

A ABORDAGEM ÁGIL E A EXPERIÊNCIA DO USUÁRIO

AGILE APPROACH AND USER EXPERIENCE

CHAMMAS, Adriana (1);

OLIVEIRA, Jhonnata (2);

QUARESMA, Manuela (3);

(1) Laboratório de Ergodesign e Usabilidade de Interfaces da PUC-Rio (LEUI),

Mestre em Design

e-mail: ttabri@yahoo.com.br

(2) Laboratório de Ergodesign e Usabilidade de Interfaces da PUC-Rio (LEUI),

Bolsista de Iniciação Científica FAPERJ

e-mail: oliveira.jhonnata@gmail.com

(3) Laboratório de Ergodesign e Usabilidade de Interfaces da PUC-Rio (LEUI),

Doutora em Design

e-mail: mquaresma@puc-rio.br

RESUMO

O mercado é ávido por produtos tecnológicos que incrementem a produtividade dos usuários e clama por inovações em escala. Para acompanhar esse ritmo, empresas desenvolvedoras de software fazem uso de metodologias que nem sempre priorizam o usuário ante à entrega e/ou lançamento de seus produtos. Nesse artigo foram comparados os processos de desenvolvimento de software Scrum, XP e Lean UX, todos oriundos da abordagem ágil e, partir dos princípios que orientam a UX, colocada a importância tanto do peso que a voz do usuário deve ter no processo projetual quanto o momento no qual ela deve ser inserida.

Palavras-chave: Metodologia de design, design centrado no usuário, Metodologias ágeis e Desenvolvimento iterativo

ABSTRACT

The market is eager for high-tech products that enhance user's productivity and calls for innovations in scale. To keep this pace, software development companies make use of methodologies that do not always prioritize the user before the delivery and/or launching of their products. This paper compares software development processes Scrum, XP and Lean UX, all from the agile approach. Based on the principles that guide UX, the importance of the user's voice in the design process is observed and shows that much weight is needed and the moment when it is inserted should be reviewed.

Keywords: Design methodology, User-centered design, Agile methodology and Iterative development

1. INTRODUÇÃO

Diante da notável importância da tecnologia na rotina dos usuários e de toda praticidade advinda dela, as empresas desenvolvedoras de *software* investem em produtos que incrementem a produtividade e pluralizam as possibilidades de seus usuários. Ávidos por tecnologia de ponta, os usuários valorizam o inédito e aceleram a obsolescência de produtos tecnológicos. Nota-se que o mercado clama por inovações em escala. Para acompanhar o ritmo imposto, as empresas desenvolvedoras precisam entregar cada vez mais rápido seus produtos. O desafio dos desenvolvedores é equilibrar as restrições orçamentárias e temporais (cronogramas restritos) às limitações do próprio projeto e fornecer produtos palatáveis ao público que se destina.

Até meados dos anos 2000, o processo de desenvolvimento de softwares mais utilizado na indústria tecnológica foi o *Waterfall* (Cascata). O método *Waterfall* fraciona o projeto em estágios seguidos passo-a-passo, inspirado no Gráfico de Gantt. A maior crítica a respeito dele é a falta de controle sobre o cronograma e o orçamento, já que é necessária a conclusão do próximo passo para se se dê início ao seguinte. Para (Beck, 2005) a *Waterfall* pode ter funcionado até então, mas talvez não seja a melhor escolha no cenário de desenvolvimento de *software* de hoje, já que os hábitos, atitudes e valores dos primeiros anos são muito diferentes. Diante da necessidade premente de entrega desses produtos, foi criado o manifesto Ágil e, a partir dele, uma série de variações de sua aplicação. De acordo com a abordagem Ágil, equipes pequenas se organizam para encontrar meios efetivos de trabalho e que comportem as projeções temporais e orçamentárias. Dentre as variações da Ágil, foram escolhidas pela sua popularidade no mercado o Scrum, o XP (*Extreme Programming*) e a *Lean UX*, brevemente descritas abaixo:

Jeff Sutherland e Ken Schwaber criaram um método de desenvolvimento de softwares para a indústria tecnológica que tinha por missão acelerar o processo sem que a qualidade do produto fosse comprometida. Os autores o batizaram de Scrum e o descreveram como mais rápido, mais efetivo e mais confiável do que de gerenciamento do passado, que eram preditivas e em formato *top-down*¹. “O resultado almejado pelo Scrum é melhorar dramaticamente a produtividade da equipe” (Sutherland, 2014).

