

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
ESPECIALIZAÇÃO EM ENGENHARIA DE SOFTWARE

LEANDRO QUINGERSKI

**ESTUDO DE COMO O DESENVOLVIMENTO ÁGIL COM DESIGN
CENTRADO NO USUÁRIO PODE PROMOVER MELHORIA NA
QUALIDADE DE ENTREGÁVEIS AO CLIENTE**

MONOGRAFIA DE ESPECIALIZAÇÃO

MEDIANEIRA

2012

LEANDRO QUINGERSKI

**ESTUDO DE COMO O DESENVOLVIMENTO ÁGIL COM DESIGN
CENTRADO NO USUÁRIO PODE PROMOVER MELHORIA NA
QUALIDADE DE ENTREGÁVEIS AO CLIENTE**

Monografia apresentada como requisito parcial à obtenção do título de Especialista na Pós Graduação em Engenharia de Software, da Universidade Tecnológica Federal do Paraná – UTFPR – Campus Medianeira..

Orientador: Prof. M.Sc Alan Gavioli

MEDIANEIRA

2012



TERMO DE APROVAÇÃO

ESTUDO DE COMO O DESENVOLVIMENTO ÁGIL COM DESIGN CENTRADO NO USUÁRIO PODE PROMOVER MELHORIA NA QUALIDADE DE ENTREGÁVEIS AO CLIENTE

Por

Leandro Quingerski

Esta monografia foi apresentada às 16:30 h do dia 23 de março de 2012 como requisito parcial para a obtenção do título de Especialista no curso de Especialização em Engenharia de Software, da Universidade Tecnológica Federal do Paraná, *Campus* Medianeira. Os acadêmicos foram argüidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado com louvor e mérito.

Prof. M.Sc Alan Gavioli
UTFPR – Campus Medianeira
(orientador)

Prof M.Sc Fernando Schutz
UTFPR – Campus Medianeira

Prof **M.Sc. Beltano da Silva**
UTFPR – Campus Medianeira

Dedico este trabalho a meus pais que sempre me apoiaram com toda a estrutura para que eu pudesse estudar e evoluir como profissional e acadêmico. E também a minha namorada que me motivou para que realizasse e concluísse estes estudos.

AGRADECIMENTOS

A Deus pelo dom da vida, pela fé e perseverança para vencer os obstáculos.

Aos meus pais, pela orientação, dedicação e incentivo nessa fase do curso de pós-graduação e durante toda minha vida.

A minha namorada que sempre me incentivou e se dispôs a me dar toda ajuda que precisei para que conseguisse terminar este trabalho.

A meu orientador professor Alan Gavioli, que me orientou, pela sua disponibilidade, interesse e receptividade com que me recebeu e pela prestabilidade com que me ajudou e expôs seus conhecimentos em sala de aula.

Agradeço aos pesquisadores e professores do curso de Especialização em Engenharia de Software da UTFPR, campus de Medianeira.

Enfim, sou grato(a) a todos que contribuíram de forma direta ou indireta para realização desta monografia.

“A diferença entre falhar e ter sucesso é a diferença entre fazer algo quase certo ou
fazer algo certo”.

(Benjamin Franklin)

RESUMO

LEANDRO, Quingerski. Estudo de como o desenvolvimento ágil com design centrado no usuário pode promover melhoria na qualidade de entregáveis ao cliente. Monografia (Especialização em Engenharia de Software). Universidade Tecnológica Federal do Paraná, Medianeira, 2011.

Este trabalho tem por objetivo apresentar o funcionamento de práticas ágeis de gestão e desenvolvimento de projetos de *software*, abordando duas das principais metodologias existentes no mercado de trabalho. Evidencia também a importância do *design* e como ele pode trazer qualidade aos projetos de *software*. Por fim, uma proposta de mudanças das iterações de desenvolvimento. Essas mudanças incluem métodos de *design* e levam uma nova cultura à equipe que está atuando no projeto, trabalhando com o *design* centrado no usuário.

Como resultado do trabalho se pretende evidenciar qual o impacto de trabalhar com o *design* centrado no usuário em projetos de *software* desenvolvidos com práticas ágeis.

Palavras-chave: Práticas Ágeis, *Design* centrado no Usuário, Experiência do Usuário e Usabilidade.

ABSTRACT

LEANDRO, Quingerski. Agile practices allied to *design* methods to improve the quality of deliverables to the client. Monograph (Specialization in Software Engineering). Universidade Tecnológica Federal do Paraná, Medianeira, 2011.

This paper aims to present the operation of agile practices management and development of software projects this can address the two methodologies existing in the job market. Highlighting the importance of design and how it can bring quality to software projects presenting the design concepts. Finally a proposal for changes development iterations. These changes include design methods and lead a new culture to the team that is working on the project, working with user-centered design.

As a result of the work is intended to highlight the impact of working with user-centered design in software projects developed using agile practices.

Palavras-chave: Agile, User-centered design, User eXperience and Usability.

LISTA DE FIGURAS

Figura 1 – Estrutura Scrum.	16
Figura 2 – Gráfico Burndown.....	20
Figura 3 – Quadro Kanban.....	22
Figura 4 – Proposta de iteração.	36
Figura 5 – Localização Geográfica do Município de Foz do Iguaçu	39
Figura 6 – Mapa mental da pesquisa	41

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVO GERAL	12
1.2	OBJETIVOS ESPECÍFICOS	12
1.3	JUSTIFICATIVA	12
1.4	ESTRUTURA DO TRABALHO	13
2	PRÁTICAS ÁGEIS	14
2.1	SCRUM	14
2.1.1	CERIMÔNIAS	16
2.1.2	PAPÉIS	16
2.1.3	O TIME DE DESENVOLVIMENTO (<i>TEAM</i>)	17
2.1.4	O DONO DO PRODUTO (<i>PRODUCT OWNER</i>)	18
2.1.5	SCRUM MASTER	18
2.1.6	ARTEFATOS	19
2.1.7	PRODUCT BACKLOG	19
2.1.8	SPRINT BACKLOG	19
2.1.9	GRÁFICO BURNDOWN	19
2.2	EXTREME PROGRAMMING	20
2.3	KANBAN	21
3	DESIGN: CONCEITOS E ABORDAGEM PARA SOFTWARE	23
3.1	ERGONOMIA	24
3.2	INTERAÇÃO HUMANO-COMPUTADOR	25
3.3	USABILIDADE	26
3.4	ENGENHARIA DE USABILIDADE	28
4	PROPOSTA DE INTERAÇÕES E ENTREGÁVEIS COM FOCO NO DESIGN (AGILE USER-CENTERED DESIGN)	31
4.1	MÉTODOS DE <i>DESIGN</i>	32
4.1.1	INVESTIGAÇÃO DE CONTEXTO (<i>CONTEXTUAL INQUIRY</i>)	32
4.1.2	ENTREVISTAS (<i>INTERVIEWS</i>)	32
4.1.3	PERSONAS	33
4.1.4	TESTES DE USABILIDADE	33

4.1.5	GRUPO FOCO	33
4.1.6	PROTOTIPAGEM.....	34
4.1.7	AVALIAÇÃO HEURÍSTICA.....	34
4.2	CENÁRIO DE DESENVOLVIMENTO COM INTERAÇÕES DE DESIGN.....	35
4.3	A EXPERIÊNCIA DO USUÁRIO.....	37
5	PROCEDIMENTOS METODOLÓGICOS DA PESQUISA	39
5.1	LOCAL DO ESTUDO.....	39
5.2	TIPO DE PESQUISA OU TÉCNICAS DE PESQUISA.....	39
5.3	COLETA DOS DADOS	39
5.4	ANÁLISE DOS DADOS	40
6	RESULTADOS E DISCUSSÃO	42
7	CONSIDERAÇÕES FINAIS	44
7.1	CONCLUSÕES.....	44
7.2	TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO.....	44
	REFERÊNCIAS.....	46

1 INTRODUÇÃO

O mercado de tecnologia da informação passou por várias transformações desde sua origem e continua se renovando constantemente. Empresas têm necessidade de aumentar a produtividade e qualidade dos seus projetos.

As práticas ágeis de gerência e desenvolvimento de *software* são usadas na intenção de resolver estes pontos e são práticas novas no mercado de TI. Em comparação com metodologias mais antigas como o modelo cascata que durante muito tempo foi referência na Engenharia de *Software*. Na área de *design* existem diversas metodologias e ferramentas que auxiliam profissionais da área e isso promove maior qualidade nos projetos. Estudos sobre ergonomia e usabilidade comprovam a importância da interface para o usuário e do impacto na qualidade do produto. Neste trabalho o tema *design* é abordado com foco para projetos de *software*. (COCKBURN, 2000; ALLEMAN, 2002).

Com o tempo e experiência praticando os princípios ágeis assim como os conceitos da área de *design* em projetos de TI, enxergou-se uma lacuna entre *design* e desenvolvimento. Com esta necessidade surgiram estudos e discussões sobre como criar uma boa interação e comunicação entre projeto e cliente, tais como os trabalhos (MILLER, 2005), (SY, 2007), (GARRET, 2011).

