantes de vendas, avaliando e sumarizando suas opiniões construtivas sobre a ferramenta, em aspectos de usabilidade, design interativo, performance e organização de informações, mesmo que ainda informações não reais.

Suas opiniões e sugestões foram suficientes para considerar o projeto como um sucesso. E tudo isso alcançado com uma curta pesquisa e sem especialistas de domínio, como designers, especialistas em usabilidade e/ou especialistas em experiência do usuário.

Todas as opiniões e sugestões obtidas durante as reuniões foram documentadas e repassadas à equipe, a fim de auxiliar na produção dos próximos ciclos de desenvolvimento.

Um importante fator para a satisfação e confiabilidade da ferramenta foi o desenvolvimento orientados a testes, onde diversos módulos de testes de unidade foram criados e também os testes de aceitação conforme sugeridos nos cartões de histórias dos usuários através dos critérios de aceitação.

Durante algumas reuniões onde o usuário navegou na ferramenta Web, percebeu-se que mais um aspecto foi atendido, a facilidade de aprendizado, minimizando assim a necessidade de treinamento. A ferramenta ficou intuitiva e facilmente navegável conforme opinião dos próprios usuários finais.

Conforme o cenário atual dos representantes de vendas, no que diz respeito ao uso de *notebooks* e *PCs*, já que muitas vezes há dificuldade em levar este tipo de equipamento até o cliente ou ao ponto de venda, o leiaute responsivo ganhou um grande valor. A possibilidade de utilizar a mesma ferramenta Web no *notebook*, *tablet* e *smartphone* impressionou os usuários.

Contudo foram identificadas algumas dificuldades, são elas:

- Com a consideração de colaboradores internos como usuários finais, ou representantes dos usuários finais, a equipe esbarrou naqueles que não deixavam seus trabalhos rotineiros para se dedicar às tarefas solicitadas pela equipe de desenvolvedores;
- Quando era necessário que o usuário realizasse algum tipo de teste, um membro da equipe de TI tinha que ficar ao seu lado acompanhando, senão eles não eram finalizados em tempo hábil;
- Algumas tarefas precisaram de diversos usuários participantes, como no levantamento das histórias. Não houve o comprometimento de muito deles o que dificultou o trabalho de levantamento de requisitos;
- Alguns usuários chave para o projeto estavam fora da empresa e a equipe não pôde se locomover até seus ambientes. Mesmo com os representantes dos usuários, o projeto ficou com *gaps* pertinentes a segmentos do sistema relacionados a interação do cliente e do representante de vendas;

- Em contradição à proposta de alguns autores sobre a duração para as atividades do Ciclo Zero, o projeto estourou as estimativas com gasto superior a 14 semanas. Isto se deve principalmente à estimativa deficiente de participantes para as atividades de investigação e construção dos artefatos. Foi alocada uma equipe de sete membros para o trabalho, no entanto, menos da metade deles puderam se empenhar nas atividades.

Os parágrafos a seguir apresentam detalhes sobre a correlação entre os a práticas aplicadas e os resultados obtidos neste trabalho contra os pontos de observação dos autores de referência.

Conforme apresentado em [SY], num projeto completamente novo, como foi nosso caso, devem ser realizadas entrevistas e investigações contextuais durante as visitas aos clientes e derivar designs de alto nível para guiar as decisões de características do produto, e quando é uma capacidade ou mercado novo, devem ser feitas breves descrições dos usuários alvo e seus fluxos de trabalho, através de personas e cenários. [NAJAFI e TOYOSHIBA, 2008] [KOLLMANN e SHARP, 2009] e [UNGAR e WHITE, 2008] também consideram no Ciclo Zero as práticas de levantamento de cenários, construção de personas, construção de protótipos e investigação contextual.