A *Extreme Programming*, mais conhecida como XP, coloca que a produtividade e a confiança estão tão ligadas às relações entre a equipe de trabalho, quanto às atividades de codificação de cada profissional. Os autores resumem a XP como um estilo de desenvolvimento de *software* focado na excelência da aplicação de técnicas de programação, na transparência entre os desenvolvedores, na habilidade para flexibilizar as restrições do projeto e responder prontamente às mudanças necessárias no decorrer do desenvolvimento. Os criadores da XP, Beck e Andres (2004) acreditam que essa metodologia envolve mudanças de comportamento.

¹ Abordagem “*top-down*” é uma estratégia de apresentação da informação para processamento e ordenação do conhecimento, partindo de uma macro visão do sistema para a para o micro visão. A abordagem “*Bottom-up*” parte exatamente do sentido inverso, quando apresenta o detalhamento do sistema do micro para o macro.

A *Lean UX* (Gothelf, 2013) objetiva que o desperdício de tempo com a documentação seja mínimo para que a produtividade alcance seu máximo. Nessa metodologia são otimizados os recursos, processos, documentação e flexibilidade contratuais para que os desenvolvedores possam chegar à inovação contínua requerida pela demanda, sem deixar de ouvir a voz do usuário para garantir sua melhor experiência ao interagir com o produto.

Nesse artigo serão comparadas as metodologias acima com enfoque nos critérios a seguir: (a) Em que tipo de projeto a metodologia é mais indicada; (b) Quais os métodos e técnicas utilizados pela metodologia; (c) Qual a importância de cada membro da equipe desenvolvedora no processo; (d) Como se dá o compartilhamento das informações/tomada de decisões durante o processo; (e) Quais ferramentas e/ou documentação são utilizados na metodologia em questão e (f) Como se dá o envolvimento do usuário no processo.

2. METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

Para Sommerville (2007) *apud* Melcher (2012) as organizações desenvolvem abordagens adequadas ao desenvolvimento de seus *softwares* de acordo com seu modelo próprio e o grau de estruturação varia de acordo com a criticidade do sistema. Não existe um processo projetual ideal, pois em sistemas de negócios, onde requisitos mudam rapidamente, um processo flexível e ágil é provavelmente mais eficaz (Sommerville, 2007 *apud* Melcher, 2012). A autora complementa que o modelo de engenharia de *software* deve ser escolhido com base na natureza dos projetos e que diferentes tipos de sistema requerem diferentes processos de desenvolvimento. A seguir serão apresentadas as abordagens Cascata (*Waterfall*) e Ágil, bem como três de suas variações: o Scrum, o XP e a *Lean UX*.

2.1 Cascata (*Waterfall*)

Em 1970 o americano W. W. Royce publicou na conferência de engenharia IEEE *Westcom*, um artigo que descreve o que é hoje conhecido como *Waterfall* - ou modelo em cascata. O autor fez uso do método *Waterfall* como exemplo do que previa como um risco e um convite para falhas e, por ironia, W. W. Royce acabou conhecido como o criador dessa metodologia (Sims & Johnson, 2011).

A *Waterfall* uma metodologia de desenvolvimento de *software* sequencial, no qual cada fase deve ser completamente concluída antes da próxima. O desenvolvimento é como um fluir constante e sequencial, e suas fases são as seguintes: análise de requisitos, projeto, implementação, testes (validação), integração, e manutenção de *software*.

A linearidade se propõe a ser intuitiva e os defensores do método alegam que no "aperfeiçoamento" do projeto, antes de passar à fase de execução, erros podem se revelar mais cedo o que, em teoria, reduziria os custos ao longo da vida do projeto. Na prática a metodologia não tem se mostrado adequada para a criação de *softwares*, uma vez que *softwares* são sistemas complexos e dinâmicos e não objetos estáticos. Sistemas desse tipo, concebidos na ausência de quaisquer dados experimentais reais são famosos por gerar uma confusão de inesperadas consequências antes de falhar.

Sutherland (2014), ao defender o ideal Ágil, classifica a *Waterfall* como obsoleta com a seguinte citação: "Existem duas formas de fazer as coisas: a velha *Waterfall*, onde se desperdiçam centenas de milhões de dólares e algumas vezes não se entrega produto nenhum, ou a nova

forma, que permite que mais produtos sejam entregues em menos tempo, com maior qualidade e menor custo” (Sutherland, 2014). O autor complementa que “ainda que pareça bom demais para ser verdade, a prova está nos resultados: funciona”.