A intenção deste estudo é expor os papéis de uma equipe de projeto e compilar as maneiras mais eficazes de alinhar as práticas ágeis com os processos e conceitos do *design*. A proposta é gerar maior produtividade aos times, considerando que as mudanças (inevitáveis) são minimizadas e a qualidade dos entregáveis têm melhorias perceptíveis pelo cliente.

1.1 OBJETIVO GERAL

Verificar a possibilidade de melhoria da qualidade dos entregáveis de *software* com base no uso de Práticas Ágeis de Desenvolvimento com *Design* Centrado no Usuário.

1.2 OBJETIVOS ESPECÍFICOS

Apresentar as práticas ágeis mais utilizadas no mercado;

Sintetizar os principais conceitos, metodologias, ferramentas e técnicas usados em projetos de desenvolvimento ágil com *design* centrado no usuário;

Verificar como aumentar a assertividade na obtenção de requisitos do usuário visando superar suas expectativas e necessidades;

Explorar uma cultura de inovação em projetos, baseada no *design*;

1.3 JUSTIFICATIVA

Geralmente programadores de *software* não possuem habilidade com criação de interfaces até mesmo por se tratarem de áreas distintas que exigem competências e perfis diferentes. Um cenário comum atualmente entre as empresas de *software* que utilizam práticas ágeis e possuem equipes multidisciplinares (*designers*, desenvolvedores e testadores) é desenvolver seus projetos com iterações que em conjuntos geram entregáveis ao cliente. (COCKBURN, 2000).

Mesmo dentro de uma pequena iteração, podem existir alterações de interface que vão agregar valor ao entregável do cliente e isso se dá pelo fato de que é o momento em que o time efetivamente está trabalhando no requisito e durante este trabalho é possível que o cliente experimente a aplicação e dê seu feedback podendo realizar alterações pontuais. (SY, 2007).

A validação dos dados de interface no começo de um ciclo de iteração pode perder validade para o cliente por algum motivo e a partir deste momento não tem porque continuar o desenvolvimento do requisito, pois gera retrabalho e perde valor ao cliente. (FERREIRA, 2007)

Em muitos casos esse tipo de situação é tratado como requisito não levantado corretamente mas o que acontece de fato é que mudanças são inevitáveis ao projeto e o que importa de fato é a satisfação do cliente. Essa satisfação se dá em prol dos resultados gerados no projeto e a qualidade do entregável está totalmente ligada à forma que o problema do cliente está sendo resolvido. (GARRET, 2011)

1.4 ESTRUTURA DO TRABALHO

Introdução;

Capítulo 2 – Práticas ágeis;

Capítulo 3 – *Design*: conceitos e abordagem para *software*;

Capítulo 4 – Desenvolvimento ágil com design centrado no usuário;

Conclusão;

2 PRÁTICAS ÁGEIS

A tecnologia de informação é uma área considerada nova quando comparada às demais áreas da indústria. Isso fez com que se herdassem práticas de engenharia não tão usuais para o desenvolvimento de projetos de *software*.

Como alternativa aos processos de cascata do RUP, as práticas ágeis emergiram com grande força trazendo uma série de novos conceitos nos quais quebraram muitos paradigmas. (COCKBURN, 2000)

Ao tratar de agilidade no desenvolvimento de *software* é necessário apresentar os princípios ágeis, o que é chamado também de manifesto ágil:

Indivíduos e interações mais que processos e ferramentas;
Software em funcionamento mais que documentação abrangente;
Colaboração com o cliente mais que negociação de contratos;
Responder a mudanças mais que seguir um plano;
(MANIFESTO ÁGIL, 2001).

Em comum as práticas ágeis têm foco no empirismo onde o time se utiliza da experiência e aprendizado adquirido nas iterações do projeto para aumentar sua produtividade. Em seguida constam as principais ferramentas e metodologias ágeis do mercado.

2.1 SCRUM

É um *framework* dentro do qual as pessoas podem resolver problemas adaptativos complexos, enquanto, produtivamente e criativamente entregam produtos com o mais alto valor possível. O Scrum é leve, de entendimento simples e extremamente difícil de ter o domínio. (SCHWABER; SUTHERLAND, 2011).

“O Scrum é um framework de processo quem tem sido usado para gerenciar o desenvolvimento de produtos complexos desde o início dos anos 90. Scrum não é um

processo ou técnica para construir produtos, é mais um framework dentro do qual se pode empregar processos e técnicas variadas. O Scrum torna clara a eficácia relativa das práticas de gerenciamento e desenvolvimento de produtos, para que você possa melhorá-las.” (SCHWABER; SUTHERLAND, 2011).

Framework é um conjunto de conceitos, valores e práticas. Dessa maneira é possível concluir que Scrum está muito mais ligado às definições de um *framework* do que de uma metodologia como é comumente referenciado no mercado de trabalho.

Metodologia não é a palavra certa para se definir Scrum simplesmente pelo fato de que o Scrum por si só não possui um passo-a-passo ou métodos que garantam sucesso do objeto em questão. É um conjunto de papéis, artefatos, valores, eventos e regras prescritos que auxiliam no processo. Ou seja, Scrum é muito mais valores do que processos ou métodos. Usar Scrum não é a garantia de sucesso do produto.

Como é possível observar, o Scrum não está relacionado à engenharia de *software*. Até porque sua proposta é ir além de projetos de *software* e atender de forma genérica o desenvolvimento de produtos no geral.

Ciclo principal no Scrum pode ser visto na Figura 1, que demonstra interações, estas duram de 2 a 3 semanas em média e são chamadas de Sprints. Elas se repetem até o objetivo final do produto ter sido concluído. Existem diversos momentos em que acontecem inspeções no trabalho, algumas se dão diariamente, outras com diferentes periodicidades ambas são realizadas através de cerimônias. (SCHWABER, 2004).

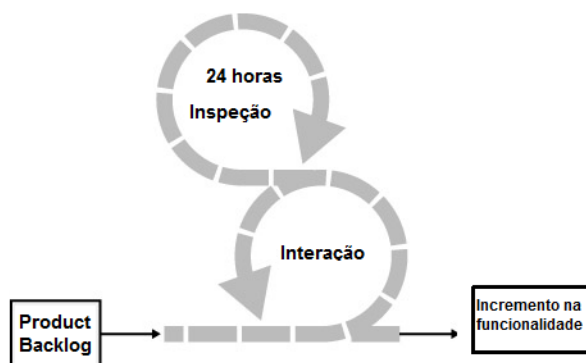


Figura 1 – Estrutura Scrum.

Fonte: Agile Project Management with Scrum

2.1.1 CERIMÔNIAS

Várias cerimônias são prescritas no Scrum, conhecidas também como eventos. Estes criam uma rotina com o intuito de diminuir a necessidade de reuniões esporádicas. A principal característica é que estes eventos são controlados por um tempo pré-determinado, conhecido como *time-box*. Isso permite que não sejam gastos tempos demais com assuntos irrelevantes e mantém o foco de todos nos resultados esperados com as interações. (SCHWABER; SUTHERLAND, 2011).

Cada cerimônia é uma oportunidade onde os papéis envolvidos possam inspecionar e adaptar algo. Mantém transparência e alinha todo trabalho que está sendo feito.

2.1.2 PAPÉIS

Uma equipe de Scrum é formada por três papéis diferentes, são esses: Time de desenvolvimento (*Team*), Dono do produto (*Product Owner*) e o Scrum Master. As equipes de Scrum são auto-organizadas, ou seja, são capazes de tomar as rédeas e escolherem a melhor forma de se realizar os trabalhos que devem ser desempenhados, ao invés de serem dirigidos por pessoas de fora da equipe. (SCHWABER, 2004).

Equipes de Scrum são trans-funcionais, que consiste em equipes que possuem diversas competências que devem ser somadas, uma equipe completa não deve depender de outros que não fazem parte da equipe. Uma equipe modelo de Scrum deve saber aperfeiçoar a flexibilidade, criatividade e produtividade. (SCHWABER; SUTHERLAND, 2011).

De forma interativa e incremental é como trabalha uma equipe de Scrum, aproveitando ao máximo as possibilidades de se realimentarem de informações para realizar melhorias na maneira como procedem trabalhando e principalmente no entregável ao cliente. (SCHWABER, 2004; SCHWABER; SUTHERLAND, 2011).

Como disse Schwaber e Sutherland (2011, p.5):

As equipes do Scrum produzem iterativamente e incrementalmente, aproveitando ao máximo as oportunidades para realimentação. Entregas incrementais de produtos “Prontos” garantem que uma versão potencialmente útil do produto esteja sempre disponível.