A facilidade de contato com o cliente e o usuário final, que estavam muitas vezes alocados ao lado do ambiente da equipe de TI, auxiliou na execução das práticas acima citadas, conforme apresentado sucintamente no capitulo 6.3. Atividades Propostas - Ciclo Zero.

Ainda conforme a mesma autora, a equipe de UX, que no nosso projeto foi disciplinada pela própria equipe de desenvolvedores, poderia trabalhar em um conjunto restrito de requisitos de usabilidade já que o processo ágil prevê a priorização do *product backlog* para cada ciclo de desenvolvimento. Para nós, foi realizada a construção de protótipos de baixa e alta fidelidade apenas para os requisitos priorizados.

De acordo com [SANTOS, 2012], há necessidade da existência da "Grande Figura", para evitar trabalho desnecessário, gerenciar a demanda e manter em mente o foco do projeto. Por mais que previsto um gerente de projetos com o perfil de a "Grande Figura", cada colaborador da equipe de TI teve que ser alto

gerenciável e multidisciplinar. Quanto ao trabalho desnecessário, gerenciamento de demanda e foco do projeto, foram aspectos discutidos frequentemente nas reuniões diárias.

Em [SOHAIB e KHAN] o autor defende que o usuário pode participar ativamente da atividade de levantamento de requisitos juntamente com a equipe de desenvolvimento e testar o produto de software. No projeto da ferramenta Web, o usuário contribuiu ativamente com o levantamento de requisitos através do uso da ferramenta EasyBacklog, onde estes mesmos preencheram as fichas de histórias e critérios de aceitação. Houve trabalho dos desenvolvedores também para a definição de algumas histórias e a transformação destas características do software em requisitos refinados.

Os autores de [SILVA, MARTIN, MAURER e SILVEIRA, 2011] consideram atividade de criação de protótipos de baixa fidelidade e histórias de usuários com critérios de aceitação para o Ciclo Zero, além das também citadas por [SY], como as entrevistas e investigação contextual.

A construção de protótipos de baixa fidelidade foi um importante fator para a participação do usuário final. Neste momento foram feitos refinamentos na interface incluindo considerações de negócio. Os critérios de aceitação apoiaram os testes de aceitação para garantir que os objetivos dos requisitos chave fossem atendidos.

8. Considerações Finais

Como conclusão deste trabalho, abaixo são apresentadas observações sobre as considerações finais e lições aprendidas conforme estudo realizado numa indústria Têxtil que necessitava de uma série de ferramentas para automatização de seu processo comercial, incluindo em especial a ferramenta Web, alvo deste estudo.

Conforme estudos realizados para a elaboração deste trabalho, percebeu-se que não existe uma definição bem estabelecida sobre quais atividades devem ser contempladas no Ciclo Zero. Não foi encontrado um guia sobre como fazer, mas os autores observam algumas atividades e as citam com caráter informal para esta fase do projeto.

Um fator de sucesso no estudo de caso se deve ao cliente estar inserido dentro do mesmo ambiente que a equipe de TI, portanto, o grau de contato e intimidade profissional foi amplo, facilitando assim o processo de investigação dos usuários, alcançando maiores testes do software funcional e aumentando o *feedback* sobre aspectos de usabilidade, interface e regras de negócio.

Em modos gerais, o objetivo para este trabalho foi alcançado. Foi entregue uma proposta de atividades para o Ciclo Zero, demonstrando detalhadamente o processo de desenvolvimento em um estudo de caso de uma indústria do ramo têxtil. A equipe também conseguiu entregar documentações, protótipos de baixa e alta fidelidade, e uma ferramenta funcional destacando desde cedo aspectos de usabilidade, performance e segurança. O protótipo funcional no Ciclo Zero nada mais foi do que um esqueleto da arquitetura e os principais requisitos chave previamente implementados no formato de protótipos funcionais.