2.2 Manifesto Ágil

Atentos ao cenário econômico e as mudanças agressivas da era do *e-business*, pensadores independentes - e em alguns casos até competidores entre si - se reuniram em busca de melhorar o processo do trabalho e estabeleceram princípios comuns. Como resultado, esses especialistas em processos de desenvolvimento de *software* criaram a Aliança Ágil e partir dela, estabeleceram o “Manifesto Ágil” (Beck, 2000 apud Hartson & Pyla, 2012). Do Manifesto Ágil emergiram os seguintes objetivos: satisfazer clientes com entregáveis rápidos e contínuos; entender que restrições de tempo e orçamento devem ser gerenciáveis e reconhecer que fazer alterações nos requisitos de desenvolvimento de softwares é norma inexorável. Estabeleceram ainda que a metodologia Ágil valoriza:

- A pronta resposta às mudanças de acordo com o planejamento;
- A permissão e o incentivo às interações entre os indivíduos e ferramentas durante o processo;
- A compreensiva documentação de apoio e espera a contínua e necessária comunicação entre clientes e desenvolvedores;
- A flexibilidade dos clientes no tocante aos pontos estabelecidos no contrato.

Apesar das críticas que faz ao modelo *Waterfall* e ao contrário do que possa parecer, o movimento Ágil não se apresenta como uma “anti-metodologia”. Highsmith (2001), membro da Aliança Ágil, publicou no site oficial do manifesto alguns importantes esclarecimentos: que a modelagem existe, mas que é liberta da “poeira” dos diagramas corporativos; que a documentação é importante, mas somente a que é realmente importante; que existe planejamento nos princípios do movimento, mas que devem ser respeitados os limites impostos à ele pela própria turbulência do mercado.

O “Manifesto Ágil de Desenvolvimento de Software” direcionou representantes do XP e do Scrum entre outros simpatizantes, a se unificarem perante a premissa das variações Ágeis de desenvolvimento, onde a condução da documentação fosse alternativa e os processos mais flexíveis. A própria Aliança Ágil comenta que, tanto o interesse recrudescente sobre a abordagem, quanto as críticas que possam surgir são derivados da maleabilidade da metodologia estabelecida pelos princípios do Manifesto.

2.3 Metodologia Ágil

Com a missão de priorizar indivíduos e iterações ante a processos e ferramentas, a entrega do *software* ante uma documentação abrangente, a parceira com o cliente mais que contratos estabelecidos e a resposta às mudanças, mais que seguir um plano rígido, os vinte princípios que balizam o Manifesto Ágil ainda são a referência para o mercado. Em linhas gerais, a proposta dessa metodologia de desenvolvimento se organiza em uma série de iterações limitadas por espaços de tempo, que são os “ciclos” (também chamados de *sprints*). Os ciclos costumam se dividir em três categorias: iterações iniciais, iterações construtivas e *release*. A Ágil engloba vários processos diferentes, elencados abaixo:

- *Acceptance Test Driven Development (ATDD);*
- *Agile Modeling;*
- *Agile Unified Process (AUP);*
- *Continuous integration (CI);*
- *Crystal Clear;*
- *Crystal Methods;*
- *Dynamic Systems Development Method (DSDM);*
- *Extreme Programming (XP);*
- *Feature Driven Development (FDD);*
- *Graphical System Design (GSD);*
- Kanban;
- *Lean software development;*
- Scrum;
- Scrum-ban;
- *Story-driven modeling;*
- *Test-driven development (TDD);*
- *Behavior-driven development (BDD);*
- *Velocity tracking;*
- *Software Development Rhythms;*

Dentre esses processos, foram selecionados para esse estudo, o Scrum, o XP e o *Lean UX* por serem os mais populares no mercado de desenvolvimento de softwares. A seguir serão apresentadas as características básicas de cada um dos processos escolhidos.

2.4 Características das Metodologias Scrum, Xp e *Lean Ux*

2.4.1 SCRUM

Graças à sua popularidade, o Scrum, é hoje uma estrela da abordagem Ágil. Esse processo é delineado da seguinte forma: São divididos dois pólos, sendo o primeiro composto pelos membros da equipe e o segundo pelo *backlog* do produto (Brown, 2013). Os membros da equipe são o *Scrum Master*, o *Product Owner* e a equipe de desenvolvimento. O *Scrum Master* é o grande responsável pelo andamento do projeto, e deve remover qualquer obstáculo que se anteponha à esses objetivos. O *Product Owner* (PO) deve definir as prioridades do trabalho, definir com a equipe as histórias dos usuários e representar a voz do usuário. A equipe de desenvolvimento por sua vez, deve fornecer *feedback* consensual sobre o escopo do trabalho como parte integrante desse processo. Alterações no projeto podem ser feitas, desde que por consenso. O *backlog* do produto é o repositório das histórias dos usuários, de quais tarefas serão necessárias para satisfazê-las e dos demais requisitos do projeto.