O Scrum está baseado nas teorias empíricas de controle de processo. O empirismo afirma que o conhecimento vem da experiência e tomar decisões baseados no que se conhece. O Scrum aplica uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos. Três pilares sustentam a implementação de um controle de processo empírico: transparência, inspeção e adaptação. (SCHWABER, 2004).

2.1.3 O TIME DE DESENVOLVIMENTO (*TEAM*)

O time de desenvolvimento é responsável pelo gerenciamento e desenvolvimento de suas próprias atividades. Tradicionalmente, o gerente de projetos que dita como o time deve trabalhar, o que deve fazer e gerencia este trabalho.

No Scrum é a equipe de desenvolvimento que seleciona o trabalho que deverá ser desempenhado a cada iteração (Sprint). Em seguida a própria equipe que deve definir como será desempenhado este trabalho do qual assumiu responsabilidade. Decidindo como os requisitos serão incrementalmente feitos, visando sempre ter algo funcional para entregar ao cliente no final da iteração (Sprint). (SCHWABER, 2004).

2.1.4 O DONO DO PRODUTO (*PRODUCT OWNER*)

O Dono do produto deve garantir o retorno sobre o investimento que está sendo feito (ROI). Ele é quem sabe dos requisitos e quem vai dizer se os mesmos estão sendo atendidos durante as interações. (SCHWABER; SUTHERLAND, 2011).

Mantém um contato direto com os demais papéis da equipe de Scrum e atua efetivamente colaborando com o projeto, participando de diversas cerimônias que acontecem e principalmente validando os trabalhos desempenhados pelo time de desenvolvimento. (SCHWABER, 2004).

2.1.5 SCRUM MASTER

O Scrum Master é responsável por remover qualquer barreira entre o Time de desenvolvimento e o Dono do produto. É a peça responsável por mostrar ao Dono do produto como usar o Scrum para maximizar o retorno do investimento(ROI) do projeto e cumprir com os objetivos do projeto. Sua autoridade existe mas se dá de forma altamente indireta no processo. (SCHWABER, 2004; SCHWABER; SUTHERLAND, 2011).

Deve saber todas as regras e práticas do Scrum e fazer com que as mesmas sejam seguidas. É o responsável pelo sucesso do projeto e ajuda a melhorar a probabilidade de sucesso auxiliando o Dono do produto a definir itens de maior prioridade e que irão de fato trazer mais resultados e agregar valor a cada iteração do projeto. (SCHWABER, 2004; SCHWABER; SUTHERLAND, 2011).

2.1.6 ARTEFATOS

Existem alguns artefatos que são utilizados nos ciclos do Scrum. Estes documentos auxiliam no processo, se tornando uma fonte de dados para todos os papéis de uma equipe Scrum.

Não existe uma prescrição que obrigue o uso destes artefatos porém é um recurso comumente usado no mercado de trabalho por empresas que trabalham com as práticas ágeis. Justamente pela simplicidade e conteúdo relevante que possuem. (SCHWABER, 2004).

2.1.7 PRODUCT BACKLOG

Documento que contém todos os requisitos do produto em forma de lista. O papel responsável por manter este conteúdo é do Dono do produto, ele deve deixar todas as funcionalidades explícitas e organizadas de maior prioridade para menor. O Product Backlog é utilizado em diversas cerimônias do Scrum para dar uma visibilidade de todas as funcionalidades que o produto exige e para auxiliar a equipe em seus planejamentos. (SCHWABER, 2004).

2.1.8 SPRINT BACKLOG

O Sprint Backlog possui o trabalho que o time de desenvolvimento deverá realizar durante o ciclo da *sprint*. Estes itens são parte da lista geral de funcionalidades do sistema, o *Product Backlog*. (SCHWABER, 2004).

2.1.9 GRÁFICO BURNDOWN

Na Figura 2 mostra de forma gráfica e simples como está o andamento da sprint, é um gráfico com eixos x e y onde o eixo x é tempo e y o volume de atividades a serem desempenhadas. (KNIBERG, 2007).

“Em Scrum, gráficos de burndown do sprint são usados como uma das principais ferramentas para acompanhar o andamento de uma iteração.” (KNIBERG; SKARIN, 2007, p.68).

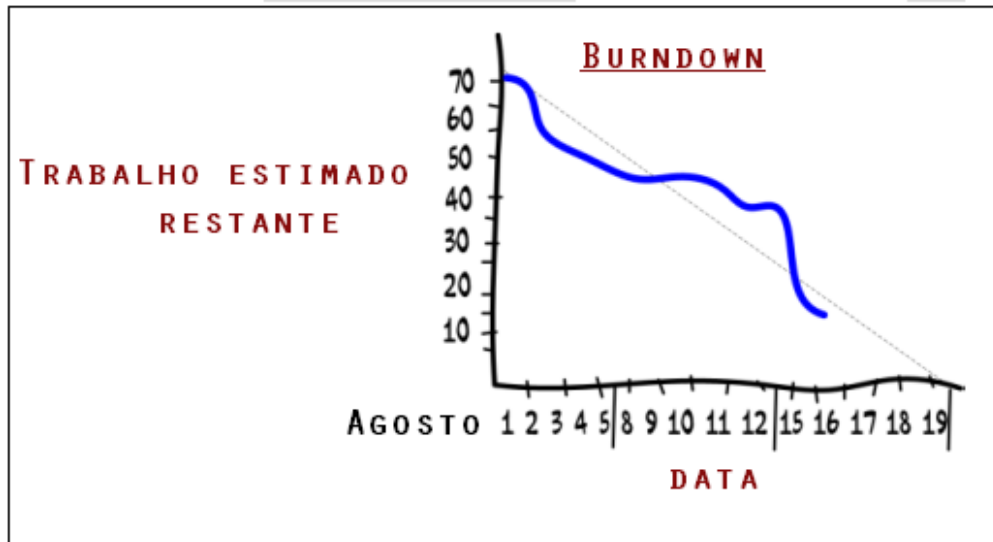


Figura 2 – Gráfico Burndown.
Fonte: Adaptado de Kanban e Scrum pg. 68

2.2 EXTREME PROGRAMMING

Extreme Programming chamado também de XP, é um processo de desenvolvimento de *software* que segue alguns valores comuns das práticas ágeis sendo englobado também dentro desta categoria. (BECK, 2004).

Compartilha da ideia de que o cliente aprende sobre suas necessidades conforme é capaz de experimentar as mesmas e com base neste *feedback* são reavaliadas as necessidades e prioridades, gerando constantes mudanças que devem ser consideradas e incluídas no produto. (BECK, 2004).

O *Extreme Programming* possui quatro valores fundamentais: *Feedback*, comunicação, simplicidade e coragem. Onde o aprendizado do cliente com o sistema é utilizado para gerar *feedback*. As dúvidas que a equipe tem no decorrer do desenvolvimento são sanadas com uma boa comunicação gerando agilidade no processo. A equipe implementa de forma simples somente o que é suficiente para

atender a necessidade do cliente. A equipe é confiante e acredita que com estes princípios é capaz de se desenvolver o produto e evoluir com segurança e agilidade. (BECK; FOWLER; MARTIN, 2000).

Existem diversas práticas prescritas pelo *Extreme Programming*, tais como: Cliente presente, Jogo do planejamento, *Stand up meeting*, programação em par, desenvolvimento guiado pelos testes, *refactoring*, código coletivo, código padronizado, *design* simples, metáfora, ritmo sustentável, integração contínua e *releases* curtos. (BECK; FOWLER; MARTIN, 2000).

Segundo Beck, Fowler e Martin (2000, p. 33) na equipe definida pelo XP existem os papéis de:

- Gerente de projetos, responsável pelos assuntos administrativos do projeto.
- Coach, responsável técnico que guia os desenvolvedores mas também codifica.
- Analista de teste ajuda o cliente a escrever os testes de aceitação e está ligado à qualidade do produto.
- Redator técnico ajuda a equipe na documentação do sistema.
- Desenvolvedor, analisa, projeta e codifica o sistema.

2.3 KANBAN

Kanban significa literalmente “cartões visuais” em Japonês. No mercado de trabalho inicialmente foi usado para controlar os trabalhos de chão de fábrica pela Toyota. É visto como uma ferramenta que auxilia no controle e visualização de informações do projeto que está usando-a, pelas suas características enxutas e muito úteis. (KNIBERG, 2007).

A Figura 3 apresenta um quadro Kanban e mostra que ajuda na visualização do trabalho em andamento, limita a quantidade de trabalho em progresso (WIP - *Work in progress*) e ajuda no acompanhamento de tempo de execução das tarefas.