Diferente da obrigatoriedade de entregas em todas as iterações dos processos ágeis contra considerações de alguns autores sobre o problema da não entrega de produto funcional no Ciclo Zero, neste projeto base do estudo de caso, houve sim entrega de software funcional ao final do Ciclo Zero em forma de protótipos já mencionados. Ao navegar nestes protótipos, o usuário final já poderia dar *feedbacks* sobre sua satisfação, facilidade de uso, organização das informações, funcionalidade dos recursos e estética da interface.

Como proposta para este trabalho, na inclusão de atividades de UCD dentro do processo de desenvolvimento ágil, houve completa participação do usuário final

em todas atividades da fase de iniciação como é possível ser visto no capítulo 6.3. Atividades Propostas - Ciclo Zero.

Implicitamente outro objetivo foi alcançado, conforme já mencionado anteriormente, a equipe de TI por ser constituída apenas por prestadores de serviços, procurava reconhecimento profissional e visibilidade sobre toda organização. Conforme *feedbacks* obtidos em reuniões realizadas após a conclusão da fase de iniciação, representantes de vendas, gerentes e diretores elogiaram os resultados conquistados pela equipe de analistas.

Com o prazo apertado, a equipe pequena e dificuldade de contato com o cliente final ou consumidor final da organização, não foi possível incluí-los no centro do desenvolvimento, no entanto, a equipe pretende considerá-los em futuras iterações. Vale destacar que o consumidor final faz parte de uma parcela considerável de possíveis usuários da ferramenta Web, de diferentes etnias, faixas etárias e grau de conhecimento tecnológico, o que demonstra a importância deste no ciclo de desenvolvimento, e por sua vez, no UCD.

Mesmo com todo trabalho de pesquisa realizada na fase de iniciação, alguns artefatos do desenvolvimento foram descartados por conta de mudanças nos requisitos, prioridades ou conhecimento de detalhes do negócio anteriormente desconhecidos pela equipe.

No Ciclo Zero foram entregues diversos artefatos, como: Documento de Visão, Documento de Descobertas, Cenários de Uso, Personas, *Product Backlog* contendo as principais características esperadas para o sistema, Protótipos de Baixa Fidelidade, Protótipos de Alta Fidelidade e o software utilizável conceituando desde cedo os aspectos de usabilidade, performance e segurança. Vale ressaltar que todos estes artefatos foram demonstrados em reunião para o cliente, que por sua vez, os considerou como entregas válidas para a fase de iniciação do projeto.

A confiabilidade no sistema foi possível ser compreendida a partir da análise de LOG, rastreio de acessos e até monitoração de cliques em áreas da tela, com isto, também foi possível compreender a facilidade de uso. A equipe se preocupou em entregar uma ferramenta confiável e de fácil uso, e um dos fatores para isto foi apoiado pela migração drástica de interface, lembrando que o usuário deixaria de utilizar um sistema ultrapassado de "tela azul", onde nem ao menos era possível utilizar um mouse.

Outra característica de usabilidade conquistada no projeto foi o leiaute responsivo, compatível com diversos tipos de dispositivos, como: *Smartphones*, *Tables* e *PCs*. Esta característica surpreendeu diversos usuários, em destaque os gerentes que atualmente vão para a rua realizar o acompanhamento comercial munidos de *Tablets*.

Por fim, a organização onde o estudo foi aplicado conquistou maior valor em seus sistemas computacionais, já que os usuários passaram a emitir 100% dos pedidos de venda no ambiente web, além de realizar todo o acompanhamento comercial, de onde estiver, sem a necessidade de entrar em contato com os assistentes comerciais e de SAC, quando eles não conseguem sozinhos atingir este objetivo. O sistema que garante a melhor experiência do usuário diminuiu também o trabalho da equipe de suporte técnico e ofereceu à organização maior ganho competitivo no mercado, por disponibilizar uma ferramenta ágil e positivamente diferente da que está no concorrente.