Conforme Chammas et al. (2013), no Scrum são feitas reuniões diárias de 15m (*daily standup*) para compartilhar o que foi feito no dia anterior e decidir o que deve ser feito no mesmo dia. Essa comunicação é vital para o processo, e a ideia é facilitar ao máximo a comunicação efetiva entre os membros da equipe em nome do andamento produtivo do projeto. No começo dos ciclos são feitos os *sprints* mais longos do Scrum, com duração máxima de 8h, para planejamento dos *sprints* vindouros e dos entregáveis específicos de cada *sprint* subsequente.

Embora Brown (2013) sugira que a melhor opção para incluir a voz do usuário no processo seria testá-lo desde o início e semanalmente, a autora coloca que dedicar um *sprint* ou dois para sessões de usabilidade pode ser uma opção factível. Em termos gerais, o profissional de UX da equipe (quando existe) tem pouco tempo para trabalhar no material antes que ele seja codificado. As maiores chances de sucesso estão baseadas na cultura, na dinâmica da equipe e do projeto e no quão essa equipe está preparada para os ciclos iterativos que envolvem essa abordagem (Chammas et al., 2013).

Os autores Hartson e Pyla (2012) apontam que os métodos ágeis de desenvolvimento de softwares normalmente são caracterizados por optarem pela informalidade, para que o *feedback* seja contínuo. Entretanto os autores colocam que, durante esta etapa, a literatura sobre a engenharia de *software* não menciona praticamente nada sobre design e que a usabilidade não é definida e nem sequer considerada. E ainda, os representantes de usuários não são usuários e não podem falar sobre a prática de outros. Nota-se que a participação do usuário é preterida em nome da agilidade do processo.

2.4.2 EXTREME PROGRAMMING (XP)

Extreme Programming ou XP é uma metodologia ágil de desenvolvimento de *softwares*, centrada no código (Beck, 2005). A XP é direcionada a equipes e/ou projetos de pequeno a médio porte, face às rápidas alterações nos requisitos do projeto. Para isso, adota uma estratégia de constante acompanhamento, direcionamento e ajustes durante o processo de desenvolvimento do *software*.

A metodologia XP é embasada em quatro valores principais: comunicação, simplicidade, *feedback* e coragem. E é a partir dos seguintes valores que se desdobram os seus princípios básicos: (a) *Feedback* rápido, onde se aplicam e interpretam o que é aprendido no sistema e o mais rápido possível; (b) Mudanças incrementais, quando problemas podem ser resolvidos com uma série de pequenas mudanças; (c) Simplicidade, quando a opção para resolver os problemas é sempre a mais simples possível; (d) Mudanças, onde deve-se manter um grande número de opções disponíveis para resolver o problema; (e) Qualidade, pois deve-se ter a qualidade como meta para todos os trabalhos desenvolvidos sob a XP.

Para Beck (2005), a XP foca explicitamente no escopo e recomenda-se a priorização de funcionalidades que representem maior valor possível para o negócio. O controle da qualidade é uma variável importante do projeto, pois não se admite que a qualidade seja preterida em função de ganhos a curto prazo. Em contrapartida, funcionalidades menos relevantes podem ser adiadas ou canceladas.

A XP sugere que uma variedade de pessoas entrelace seus *backgrounds* e trabalhe coesa, em busca de fazer com que o projeto seja o mais efetivo possível. A metodologia eleva o *Flow* a uma importância particular no processo, especialmente ao enredar diferentes tipos de trabalho

no desenvolver de um *software*, embora reconheça que o estado de “*Flow*”, apesar de sempre desejável, não é fácil de ser alcançado.

Os *stakeholders*² na XP são bem delimitados por Beck (2005), conforme pode-se conferir: (a) Testadores; (b) Designers de Interação; (c) Arquitetos; (d) Gerentes de Projeto; (e) Gerentes de Produto; (f) Executivos; (h) Escritores Técnicos; (i) Usuários; (j) Programadores; (k) Recursos Humanos. Apesar de bem definidos, os papéis de cada um não são nem fixos, nem rígidos. O objetivo é fazer com que cada um dos *stakeholders* contribua o máximo possível para o trabalho da equipe, assumindo os papéis que couberem suas contribuições. Como seria de se esperar numa equipe madura, todos podem fazer recomendações, embora devam estar preparados para as consequências de suas ações (Beck, 2005).