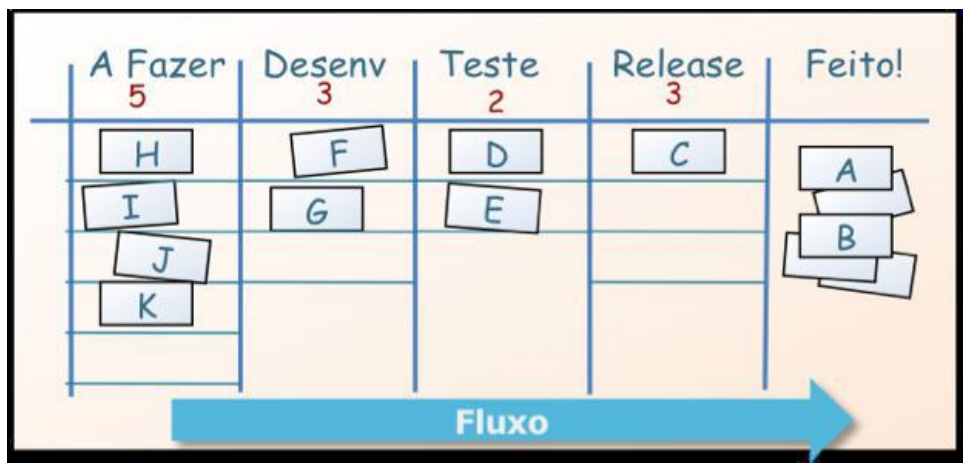


Figura 3 – Quadro Kanban.
Fonte: Kanban e Scrum pg. 25

É uma ferramenta nova para equipes desenvolvedoras de *software* e que seguem práticas ágeis como Scrum e XP. Porém, o Kanban para sistemas enxutos (*Lean Production*) de produção já possui mais de meio século de uso. O Kanban é tido como um aliado para auxiliar o funcionamento adequado de práticas como XP e Scrum que foram abordadas neste trabalho. (KNIBERG; SKARIN, 2007).

3 DESIGN: CONCEITOS E ABORDAGEM PARA SOFTWARE

A palavra *design* possui várias definições que podem gerar ambiguidade na interpretação. Este é um dos motivos que tornam hoje o conceito de *design* no mercado de trabalho um pouco distorcido. Com frequência a figura do *designer* é referenciada apenas como um “artista” que faz “belos *designs*”, porém, considerar somente a estética do produto se torna uma visão muito restrita a toda abrangência do *design*. (LOBACH, 2001)

“Nos últimos 2 séculos, o poder humano de controlar e dar forma ao ambiente que nos cerca e no qual habitamos aumentou progressivamente: hoje, é lugar-comum falar de um mundo feito pelo homem. O instrumento desta transformação foi a indústria mecanizada: de suas oficinas e fábricas saiu um verdadeiro dilúvio de artefatos e mecanismos para satisfazer as necessidades e desejos de parte cada vez maior da população mundial. A mudança não foi apenas quantitativa, mas alterou também radicalmente a natureza qualitativa da vida que vivemos ou desejamos viver”. (HESKETT, 1998).

Lobach (1996, p. 23) disse que *design*: “é um processo de adaptação do ambiente artificial às necessidades físicas e psíquicas dos homens na sociedade [...]”.

Para Schulmann (1994) o *design* é um método criador, integrador e horizontal que considera a figura do *designer* como um especialista responsável pela análise e resolução de problemas ligados ao desenvolvimento de um novo produto.

Norman (2002, p. 44) afirmou que: “Design é a aplicação sucessiva de restrições até que resulte em apenas um único produto.”

No dicionário existem diversas opções, tais como: Projeto, plano. Esboço, desenho, croqui. Construção, configuração, modelo. Ou seja, consiste em uma ideia, projeto ou plano de solução para um problema determinado. Como também se refere a um objeto concreto resultante da ideia propriamente dita. (LOBACH, 2001).

Buscando a origem desta palavra encontra-se uma definição no latim com o verbo *designare* que tem os sentidos de designar e desenhar. Esta etimologia da palavra por si só já causa ambiguidade com duas diferentes interpretações. Uma é de algo abstrato, assim como: conceber, projetar e atribuir. A outra é a ideia de concreto, de registrar, configurar ou formar. (DENIS, 2000).

Não existe uma única definição para *design*, por se tratar de uma disciplina flexível e que possibilita diferentes interpretações é possível que sofra modificações ao longo do tempo. O que fica claro é a questão de que o *design* surgiu para criar maior qualidade de vida ao homem usando de forma conjunta as linguagens visuais e funcionalidades. (COUTO e OLIVEIRA, 1999).

3.1 ERGONOMIA

Disciplina científica relacionada à compreensão das interações entre o ser humano e outros elementos de um sistema, é a profissão que aplica teoria, princípios, dados e métodos para projeto de forma a otimizar o bem-estar do ser humano e a performance geral do sistema. (ISO 6385, 2004).

Apesar de Henry Dreyfuss aplicar, desde a década de 1930, uma metodologia ergonômica em seu trabalho, HIRATSUKA (1996) relata que o termo “ergonomia” foi criado e utilizado pela primeira vez pelo inglês Murrel, posteriormente adotado oficialmente em 1949, durante a criação da primeira sociedade de ergonomia, a Ergonomics Research Society (Sociedade de Pesquisa em Ergonomia). Segundo DUL e WEERDMEESTER (2004), a ergonomia desenvolveu-se durante a II Guerra Mundial, quando houve, pela primeira vez, a união de esforços entre a tecnologia, ciências humanas e biológicas, para resolver problemas causados pela operação de equipamentos militares complexos. O trabalho conjunto de médicos, psicólogos, antropólogos e engenheiros gerou resultados gratificantes, posteriormente

aproveitados nos projetos desenvolvidos pela indústria no pós-guerra.” (BRANDAO, 2006, p. 35).

Teve crescimento alavancado pelo interesse de psicólogos, fisiologistas e engenheiros em estudar a relação entre trabalho e o homem. Ressaltando a palavra “trabalho” que demonstra que não é apenas a relação entre pessoas com máquinas, mas sim, entre toda situação em que existe um relacionamento do homem com seu trabalho. As mudanças e adaptações sempre são feitas neste sentido, do trabalho para o trabalhador, ajustando-o para as capacidades e limitações humanas. (HIRATSUKA, 2006).

A Ergonomia objetiva modificar os sistemas de trabalho para adequar a atividade neles existente às características, habilidades e limitações das pessoas com vistas ao seu desempenho eficiente, confortável e seguro. (ABERGO, 2000).

A etimologia da palavra vem do grego, onde *Ergon* significa trabalho e *Nomos* significa normas, regras ou leis. Entendendo-se como uma ciência que aplica mudanças em máquinas, equipamentos, sistemas ou tarefas e causam melhorias na saúde, segurança, conforto e eficiência no trabalho. (WEERDMEESTER, 2004).

Assim como *design*, a ergonomia tem um conceito amplo, que possui aspectos abstratos e concretos. Com o crescimento do uso de *software* para otimizar o trabalho surgiram estudos para trazer maior qualidade nas aplicações e como consequência gerar maior satisfação nos usuário. Como mostra a Interação Humano-Computador.

3.2 INTERAÇÃO HUMANO-COMPUTADOR

É um campo de estudos interdisciplinar com o objetivo geral de entender como e o porque as pessoas utilizam (ou não utilizam) a tecnologia da informação. O termo IHC começou a ser usado em meados dos anos 1980, como uma referência a um novo campo de estudos. No qual a principal preocupação é em como o uso dos computadores pode enriquecer a vida pessoal e profissional dos seus usuários. (MORAES; ROSA, 2010).

O principal objetivo desta área de interação humano-computador é o projeto (*design*) e desenvolvimento de sistemas que proporcionem maior eficácia e satisfação ao usuário. Segundo Moraes e Rosa (2010, p. 12) é a área que se destina ao estudo de como projetar, implementar e utilizar sistemas computacionais interativos e como os computadores e sistemas afetam indivíduos, organizações e sociedades.

Projetar um sistema que cumpra com funcionalidades necessárias para o cumprimento do trabalho não é o suficiente. Deixar o usuário com a responsabilidade de aprender e usar um sistema complexo é algo que precisa ser minimizado. A interface deve ser fácil de usar, trabalhar de maneira prevista e ter consistência durante toda interação. (NORMAN, 2003).

Segundo Preece (1994, p. 52), os objetivos da interação humano-computador são de desenvolver e aprimorar sistemas computacionais nos quais os usuários possam executar suas tarefas com segurança, eficiência e satisfação. Esses aspectos de forma coletiva são conhecidos como usabilidade.

3.3 USABILIDADE

O termo usabilidade foi cunhado para substituir o termo “amigável ao usuário” que no início dos anos 1980, acabou adquirindo conotações indesejavelmente vagas e subjetivas. (MORAES; ROSA, 2010).

Para Nielsen (2003), a usabilidade é um atributo qualitativo que determina quão fácil é usar as interfaces do usuário. O termo usabilidade também se refere aos métodos para facilitar o uso durante o processo de *design*.

Usabilidade é a capacidade de um produto ou sistema, em termos funcionais-humanos, de ser usado com facilidade e eficácia por um segmento específico de usuários, fornecendo-lhes treinamento e suporte específico, visando à execução de um elenco específico de tarefas, no contexto de cenários ambientais específicos. (MORAES; ROSA, 2010).