Ao termino deste documento, 14 meses trabaho em nosso projeto se passaram e muito foi aprendido com as experiências, resultando no aperfeiçoamento das atividades e motivando a equipe a enfrentar os desafios. Os desafios agora parecem um pouco mais simples, já que temos o usuário ao nosso lado.

8.1. Trabalhos Futuros

Primeiramente, futuros pesquisadores e profissionais poderiam dar continuidade a este trabalho explorando os resultados a longo prazo obtidos com a aplicação de técnicas de UCD e processos ágeis na fase de concepção do projeto, avaliando assim a qualidade do produto final obtido.

Poderiam ser utilizados também métodos quantitativos para análise de usabilidade em ferramentas desenvolvidas com estas práticas, analisando por sua vez métricas de eficiência, eficácia e satisfação.

Seria interessante também explorar as diferenças da participação do usuário final em diferentes perfis de projetos, como o desenvolvimento de sistemas *e-commerce* e *e-learning* que possuem alto grau de interatividade e diversidade de usuários.

REFERÊNCIAS

[AMBLER, S.; LINES, M.] **Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise**. 1ª Ed., IBM Press, 2012.

[BAKER L.; BROSCHINSKY, D.] **Using Persona with XP at LANDesk Software, an Avocent Company**. IEEE *Computer Society*, 2008.

[BAXTER, K.; KELKAR, K.; MILLER L.; PATTON J.; ROSENSTEIN A.; VILLAMOR C.; FEDEROFF M.] **Extreme Usability: Adapting Research Approaches for Agile Development**. ACM *Digital Library*, 2008.

[BECK K.] **Extreme Programming Explained**. 1ª Ed., Addison-Wesley, 1999.

[BOOTSTRAP] **Framework Bootstrap Twitter**. Disponível em: <https://dev.twitter.com/blog/bootstrap-twitter>. Acesso em: Março de 2014.

[EASYBACKLOG] **Software EasyBacklog**. Disponível em: <http://easybacklog.com>. Acesso em: Junho de 2013.

[CARNEIRO, C.; BARAZI R.] **Beginning Rails 3**. Disponível em: <http://beginningrails.com>. Acesso em: Agosto de 2013.

[CONSTANTINE L.; LOCKWOOD L.] **Usage-Centered Engineering for Web Applications**. ACM *Digital Library,* 2001.

[GARRET J.] **The Elements Of User Experience - User-Centered Design for the Web and Beyond. Second Edition**. Berkeley, CA. 2011, p. 44-51.

[GEORGE C.] **User-Centered Library Websites: Usability evaluation methods**. ACM *Digital Library*, 2008.

[HODA R.; KRUCHTEN P.; NOBLE S.; MARSHALL S.] **Agility in Context**. ACM *Digital Library,* 2010.

[HOM J.] **The Usability Methods Toolbox Handbook**. Disponível em: <http://www.idemployee.id.tue.nl>. Acesso em: Setembro de 2013.

[ISO] **ISO 25000 Standards – ISO 25010.** Disponível em: <http://iso25000.com>. Acesso em: Abril de 2014.

[JACOBSEN C.; SUTHERLAND J.] **Mature Scrum at Systematic**. Disponível em: <http://systematic.com>. Acesso em: Julho de 2013.

[KEINONEN T.] **User-Centered Design and Fundamental Need**. ACM *Digital Library*, 2008.

[KOLLMANN J.; SHARP H.] **The importance of Identity and Vision to user experience designers on agile Projects**. IEEE *Computer Society*, 2009.

[KUNIAVSK M.] **Observing the User Experience: A Practitioner's Guide to User Research**. ACM *Digital Library*, 2003. p. 43-44.

[LEAVITT M.; SHNEIDERMAN B.] **Research-Based Web Design & Usability Guidelines**. Disponível em: <http://www.usability.gov>. Acesso em: Julho 2013.

[PC MAGAZINE] **Definition of WebServices**. Disponível em: <http://www.pcmag.com/encyclopedia/term/54345/web-services>. Acesso em: Junho de 2014.