Para o autor, a XP envolve uma mudança de hábitos do passado em nome de novas formas de produzir melhor. Envolve que os *stakeholders* estejam abertos à perceber - e usar - as suas capacidades e, além de possibilitar, esperar que os demais façam o mesmo. O autor atrela a produtividade às relações humanas no ambiente corporativo e ressalta que tanto técnica quanto boas relações são cruciais para o sucesso. Beck (2005) define a XP como um estilo de desenvolvimento de *software* focado na excelente aplicação de técnicas de programação, clara comunicação e equipe que permita produzir o que não tinha sido previamente sequer imaginado.

2.4.3 LEAN UX

A *Lean UX*, desenvolvida por Gothelf (2013), é resultado da junção da *Lean Startup* (Ries apud Gothelf, 2013) do *Design Thinking* (Brown apud Gothelf, 2013) e da metodologia Ágil e prima pela certificação de que o produto entregue será realmente o desejado / esperado pelo usuário. Essa certificação se dá através de pesquisas, iteração do processo e métricas de qualidade. Conforme Chammas et al. (2013), a origem do termo *Lean* envolve uma filosofia de gestão que objetiva o máximo da produtividade através do mínimo de desperdício de tempo, recursos, processos, documentação e flexibilidade relativa entre clientes e equipe de desenvolvedores, para responder às mudanças e inovação contínuas, inerentes ao mercado e à própria tecnologia.

O processo só dedica o mínimo de informação na documentação “entregável”, só o necessário para assegurar o início - ou a implementação do projeto. Essa prática encoraja rascunhos e protótipos de baixa fidelidade, enriquecidos por comentários e insights da equipe envolvida, que deve ser coesa e colaborativa para que os múltiplos talentos possam se complementar. A filosofia *Lean UX*, focada na fase de design do projeto, encoraja o crescimento mútuo através da troca de conhecimento em busca da melhor experiência do usuário ao interagir com o produto.

² De acordo com a definição da ISO 9241210 (2010), *Stakeholders* são os Indivíduos ou a organização que tenha o direito, ação, reivindicação ou interesse em um sistema ou na sua posse das características que atendam às suas necessidades e expectativas, donde se conclui que são todos os diretamente envolvidos no projeto.

A *Lean UX* incentiva a prática de *benchmarks* e a criação de personas que representem os usuários do produto, embora nesse caso, a ordem dos passos para executar essa técnica tenha sido adaptada. Gothelf (2013) considera que perder muitas horas de entrevista (pesquisa de campo) antes de fazer pesquisas para validar as "crenças" da equipe seja desperdício e prega que as personas sejam desenvolvidas pela equipe reunida, onde prováveis usuários ilustram quem é o usuário real do produto e quais razões o levam a utilizá-lo. Chammas et al. (2013) colocam que a metodologia não aponta o momento preciso da participação do usuário na etapa projetual, embora tenha por premissa que ele deva ser onipresente nela.

3. METODOLOGIA DA PESQUISA

Os dados dessa pesquisa foram extraídos de uma extensa revisão bibliográfica das metodologias Scrum, XP e *LEAN UX*. Os critérios de escolha foram os seguintes: Em primeiro lugar, a data de lançamento da primeira publicação sobre o assunto e, em segundo, a relevância da obra, medida pelo volume de vendas. Após a identificação da bibliografia principal, foi realizado então um fichamento das mesmas. A partir dos fichamentos, foram discutidas as principais características de cada uma das metodologias e, enfim, selecionados os pontos possíveis de serem apontados em cada uma delas:

1. Que métodos e técnicas são utilizados no processo?
2. Qual é a "aplicabilidade" da metodologia?
3. Qual é a equipe envolvida?
4. Como se dá o relacionamento desta equipe durante o processo e quais documentações e/ou ferramentas são utilizadas?
5. Os usuários são inseridos no processo? Como?
6. Se são, como são obtidos os dados a respeito das necessidades e expectativas dos usuários?

Esses critérios possibilitaram a criação um quadro comparativo entre as metodologias, conforme pode-se conferir abaixo (tabela 1).