Segundo a norma ISO 9241-11 (2002) usabilidade é a “capacidade que um sistema interativo oferece a seu usuário, em um determinado contexto de operação, para a realização de tarefas de maneira eficaz, eficiente e agradável”. Já segundo a norma ISO/IEC 9126 (2002 apud MORAES; ROSA, 2010, p. 15) usabilidade é a “facilidade com que um usuário pode aprender a operar, preparar entradas para e interpretar as saídas de um sistema ou componente”.

Usabilidade é um atributo do produto assim como a funcionalidade. A funcionalidade é o que o produto pode fazer, sendo assim, quando é feito um teste de funcionalidade está sendo certificado se o produto faz o que está na especificação. Já a usabilidade se refere em como o usuário interage com o produto. Um teste de usabilidade implica em verificar se o usuário é capaz de reconhecer e interagir com funções que satisfaçam suas necessidades. (DUMAS; REDISH, 1999 apud MORAES; ROSA, 2010).

É importante ressaltar que usabilidade não é uma simples propriedade da interface do usuário. Usabilidade tem múltiplas propriedades e é tradicionalmente associada com cinco atributos chaves como comenta NIELSEN (2003):

- **Aprendizado:** O sistema deve ser fácil de aprender para que o usuário possa rapidamente conseguir cumprir com seus objetivos usando o sistema.
- **Eficiência:** O sistema deve ser eficiente, assim que o usuário aprender a usá-lo ele possa realizar suas tarefas com um alto grau de produtividade.
- **Memorização:** O sistema deve possibilitar uma maneira fácil do usuário recordar os passos que realiza, podendo ficar um tempo sem usar o sistema e quando voltar a mexer novamente seja possível lembrar de forma simples como cumprir suas tarefas, ao invés de precisar aprender novamente.

- Erros: O sistema deve ter uma forma agradável de tratar os erros e os mesmos devem possibilitar com que o usuário consiga corrigir e evitar com que aconteçam novamente.
- Satisfação: Os usuários devem estar satisfeitos quando usarem o sistema e permanecerem dispostos a usar novamente.

A partir da década de 90 o mercado de tecnologia de informação começou a empregar com mais forças métodos de usabilidade para projetar e testar seus sistemas. O aparecimento de testes, laboratórios de usabilidade e materiais de estudo sobre usabilidade mostram que mesmo sendo algo de extrema importância para o produto, muitos desenvolvedores e gerentes de projetos demoraram para incluir de fato em seus processos de testes, muitas vezes devido a baixo orçamento ou pressões de tempo dos investidores. (MORAES; ROSA, 2010);

O retorno positivo dos projetos que utilizam os métodos de usabilidade é de fato a prova de que um projeto de sistema que tem o usuário como foco principal, tende ao sucesso. E isso está totalmente ligado a satisfação do usuário e do investidor. Esses indicadores fazem com que cada vez mais o mercado de tecnologia de informação seja aderente a esses métodos e faça investimentos na área.

Ao considerar o ser humano como um elemento fundamental no sistema, a tecnologia deve ser apenas um meio para se atender as necessidades e características humanas. A tecnologia não existe isoladamente, ela é influenciada pelo usuário e o usuário por ela, é um ciclo interativo de uso. (SANTOS, 2004 apud MORAES; ROSA, 2010).

3.4 ENGENHARIA DE USABILIDADE

Somente com uma definição de usabilidade é que se torna possível chegar a uma disciplina de engenharia, onde a usabilidade não é apenas argumentos e conceitos e sim pode ser sistematicamente abordada, melhorada e avaliada (possivelmente medida). Mesmo se a intenção não for realizar estudos formais de

medição da usabilidade, esclarecer estes aspectos é muito mais interessante do que simplesmente achar que a interface está com uma boa usabilidade. (SHACKELL, 1991 apud NIELSEN, 2003).

Segundo o dicionário Aurélio, engenharia é a “arte de aplicar conhecimentos científicos e empíricos e certas habilitações específicas a criação de estruturas, dispositivos e processos que se utilizam para converter recursos naturais em formas adequadas ao atendimento das necessidades humanas”.

A Engenharia de Usabilidade visa o desenvolvimento da interação entre o usuário e sistemas informatizados. A engenharia de usabilidade tem por objetivo oferecer técnicas e métodos que possam ser utilizados sistematicamente para assegurar um alto grau de usabilidade da interface de programas de computador. (PADUA, 2011).

O envolvimento da engenharia de usabilidade com engenharia de *software* acaba não sendo tão claras, mas por definição é possível afirmar que engenharia de *software* trata de aspectos construcionais de sistemas de *software* enquanto a engenharia de usabilidade cuida dos aspectos comportamentais, que são relacionados a interação humano-computador (NIELSEN, 2003).

A engenharia de usabilidade não acontece momentos antes de gerar *release* de um produto. De fato, está envolvida com varias atividades durante o ciclo de vida do produto, possuindo atividades significantes até mesmo antes de a interface ser efetivamente desenhada.

A usabilidade não deve ser considerada parte isolada na produção de um produto, pois, existem atividades que auxiliam nas definições do *design* do produto. Considerando que essas definições de *design* quando não envolvem a preocupação com usabilidade podem sofrer do efeito dominó. Podendo refletir em incompatibilidades entre *releases* e demais necessidades que não foram consideradas no início do desenvolvimento (NIELSEN, 2003).

Como disse Grudin et. al. 1987 apud NIELSEN 2003, o envolvimento de fatores humanos em um determinado produto pode vir a ter o maior impacto sob lançamentos futuros.

4 PROPOSTA DE INTERAÇÕES E ENTREGÁVEIS COM FOCO NO DESIGN (AGILE USER-CENTERED DESIGN)

O uso de práticas ágeis em comum trata o cliente como uma fonte de requisitos para o produto, pois é logicamente a pessoa que conhece o negócio em questão. O problema é que na maior proporção, o cliente não sabe o que o negócio quer de fato.

Quando se trabalha centrado no usuário é possível mitigar alguns riscos e fatos que acontecem comumente pelo mal uso de práticas ágeis, ou necessidades que o usuário acaba não revelando no levantamento de requisitos. As técnicas apresentadas acima e demais existentes no mercado, quando bem utilizadas no ciclo de desenvolvimento de um produto, conseguem auxiliar na abstração dos problemas e necessidades que os usuários possuem. Como consequência reflete com uma maior qualidade do produto e satisfação do usuário. (SY, 2007).

Práticas ágeis têm pontos que vão de encontro as práticas de *design*, ambos trabalham com forte *feedback* do usuário, fazendo com que tenha maior sucesso nos entregáveis e *releases*. Uma vez que o próprio usuário é quem valida e dá seu *feedback*.

Esse *feedback* acontece no final de um ciclo de uma iteração, que como já foi comentado é finalizada entre duas a quatro semanas na Sprint do Scrum. O encerramento de uma Sprint implica em ter algo funcional para validar com o cliente e com isso é inevitável que o cliente encontre mudanças no que já foi feito.

Por mais que um ciclo é um período de pouco tempo e é possível identificar e realizar mudanças pertinentes ao produto nesse período fazendo com que não comprometa a release, que é o resultado de vários ciclos rodados. Mesmo assim, existe um retrabalho que muitas vezes é responsável por estourar prazos e planejamentos. (MILLER, 2005).

Algumas técnicas do *design* centrado no usuário devem acontecer antes da codificação, auxiliando a evitar o máximo de retrabalho. Porém, hoje no mercado é

comum encontrar empresas de tecnologia que utilizam práticas ágeis e envolvem seus profissionais de *design* somente momentos antes da entrega de um *release*.

4.1 MÉTODOS DE *DESIGN*

Efetivamente no levantamento de requisitos com o usuário é necessário um arsenal de métodos diferentes, onde os mesmos exigem conhecimento para serem aplicados e conseguirem obter êxito, que consiste em levantar o maior número de informações que possam agregar ao projeto. (MILLER, 2005).

Os *designers* de interação são profissionais responsáveis por executar as atividades ligadas ao *design* do produto dentro da equipe do projeto. Alguns métodos mais usados por eles serão apresentados.

4.1.1 INVESTIGAÇÃO DE CONTEXTO (*CONTEXTUAL INQUIRY*)

Técnica de pesquisa estruturada para estudos etnográficos. É de natureza investigativa. Usado para descobrir como é o ambiente de trabalho, como o usuário faz o seu trabalho, quais suas metas de trabalho, quais problemas tem que resolver, quais dados são produzidos e como o trabalho do usuário é avaliado. Não determina como o produto será ou quão fácil será seu uso. (GARRET, 2011).

4.1.2 ENTREVISTAS (*INTERVIEWS*)

Acontecem com seções de perguntas e respostas. A intenção é ganhar compreensão de áreas que não estão tão claras. Geralmente na maioria as perguntas são abertas a conversação e não fechadas e quantitativas. (FERREIRA, 2007).