[MANIFESTO] **Manifesto for Agile Software Development**. Disponível em: <http://agilemanifesto.org/>. Acesso em: Agosto de 2013.

[MACHADO M.; MEDINA S.] **SCRUM – Método Ágil: uma mudança cultural na gestão de projetos de desenvolvimento de software**. Disponível em:

<http://www.faculdadedoguaruja.edu.br/revista/downloads/edicao12009/Artigo_5_Prof_Marcos.pdf>. Acesso em: Agosto de 2013.

[MATTILA K.; ROTO V.; HASSENZAHL M.] **Towards Practical User Experience Evaluation Methods**. Disponível em: <http://uxpa.org>. Acesso em: Julho de 2013.

[MATERA M.; RIZZO F. TOFFETTI G.] **Web Usability: Principles and Evaluation Methods**. Polytechnic Institute of Milano Piazza Leonardo da Vinci, 2006.

[MAO J.; VREDENBURG K.; SMITH P.; CAREY T.] **The state of user-centered design practice**. Magazine Communications of the ACM, 2005.

[NIELSEN] **Nielsen Norman Group**. Disponível em: <http://www.nngroup.com>. Acesso em: Agosto de 2014.

[NORMAN D.] **The design of everyday things**. Internet Archive, 2002.

[NAJAFI M.; TOYOSHIBA L.] **Two Case Studies of User Experience Design and Agile Development**. IEEE *Computer Society*, 2008.

[PATTON, J.] **Hitting the Target: Adding Interaction Design to Agile Software Development**. ACM *Digital Library*, 2002.

[PAMPLONA V. KUHN G.] **Apresentanto XP: Encante seus clientes com Extreme Programming**. Disponível em: <http://www.dignow.org>. Acesso em: Setembro de 2013.

[POLLICE G.] **Using the IBM Rational Unified Process for Small Projects**. IBM 2003. p. 12.

[SANTOS A.] **Aplicação de práticas de usabilidade ágil em software livre**. Biblioteca Digital USP, 2012.

[SVANAES D.; GULLIKSEN J.] **Understanding the Context of Design – Towards Tactical User Centered Design**. ACM *Digital Library*, 2008.

[SILVA T.; MARTIN A. MAURER F.; SILVEIRA M.] **User-Centered Design and Agile Methods: A Systematic Review**. IEEE *Computer Society*, 2011.

[SOHAIB O.; KHAN K.] **Integrating Usability Engineering and Agile Software Development**. IEEE *Computer Society*, 2010.

[SOHAIB O.; KHAN K.] **Incorporating Discount Usability in Extreme Programming**. Disponível em: <http://www.sersc.org>. Acesso em: Julho de 2013.

[SUYDAM D.] **Architect Solutions, Sprint Zero - Where Your Software Project Begins, White Paper**. Disponível em: <http://blog.architech.ca>. Acesso em: Julho de 2013.

[SHORE J.; WARDEN S.] **The art of agile development**. O'Reilly Media, 2008.

[SY D.] **Adapting Usability Investigations for Agile User-centered Design**. Disponível em: <http://www.usabilityprofessionals.org>. Autodesk Inc., 2009. Acesso em: Julho de 2013.

[UNGAR J.; WHITE J.] **The Design Studio: Interface Design for Agile Teams**. IEEE *Computer Society*, 2008.

[USABILITY] **Usability.gov – Improving the User Experience**. Disponível em: <http://usability.gov>. Acesso em: Junho de 2013.

[XIONG Y.; WANG A.] **A New Combined Method for UCD and Software Development and Case Study**. IEEE *Computer Society,* 2010.

[XP] **Extreme Programming: A Gentle Introduction**. Disponível em: <http://www.extremeprogramming.org>. Acesso em: Setembro de 2013.