	Scrum	Extreme Programming	Lean UX
Que métodos e técnicas são utilizados no processo?	Reuniões - <i>Daily Scrum</i> ; <i>Story Time</i> ; <i>Sprint Review</i> ; <i>Retrospective</i> . Técnicas - <i>Product Backlog</i> ; <i>Task Board</i> ; <i>Burn Up Chart</i> ; <i>User Stories</i> ; <i>Planning Poker</i> e iterações (<i>sprints</i>).	Programação em dupla, testes automáticos de código, teste com o cliente, estimativa de tarefas e iterações curtas.	<i>Business Assumptions Worksheet</i> ; <i>Hypothesis Statements</i> ; <i>proto-Personas</i> ; <i>Feature Brainstorming Process</i> ; <i>Style Guides</i> ; <i>prototyping</i> ; <i>Prototype Test</i> ; <i>Research and survey</i> e <i>Lean UX cycle</i> (iteração).
Qual é a aplicabilidade" da metodologia, para que ela é indicada?	Pode ser utilizada para uma grande variedade de projetos, embora tenha sido criada para a produção de <i>softwares</i> . Pode ser utilizada tanto em projetos de qualquer porte (tamanho).	Metodologia criada para o desenvolvimento de softwares, que deve ser utilizada em projetos de pequeno a médio porte. As equipes podem ser pequenas ou médias.	Metodologia indicada para a criação de softwares, sem restrições quanto a complexidade do projeto.

Qual é a equipe envolvida?	Equipes pequenas. Não há especificações sobre funções referentes às habilidades ou talentos de cada um. A equipe SCRUM possui uma divisão "hierárquica" dividida em: The Product Owner, The scrum master e The Team Member.	Equipes pequenas ou médias, formadas por: engenheiros de software, programadores, arquitetos da informação, clientes, gerentes de produto, gerentes de projeto, testadores e consultores.	Equipes pequenas formadas por: engenheiro de software, gestor de produtos, designer de interação, designer visual, gerente de marketing e desenvolvedores.
Como se dá o relacionamento desta equipe durante o processo, qual documentações e/ou ferramentas são utilizadas?	Toda a equipe participa ativamente dos principais processos. Só se documenta o que for relevante e útil. Os registros são feitos por anotações das inúmeras reuniões, histórias do usuário, Gráficos feitos à mão, agendas, etc.	O foco principal é a criação de códigos e o testes. A comunicação é um dos valores mais importantes. Os programadores são orientados a trabalharem em duplas. As documentações utilizadas são: defeitos reportados, caso de testes, histórias do clientes e estimativas.	A equipe deve estar envolvida com o projeto do primeiro dia até sua conclusão. A documentação extensa e inócua deve ser evitada.
Os usuários são inseridos no processo? Como?	Através de testes realizados com protótipos de baixa fidelidade.	O usuário orienta as iterações do projeto através de "histórias", suas e de suas comunidades, que utilizadas pela equipe.	Através de testes realizados com os MVP, produtos viáveis mínimos, com o objetivo de se validar uma hipótese.
Se são, como são obtidos os dados a respeito das necessidades e expectativas dos usuários?	Através de personas de usuários, com base nos comportamentos das pessoas reais. Os usuários são divididos em duas categorias de personas: Primary Persona e Negative Persona.	Os usuários devem participar de comunidades que usem sistemas similares ao que estiver sendo desenvolvido, dessa forma trazem aos demais stakeholders mais histórias dessa comunidade.	As proto-personas são criadas através de um brainstorm junto com a equipe e, posteriormente, validadas com pesquisa de campo, a exemplo do Google analytics ou ferramentas de pesquisa similares.

Tabela 1 – Quadro comparativo entre metodologias.

4. COMPARAÇÃO DAS METODOLOGIAS

O quadro comparativo entre as metodologias possibilitou as reflexões enumeradas abaixo:

1. De acordo com o levantamento teórico, cada metodologia apresentada possui seus próprios métodos e técnicas, porém é possível observar padrões em comum, embora com nomenclaturas diferentes de uma metodologia para outra. Os métodos e técnicas comuns a todas são: (a) Reuniões periódicas com toda a equipe para direcionar ou redirecionar o projeto; (b) O andamento dos trabalhos é externalizado em locais visíveis para que todos possam compartilhar e/ou contribuir; (c) O cliente e/ou usuário dão o *feedback* sobre o trabalho; (d) A iteração é regra nas três metodologias.

2. Quanto à aplicabilidade da metodologia, as três objetivam a criação de softwares. A única que amplia a metodologia para o desenvolvimento de outros produtos é a SCRUM, embora o conceito de todas siga os princípios da abordagem Ágil, e possam ser replicados em projetos

diferentes dos de puro código. Quanto a dimensão do projeto, a Scrum e a *Lean UX* não se posicionam claramente, e deixam entrelinhas que a flexibilidade da metodologia molda proporcionalmente o tamanho/composição da equipe de acordo com o escopo do projeto.

A XP se propõe textualmente a projetos de pequeno a médio porte, uma vez que altera os requisitos do projeto com frequência. Todas elas, Scrum, XP e *Lean UX* priorizam e enaltecem o trabalho em equipe, assumindo que não se aplicam a trabalhos individuais.