Ajuda a ver o que o usuário gosta, quais problemas estão acontecendo e o que ele pretende em versões futuras do produto. As entrevistas não ajudam a identificar oportunidades de mercado e nem determinam como será o uso do produto. (MILLER, 2005).

4.1.3 PERSONAS

Coletar dados sobre seus usuários podem ser incrivelmente valioso, mas às vezes você pode perder de vista as pessoas reais por trás de todas as estatísticas. É possível tornar usuários mais reais transformando-os em personas (às vezes chamados de modelos de usuários ou perfis de usuário). (MORAES; ROSA, 2010).

Persona é um personagem fictício construído para representar as necessidades de toda uma gama de usuários reais. Ao colocar um rosto e um nome com os pedaços desconexos de dados de sua pesquisa com usuários e trabalho de segmentação, personas pode ajudar a garantir que você mantenha os usuários em mente durante o processo de design. (GARRET, 2011).

4.1.4 TESTES DE USABILIDADE

Excelente para obter *feedback* e encontrar problemas com aprendizagem, descoberta, taxas de erros e velocidade de uso. São usados para avaliação de um projeto do produto, observando os usuários na experiência com o protótipo e verificando a utilização do que tem feito com o que se propõem a fazer. (NIELSEN, 2003).

Auxilia na descoberta de itens que foram omitidos em prováveis entrevistas com o usuário. Podendo revelar características ausentes que são necessárias para completar um fluxo de trabalho. Não verifica se o produto se encaixa no ambiente do usuário até mesmo porque os testes são executados na maioria fora deste ambiente. (NIELSEN, 2003; MILLER, 2005).

4.1.5 GRUPO FOCO

Técnica ou método qualitativo de captura e obtenção de dados cuja principal finalidade é extrair de atitudes e respostas dos participantes, sentimentos, crenças e

opiniões a respeito do objeto em estudo. É uma discussão estruturada e planejada sobre tópicos específicos estabelecidos em um roteiro de discussão e conduzido por um moderador. (MORAES, 2002; GARRET, 2011).

Serve como meio de testar conceitos novos, produtos e mensagens. Vislumbra-se a compreensão de expectativas desejos e dificuldades encontradas pelo usuário. (MORAES, 2002).

4.1.6 PROTOTIPAGEM

Um protótipo é uma representação limitada de um *design* que permite aos usuários interagir com ele e explorar a sua conveniência. Pode ser imaginado como um modelo em escala menor ou parte de um *software* em desenvolvimento. (FERREIRA, 2007).

São usados para responder dúvidas e darem suporte aos designers na escolha de uma dentre várias opções. Testam a viabilidade técnica de uma ideia, esclarecem alguns requisitos vagos, possibilita realizações de testes e avaliações com os usuários e pode verificar decisões de design estão compatíveis com o resto do desenvolvimento do sistema. (MORAES; ROSA, 2010).

4.1.7 AVALIAÇÃO HEURÍSTICA

Método de inspeção para encontrar determinados tipos de problemas em uma interface do usuário. Envolve um especialista de usabilidade examinando a interface e julgando sua adequação com base em princípios de usabilidade reconhecidos que foram expostos no capítulo 3.3 deste trabalho. (NIELSEN, 2003).

Conforme Brinck, Gergle e Wood (2002) apud Moraes e Rosa (2002), a avaliação heurística pode ser realizada ainda na fase de projeto, antes que se faça qualquer escolha de design e antes que o cliente tenha visto seu trabalho. Depois que o design tenha sido realizado, mas antes que o projeto tenha sido construído. Como uma verificação final de qualidade, antes que o *software* seja implantado.

Trata de problemas na utilidade, que está relacionado com características que impedem do usuário efetuar com êxito alguma tarefa desejada. Problemas informacionais, que dizem respeito à apresentação, agrupamento e legibilidade de informações. E problemas de usabilidade que podem afetar a habilidade do *software* em permitir que o usuário alcance seu objetivo. (MORAES; ROSA, 2010).

4.2 CENÁRIO DE DESENVOLVIMENTO COM INTERAÇÕES DE DESIGN

Como o desenvolvimento ágil tem foco em poucas funcionalidades novas por etapa. Isto significa que os profissionais de *design* não precisam trabalhar em todo *design* do release ao mesmo tempo. Ao invés disso, podem focar nos *designs* mais importantes para o momento.

A cada tempo dado durante um ciclo, é possível conduzir atividades chave de usabilidade para itens mais complexos do *release*. Como os *designers* e desenvolvedores estão trabalhando juntos no mesmo subconjunto de funcionalidade de uma só vez, significa que todas as características que exigem trabalho de design centrado no usuário terão sua atenção focada. (SY, 2007).

Considerando que todas as funcionalidades deverão estar implementadas, trabalhar parte dessas funcionalidades não impede que se tenha um bom resultado de *design*. O principal para se chegar a um bom design é o processo do design de interação que possibilita identificar falhas e propor outras soluções o mais cedo possível dentro do projeto. (MILLER, 2005).

Não se trata só de conduzir testes de usabilidade para checagem dos protótipos, mas fazê-los antes da codificação ser iniciada, ou seja, enquanto o design ainda está maleável. Ao se iniciar um ciclo de desenvolvimento a codificação é imediata e é preciso encontrar uma maneira para separar iterações de design das iterações de implementação para melhor controle e execução dessas atividades. (SY, 2007).

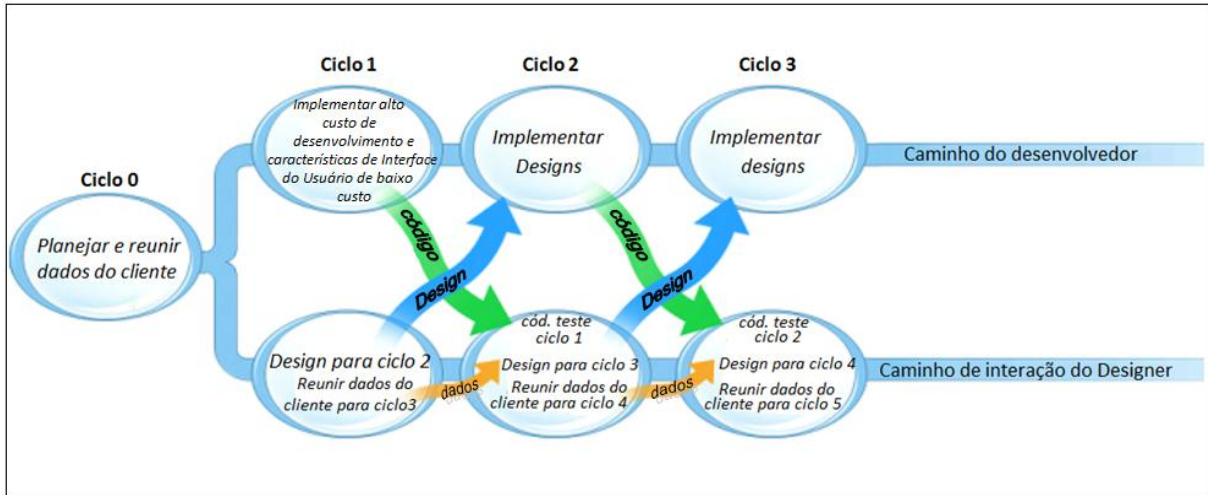


Figura 4 – Proposta de iteração.

Fonte: Adaptado de *Adapting Usability Investigations for Agile User Centered-Design* (SY, 2007, p. 118)

Este cenário de estudo sugere separar o trabalho em dois processos, um rodando para o *design* de interação e outro para os desenvolvedores que trabalham em paralelo com outras atividades. O trabalho de interação é separado da codificação, porém pode acontecer em simultâneo como é mostrado na figura 3. (MILLER, 2005).

Como elencado por Sy (2007, p. 117) as atividades de usabilidade para o ciclo 0, geralmente são:

- Obter dados para refinamento dos requisitos. Alinhando todo time com os objetivos do produto.
- Grupo foco, explorando idéias e validando o mercado.
- Análise, sumarização dos resultados das entrevistas e investigações de contexto e levantamento de alguns dados sobre usabilidade.
- Criar personas e cenários para melhor compreensão e entendimento dos usuários.

Neste ciclo geralmente os desenvolvedores trabalham em um nível arquitetural da aplicação e no ciclo 1 podem continuar com algumas atividades dessas e também iniciar com atividades arquiteturais ou atividades que exijam menos de *design*. (SY, 2007).

Para o ciclo 1 as atividades de usabilidade são expostas por Sy (2007, p.118):

- Desenhar protótipos para o ciclo 2 e conduzir testes de usabilidade e avaliação heurística para refinamento do *design*.
- Conduzir entrevistas e fazer investigação contextual para funcionalidades do ciclo 3.