[WILLIAMS H.; FERGUSON A.] **The UCD Perspective: Before and After Agile**. IEEE *Computer Society*, 2007.

[WODTKE C.; GOVELLA A.] **Information Architecture: Blueprints for the Web**. Disponível em: <http://books.google.com>, 2009. Acesso em: Novembro de 2013. p. 130-140.

GLOSSÁRIO

AJAX – Escrito em Javascript, permite que páginas web sejam mais interativas e se comportem como aplicações locais. Permite à aplicação web recuperar pequenas porções de dados do servidor sem a necessidade de recarregar toda a página [PC MAGAZINE].

Burndown Chart – “é um gráfico que mostra a quantidade de trabalho cumulativo restante de um Sprint, dia por dia. Neste gráfico, a altura indica a quantidade de tarefas do Sprint Backlog não completadas, e o comprimento são os dias. Este gráfico é um dos principais recursos de medição do processo de desenvolvimento e um diferencial para a metodologia SCRUM. Quanto mais horizontal, melhor.” [MACHADO e MEDINA].

Cobol – COmmon Business Oriented Language. Linguagem de programação de alto nível que tem sido a linguagem principal para aplicações de mainframes. É uma linguagem compilada e uma das primeiras de alto nível já desenvolvidas. Foi oficialmente adotada em 1960 [PC MAGAZINE].

CSS – Cascading Style Sheets. Folha de estilos para o formato HTML aprovado pelo World Wide Web Consortium. Contem milhares de configurações de leiaute que são aplicados sobre as páginas HTML sempre que estas são baixadas [PC MAGAZINE].

Expertise – De acordo com o dicionário, “Análise, avaliação especializada ou comprovação feita por um especialista num assunto específico (...)”

HTML – Hyper Text Markup Language. Formato de documento padrão para Web pages definido por Internet Engineering Task Force (IETF). Cada página contem um conjunto de *tags* HTML incorporadas no texto que definem o leiaute, fontes e links da página [PC MAGAZINE].

HTML 5 – é a linguagem para estruturação e apresentação de conteúdo interativo que está em sua última versão 5, trazendo novos recursos como, por exemplo, a disponibilidade de controle de multimídias (Áudio e Vídeo), e é um candidato em potencial para aplicações móveis;

JQuery – Biblioteca popular de rotinas escritas em Javascript para automatização de Web pages. Inclui inúmeras funções para manipular HTML e CSS bem como AJAX [PC MAGAZINE].

JSON – é um modelo “leve” de estrutura de dados para comunicação entre sistemas. É uma alternativa para o tradicional formato XML;

OAUTH 2.0 – permite ao desenvolvedor controlar o acesso do cliente a requisições HTTP seguras. O cliente da requisição solicita ao proprietário do recurso a autorização para acesso sem expor seus dados de autenticação, como usuário e senha.

SCRUM – não é uma sigla e não existe uma tradução, o termo é utilizado pelo esporte Rugby, onde a equipe prepara um formação, com a participação de todos integrantes da equipe num mesmo objetivo e se algum deles falha, todos falham [MACHADO e MEDINA].

XP – define-se por um jeito leve, eficiente, de baixo risco, flexível, previsível, ciêntífico e engraçado de desenvolver software. XP é uma disciplina de desenvolvimento de software, pois há certas coisas que devem ser feitas para “ser XP”. Foi desenhado para trabalhar com projetos construídos por 2 a 10 programadores e é baseado em 4 valores: Comunicação, Simplicidade, *Feedback* e Coragem [BECK, 1999].

Web Services – em geral, é qualquer serviço online entregue a partir de um web site. Quando é mencionado no plural “*Web Services*”, o termo frequentemente se refere a uma interface para uma arquitetura orientada a serviço (SOA), ao qual aplicações web interagem dinamicamente com outras aplicações web através de um padrão aberto, incluindo XML sobre HTTP, UDDI e SOAP [PC MAGAZINE].