3. Quanto a equipe envolvida, as três metodologias indicam que as equipes sejam pequenas, pois isso facilita a comunicação e os vínculos entre os membros da equipe. Na metodologia SCRUM, a hierarquia na equipe é claramente disposta em pequenos grupos, enquanto a *Lean UX* tem por primeiro princípio que as equipes sejam *cross-functional*³ e multidisciplinares. A XP se posiciona ao delimitar claramente os papéis dos *stakeholders* e elevar as relações humanas à capacidade de codificação e produção da equipe. Na XP os *stakeholders* podem - e devem - assumir outros papéis durante o processo para que seu talento possa ser aproveitado ao extremo, embora estatize que pode ter equipes de qualquer tamanho e que os valores e princípios da XP são replicáveis em qualquer escala. O Scrum multiplica a quantidade de pequenos núcleos, de forma que se mantenham reduzidos indiferente do escopo do projeto e coloca que, justamente por isso, é uma forma tão efetiva de trabalhar. A *Lean UX* é mais plástica e chega a utilizar somente dois profissionais em determinadas fases do projeto, enquanto reúne maior número em outras.

4. Na questão sobre o relacionamento entre os membros da equipe, as três metodologias apontam para uma quebra dos tradicionais modelos setorizados. Nos três casos todos os membros da equipe são convidados a experimentar atividades além de sua área de atuação original. A união e a “camaradagem” entre a equipe são fortemente incentivadas, assim como a criação em torno de um objetivo comum. Nos três casos se observou uma tácita rejeição a produção de documentações extensas dos processos. O foco, em todos os casos, é voltado para a aplicação dos métodos e entrega de valor, dentro do planejamento e orçamento iniciais, e sem detrimento das qualidades técnica e funcional.

5. No tocante à participação do usuário no processo, todas as metodologias afirmaram sua importância, embora sua participação se dê como validador de hipóteses, através de testes com protótipos de baixa fidelidade. Na XP essa validação ocorre através de conversas com o cliente, que nem sempre vai ser o usuário final do produto. O usuário, nesse caso, ajuda a equipe a escrever suas histórias e direcionar algumas das decisões de projeto, mas a XP assume que o usuário tenha conhecimento e experiência com sistemas similares, e que tenha fortes laços com a comunidade que fará uso do produto final.

6. Na fase inicial, tanto na *Lean UX* quanto na SCRUM, os dados a respeito das necessidades e expectativas dos usuários são suposições de especialistas, que possibilitam a construção de personas. As personas são baseadas em atitudes e comportamentos de pessoas reais, possíveis e prováveis usuários do produto em questão. Na *Lean UX* a validação da persona é feita através de um estudo de campo. A XP supõe que os usuários, além de serem membros de

³ Equipes *cross-functional* são compostas de pessoas de talentos e experiências diferentes e diferenciadas, que se unem em torno de um objetivo comum.

comunidades que usem sistemas similares ao que estiver sendo desenvolvido, trarão aos demais *stakeholders* outras histórias dessa comunidade, além da experiência e suposições pessoais. Em todas elas, Scrum, XP e *Lean UX*, o usuário será ouvido em fases mais adiantadas do projeto.

5 CONCLUSÃO

O objetivo desse artigo foi apresentar uma comparação entre as metodologias mais populares no mercado de desenvolvimento de software (todas oriundas da abordagem Ágil) a partir de uma revisão bibliográfica e apresentar os resultados dessa comparação. A ISO 9241-210/2010 define a experiência do usuário (UX) como as percepções e respostas de um usuário resultantes do uso ou do uso antecipado de um produto, sistema ou serviço e lista três fatores que influenciam a UX: o sistema, o usuário e o contexto de uso. Para a definição dos requisitos é fortemente recomendada a iteração da análise contextual dos usuários reais do produto e não há como fazer isso sem ouvir a voz do usuário (Chammas et al., 2013). Diferente do discurso que fundamenta a UX, a abordagem Ágil e as suas variações aqui expostas, substituem o usuário real do produto por membros da equipe.

Percebe-se que existem semelhanças entre as metodologias estudadas, uma vez que obedecem os princípios propostos pelo Manifesto Ágil. Percebe-se também que existem pontos difusos, referentes à aplicabilidade da metodologia e que isso se reflete nos métodos de cada uma. A iteração tem importante papel em todos os processos e se mostra uma arma eficaz em respostas às mudanças urgentes que possam surgir durante a atividade. Equipes pequenas e focadas são utilizadas nos três casos, além da exposição e compartilhamento do conhecimento e do andamento dos projetos, além da valorização dos talentos individuais como soma aos dos demais membros da equipe. A documentação abrangente dá lugar a constante entrega dos produtos - e valores - ao cliente, que também tem importante papel como parceiro de projeto.