No ciclo 2, o *design* do ciclo 1 é apresentado para os desenvolvedores quem começam a implementação. A comunicação é importante para que *designers* de interação possam transmitir todas as definições para os desenvolvedores. E as atividades de usabilidade para o ciclo 2 são propostas por Sy (2007, p.118):

- Prototipagem e testes de usabilidade do ciclo 3, usando as informações coletadas no ciclo 1.
- Investigação de contexto para o ciclo 4.
- Testando usabilidade do trabalho implementado do ciclo 1.

Deste modo segue até a *release* do produto, com o *design* sempre um ciclo a frente do desenvolvimento e o refinamento de requisitos pelo menos 2 ciclos na frente.

A prática de criar uma experiência de usuário atraente e eficiente é chamada de *design* centrado no usuário. O *design* centrado no usuário é muito simples: Levar o usuário em consideração a todo passo dado no desenvolvimento do produto. As implicações deste simples conceito são surpreendentemente complexas. (GARRETT, 2011).

4.3 A EXPERIÊNCIA DO USUÁRIO

A experiência do usuário se refere à experiência que o produto causa em uma pessoa que está em contato com o produto no mundo real. Quando se inicia o desenvolvimento do produto as pessoas despendem a atenção para o que o produto

faz. A experiência do usuário trata o “como ele funciona”. Este aspecto tem grande influência no sucesso ou falha de um produto. (GARRETT, 2011).

Não se trata da experiência dos envolvidos no desenvolvimento do projeto, se trata da experiência que acontece fora, onde as pessoas terão o contato com o produto. O fato é que todo produto ou serviço usado por alguém promoverá uma experiência, assim como: livros, cadeiras, self-services, *softwares*, potes de mostarda ou qualquer outro produto ou serviço. (GARRETT, 2011, p. 8).

São as coisas simples que importam. Quando se aperta um botão não parece ser muito, mas quando este botão faz a diferença entre realizar ou não uma grande transação, importa muito. Caso algum problema aconteça durante essa experiência, o usuário ficará frustrado e provavelmente não indicará o produto e não usará novamente se houver outros produtos que ofereçam o mesmo conteúdo, porém proporcionem melhores experiências. Trazendo grandes conseqüências para o sucesso do produto.

A grande razão da importância da experiência do usuário é porque é importante para ele. Se não proporcionar uma boa experiência ao usuário o produto não terá seu retorno investido. (GARRETT, 2011).

5 PROCEDIMENTOS METODOLÓGICOS DA PESQUISA

5.1 LOCAL DO ESTUDO

O município de Foz do Iguaçu está localizado no extremo oeste do Paraná, na fronteira do Brasil como o Paraguai e a Argentina – latitude sul $25^{\circ} 32' 45''$ longitude oeste $54^{\circ} 35' 07''$. A figura 1 ilustra a localização do Município de Foz do Iguaçu dentro do estado do Paraná.

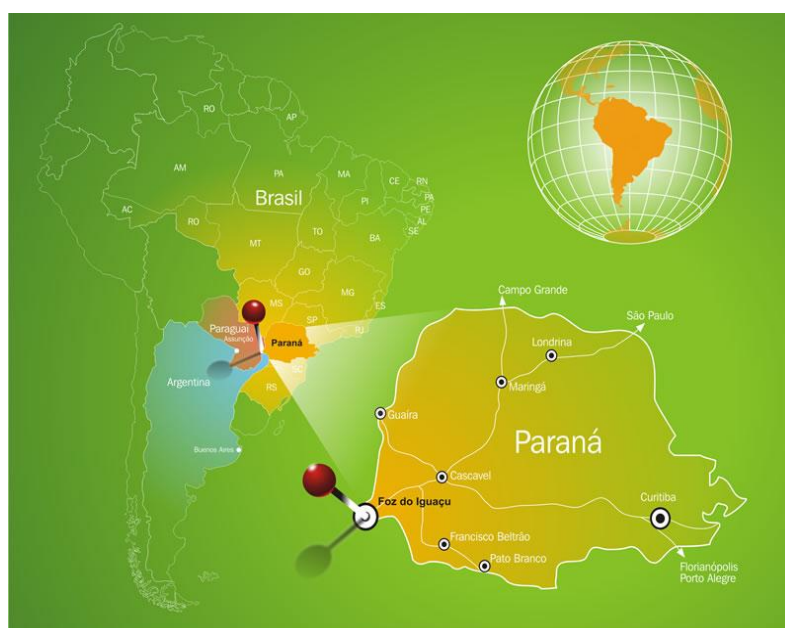


Figura 5 – Localização Geográfica do Município de Foz do Iguaçu
Fonte: Prefeitura Municipal de Foz do Iguaçu (2009).

5.2 TIPO DE PESQUISA OU TÉCNICAS DE PESQUISA

Foram usadas a técnica de observação direta para levantamento dos dados.

5.3 COLETA DOS DADOS

Foram levantadas fontes de dados secundárias como livros, blogs, vídeos e artigos na internet além de dados primários que foram obtidos com a própria experiência e entrevistas com outros gerentes de projetos de empresas da região. E

o estudos de casos da empresa Autodesk que foram publicado por Miller em 2005 e Desireé Sy em 2007.

5.4 ANÁLISE DOS DADOS

Através de anotações e reflexões feitas a partir das informações obtidas com a coleta de dados foi realizada a centralização dos dados e elaborado um mapa mental para estruturação da ideia, conforme mostra a Figura 6.

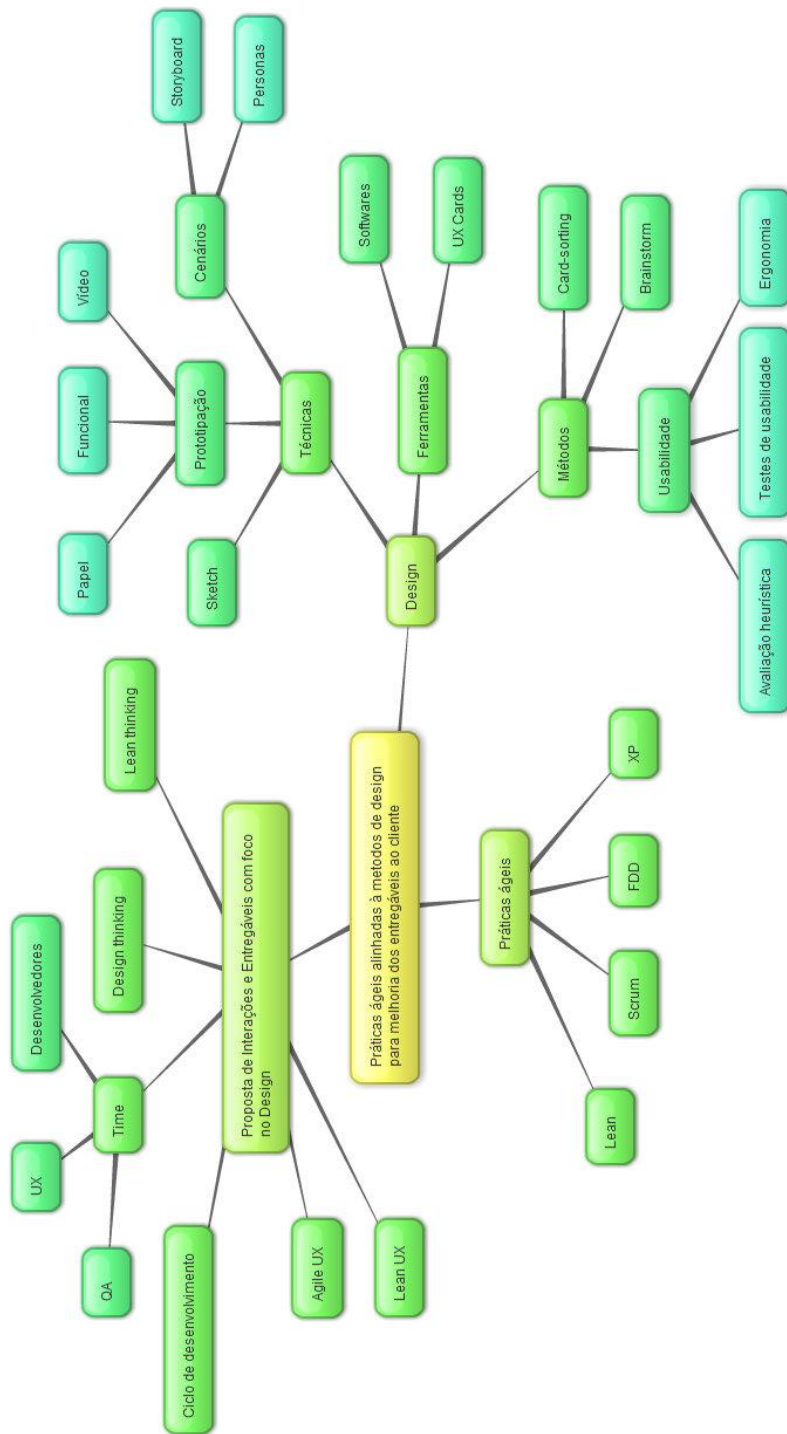


Figura 6 – Mapa mental da pesquisa

6 RESULTADOS E DISCUSSÃO

O conteúdo sintetizado de conceitos e práticas de duas áreas distintas que foram apresentadas no trabalho, mostram a abrangência do desenvolvimento ágil e do *design*. Deixando claro que ambos têm pontos em comum, mas também possuem algumas divergências. Justamente pelo fato de serem áreas que envolvem competências diferentes.

Não existe uma fórmula para sucesso nos projetos de *software*, pois existem muitas variáveis envolvidas. Porém, a junção de usabilidade, desenvolvimento ágil e práticas de *design*, quando bem aplicadas podem aumentar as chances de sucesso do projeto de *software*, refletindo relativamente na qualidade do produto.

Na essência os princípios ágeis, expostos no manifesto ágil do Capítulo 1 possuem grande abrangência aos conceitos expostos no Capítulo 2 sobre *design*. São duas áreas de estudos com raízes diferentes e que pregam uma filosofia parecida, essa junção das duas áreas trazem benefícios em muitos pontos.

As iterações das práticas ágeis quando estão centradas no usuário fazem com que mais do design do produto esteja pronto o quanto antes e possibilitam a execução de vários métodos de design dentro das ocasiões que já existem, como as cerimônias do Scrum. Criando uma integração das atividades do time que geralmente possuem: designers, desenvolvedores e *testers*.

Como foi apresentado, o *Agile User-Centered Design* deixa o *design* com grande participação e importância desde o começo do projeto, permitindo que sejam definidos e estabilizados alguns requisitos antes da própria implementação, evitando vários possíveis retrabalho durante o desenvolvimento.

As sugestões coletadas através de testes de usabilidade e investigações de contexto podem realizar mudanças no *release* atual do produto. Considerando que muitas vezes essas sugestões causam grande impacto e valor no produto.

Testes de usabilidade auxiliam na iteração de como está o *design* do produto e entrevistas, grupos de foco e investigações de contexto ajudam a tratar dos requisitos. As práticas de *design* fornecem um resultado mais verídico do que a prática de fazer o *design* intencional.

Quando se trata de qualidade, se pode falar da satisfação do cliente com o projeto. Isso significa que o produto desenvolvido tem valor para o cliente. Ou seja, o objetivo inicial genérico de quase todos projetos é alcançado.

Em um alto nível de abstração se pode dizer que aliando essas práticas e conceitos, as chances de se fazer *softwares* com maior qualidade e que atendam de o cliente se torna maior. Podendo melhorar o trabalho que está sendo feito e o “como está sendo feito”, tornando o trabalho do usuário mais simples e com uma melhor experiência de uso, e com maior produtividade para equipe, capacidade de inovação no projeto e menos retrabalho.

7 CONSIDERAÇÕES FINAIS

7.1 CONCLUSÕES

Quando se trabalha com projetos de software, as equipes de experiência do usuário dentro das empresas e as equipes de desenvolvimento perdem muito se não trabalharem juntas na construção e implementação do projeto.

A comunicação e alinhamento dessas atividades são fundamentais para o sucesso do produto. *Frameworks* ágeis trouxeram um grande potencial de produtividade do time e quando essas práticas ágeis estão alinhadas com as atividades de *design*, a produtividade pode ser ainda maximizada.

A evidência é que para alinhar as atividades de cada área em um projeto que use as práticas ágeis é necessário realizar alguns ajustes ponderados na forma que a metodologia ágil é prescrita. Visando o trabalho simultâneo de atividades de design e desenvolvimento.

Como não existe uma fórmula para seguir é de extrema importância para garantir o retorno do produto que se trabalhe centrado no usuário, tornando-o participativo em várias etapas do desenvolvimento.

7.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO

Estudar uma maneira de mensurar a satisfação do cliente e qualidade agregada ao produto com a inclusão dessas melhorias nos processos de empresas que utilizam práticas ágeis e praticam metodologias de *design*.

Tratar dos temas de Lean e Agile UX realizando pesquisas e expondo os pontos importantes destes temas para inclusão no desenvolvimento de projetos é uma tendência que aparenta estar ganhando maior visibilidade no mercado.

Conceitos como Agile UX e Lean UX que seguem o mesmo pensamento deste trabalho, onde a idéia é aperfeiçoar ainda mais o rendimento dos times ágeis e

também trazer maior qualidade ao projeto com conceitos como *Design thinking*, *Lean thinking* e *User eXperience*, explicados no decorrer do trabalho.

REFERÊNCIAS

ABERGO - ASSOCIAÇÃO BRASILEIRA DE ERGONOMIA. Disponível em: <http://www.abergo.org.br/internas.php?pg=o_que_e_ergonomia>. Acesso em 13 nov. 2011.

ALLEMAN, Glen. Agile Project Management Methods for IT Projects. Niwot: Quorum Books, 2002.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 9241-11**: Requisitos Ergonômicos para Trabalho de Escritórios com Computadores Parte 11 - Orientações sobre Usabilidade. Rio de Janeiro. 2002.

BECK, Kent. FOWLER, Martin. MARTIN, Robert. **Planning Extreme Programming**. New York: Addison Wesley, 2000.

BECK, Kent. **Extreme Programming Explained**. Stoughton: Pearson Education, 2004.

BRANDÃO, Eduardo Rangel. **Publicidade on-line, ergonomia e usabilidade**: o efeito de seis tipos de banner no processo humano de visualização do formato do anúncio na tela do computador e de lembrança da sua mensagem. 2006. 400 p. Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Artes e Design. 2006.

COCKBURN, Alistair. **Agile Software Development**. 3. ed. New York: Addison Wesley, 2000.

COUTO, Rita Maria de Souza; OLIVEIRA, Alfredo Jefferson. **Formas do design**: por uma metodologia interdisciplinar. Rio de Janeiro: PUC-Rio, 1999.

DENIS, Rafael Cardoso. **Uma introdução à história do design**. São Paulo: Edgard Blucher, 2000.

DUL, Jan; WEERDMEEESTER, Bernanrd. **Ergonomia prática**. 2 ed. São Paulo: Edgard Blucher, 2004.

FERREIRA, Jennifer. **Interaction Design and Agile Development: A Real-World Perspective**. Wellington: 2007.

GARRETT, James Jesse. **The Elements of User Experience: User the centered design for the web and beyond**. 2 ed. Berkeley: New riders, 2011.

HESKETT, John. **Desenho industrial**. 2 ed. Rio de Janeiro: José Olympio, 1998.

HIRATSUKA, Tei Peixoto. **Contribuições da Ergonomia e do Design na Concepção de Interfaces Multimídia**. 1996. 168 p. Dissertação (mestrado) - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1996.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 6385: Ergonomic principles in the design of work systems**. Brussels. 2004.

KNIBERG, Henrik. **Scrum e XP direto das Trincheiras: Como fazemos Scrum**. C4Media of InfoQ.com, 2007.

KNIBERG, Henrik. SKARIN, Mattias. **Kanban e Scrum: Obtendo o melhor de ambos**. C4Media of InfoQ.com, 2009.

LOBACH, Bernd. **Design Industrial: bases para a configuração dos produtos industriais**. São Paulo: Edgard Blucher, 2001.

MANIFESTO ÁGIL. Disponível em: <<http://manifestoagil.com.br/>>. Acesso em 2 nov. 2011.

MORAES, Anamaria de. **Design e Avaliação de Interface**. Rio de Janeiro: iUsEr, 2002.

MORAES, Anamaria de; ROSA, José Guilherme Santa. **Avaliação e projetos no design de interfaces**. Teresópolis: 2AB, 2010.

MILLER, L. **Case Study of Customer Input For a Successful Product**. Toronto: Alias, 2005.

NIELSEN, J. Usability 101: Introduction to usability. Useit.com, ago. 2003, Disponível em: <<http://www.useit.com/alertbox/20030825.html>>. Acesso em 10 de nov. 2011.

NORMAN, Donald A. **The Design of Everyday Things**. Northbrook: Basic Books, 2002.

OLIVEIRA, Alex Avellar de. **Apostila de Engenharia de Usabilidade**. Universidade Santa Úrsula, 2006.

PREECE, Jenny et al. **Human Computer Interaction**. New York: Addison-Wesley, 1994.

SCHULMANN, Denis. **O desenho industrial: ofício de arte e forma**. Campinas: Papyrus, 1994.

SCHWABER, Ken. SUTHERLAND, Jeff. **Scrum Guide 2011 - The Definitive Guide to Scrum: The Rules of the Game**. 2011.

SCHWABER, Ken. **Agile Project Management with Scrum**. Redmond: Microsoft Press, 2004.

SY, Desiree. **Adapting Usability Investigations for Agile User-Centered Design**. Journal of Usability Studiesn. 2 ed. Bloomingdale: Maio 2007.