A metodologia XP é centrada no código, enquanto no outro extremo a *Lean UX* se rotula uma metodologia centrada no usuário - de fato os testes com o usuário são parte importante nessa metodologia. Como meio termo, a SCRUM é focada no cumprimento dos *sprints* e na entrega de produtos. De certa forma o usuário é inserido no processo, embora com menor peso do que na *Lean UX*. Em todas, o usuário não é ouvido antes do projeto ter sido iniciado, o que reduz as possibilidades de ganhos maiores, quando importantes decisões ainda não foram sacramentadas.

O que se pode concluir com o estudo é que processos e ferramentas são, em tese, priorizados em relação aos usuários, tal qual a cartilha criada pela Aliança Ágil permite entender. A metodologia Ágil (Hartson e Pyla, 2012) está em plena sintonia com o mercado, pois tem a entrega por prioridade e personifica o usuário através de representantes deles (Chammas et al., 2013), mas a previsão e entrega do produto são prioritárias ante à experiência do usuário. Por exemplo, a *Lean UX* (Gothelf, 2013), mais próxima da UX dentre as estudadas, se preocupa com a inserção do usuário em busca de minimizar o risco de fracasso, mas se adequa às limitações de tempo e recursos impostas pelos responsáveis, e elimina tudo que possa se antepor ao andamento do projeto - inclusive a participação do usuário (Chammas et al., 2013).

São indiscutíveis os valores da abordagem Ágil, mas as ponderações a respeito do mais importante *stakeholder* do processo, o usuário, não podem deixar de ser levadas em conta. Os pontos que devem ser repensados são o peso do seu papel e o momento em que a participação dele deve ocorrer. Esse artigo sugere, baseado na proposta do Design Centrado

no Usuário prevista pela ISO 9241-210 (2010), que a voz do usuário não deva igual à dos demais conforme referências teóricas, em vez disso, mais importante. Esse trabalho também sugere que a participação do usuário seja promovida desde antes da definição do escopo do projeto, quando ainda há tempo de definir o conceito do projeto, e se delongue por toda a extensão do processo. Em médio e longo prazos, a quantidade de iterações no processo promete ser reduzida e os ganhos se refletirão na experiência que o usuário terá ao interagir com o produto em questão.

Este estudo faz parte de uma pesquisa maior em desenvolvimento sobre a adequação de diversos processos de design e desenvolvimento de softwares utilizados no mercado à abordagem do Design Centrado no Usuário (DCU). Para os desdobramentos da pesquisa, pretende-se analisar exemplos de aplicações dessas metodologias e processos, afim de enumerar os requisitos básicos e viáveis para uma abordagem adequada a melhor experiência do usuário com o produto final.

6. REFERÊNCIAS BIBLIOGRÁFICAS

AGILE ALLIANCE. **Manifesto for agile**. software development. 2001. Disponível em <<http://Agilemanifesto.org/>>. Acesso em: janeiro, 2015.

BECK,K; ANDRES,C. **Programação Extrema (XP) Explicada** - Escolha as Mudanças. São Paulo: ARTMED EDITORA S.A. 2004.

BROWN, Diana; **Agile User Experience Design: A Practitioner's Guide to Making it Work**. Waltham: Morgan Kaufmann, 2013.

CHAMMAS, A. S. ; QUARESMA, M. M. R. ; MONTALVAO, C. R. . **Um enfoque ergonômico sobre a metodologia de design de interfaces digitais para dispositivos móveis**. Arcos Design (Online), v. 7, p. 90-102, 2013.

GOTHELF,J; SEIDEN,J. **Lean UX Applying Lean Principles to Improve User Experience**. California: O'Reilly Media, Inc., 2013.

HARTSON, R.; PYLA, P. **The UX Book: Process and Guidelines for Ensuring a Quality User Experience**. Morgan Kaufmann, 2012

INTERNATIONAL STANDARD ORGANIZATION. Ergonomics of humansystem interaction Part 210: Human-centred design for interactive systems (9241-210). Génève: ISO, 2010.

MELCHER, Christiane. Dissertação mestrado. **Proposta metodológica para avaliações otimizadas de usabilidade em websites desenvolvidos com um método ágil**: Um estudo de caso. Rio de Janeiro: PUC, 2012.

SUTHERLAND, J. **The Art of Doing Twice the Work in Half the Time**. New York: Crown Business, 2014

SIMS,C; JOHNSON,H,L. **The Elements of Scrum**. California: DYMAXICON, 2011.