

ANA FLÁVIA YANAZE MURANOBU

UTILIZAÇÃO DE SCRUM E EXPERIÊNCIA DE USUÁRIO NO
DESENVOLVIMENTO DE SOFTWARE EM UMA APLICAÇÃO
PARA HEMOBANCO

Proposta para o desenvolvimento do Trabalho de Conclusão do Curso de Bacharelado em Sistemas de Informação, apresentado à UTFPR como requisito parcial para obtenção do título de bacharel em Sistemas de Informação.

Orientador: Prof^a Dra. Maria Cláudia Figueiredo Pereira Emer.

CURITIBA PR
2019

Resumo

Nas últimas décadas, muito tem-se ouvido falar das metodologias ágeis na indústria de software. Nesse contexto, o Scrum é a metodologia ágil que vem ganhando destaque por ser facilmente aplicável, ter maturidade no cenário tecnológico e ser bastante popular entre desenvolvedores. Ao mesmo tempo, uma boa Experiência de Usuário (UX) acabou classificando-se como um indicador de qualidade para aplicações. Um dos grandes desafios nessa indústria é integrar o UX às práticas de Scrum, estabelecendo uma correlação oficial entre ambos, nem consta uma boa prática de como correlaciona-las. Este estudo tem como objetivo apresentar maneiras de trabalhar com ambas em conjunto, eleger uma específica e aplicar efetivamente a abordagem escolhida no desenvolvimento de um software do Hemobanco. Por fim, considerações sobre a proposta serão feitas.

Palavras-chave: Scrum, UX, User Experience, Experiência de Usuário, Agile, Hemobanco.

Abstract

In the last decades, much has been heard about Agile on software development industry. In this context, Scrum is an agile method that has been gaining visibility because of its easy application, maturity on the tech scenario and its popularity among developers. At the same time, a good User eXperience (UX) has been classified as a quality standard for applications. The big challenge in this industry is how to integrate UX into Scrum practice, because there's no official correlation between them. This study aims to present approaches to work with both Scrum and UX, choose a specific one and apply it effectively into a Hemo Donor Center software development, then show the final considerations, suggestions and improvements for possible further projects.

Keywords: Scrum, UX, User Experience, Agile, Blood Bank, Blood Donor Center.

Sumário

1	Introdução	9
1.1	Objetivo	10
1.2	Objetivos Específicos	11
2	Levantamento Bibliográfico e Estado da Arte	12
2.1	Metodologias Ágeis em Desenvolvimento de Software	12
2.2	Scrum	14
2.2.1	Papéis / Time	15
2.2.2	Eventos	16
2.2.3	Artefatos	18
2.3	Experiência de Usuário - <i>UX</i>	19
2.3.1	Origem e Conceito	19
2.3.2	Experiência de Usuário em Softwares	21
2.4	Experiência de Usuário e Scrum	22
2.4.1	Trabalho Paralelo	23
2.4.2	Trabalho na mesma Sprint	23
2.4.3	Lean UX	24
3	Metodologia	25
3.1	Pesquisa e Revisão Bibliográfica	25
3.2	Elegendo uma Metodologia	25
3.3	Aplicação da abordagem escolhida	26
3.4	Fundamentos	26
3.5	Tecnologias	26
3.6	Cronograma	27

4	Recursos de Hardware e Software	28
4.1	Recursos de Hardware	28
4.2	Recursos de Software	28
5	Considerações Finais	30
	Referências Bibliográficas	31

Lista de Figuras

2.1	12 princípios Ágeis.	14
2.2	"The Scrum Framework Poster"[33]	15
2.3	"UX Design, um "guarda-chuva"entre Usabilidade e Design de Interação. [29], [16]	20
2.4	Fluxo de trabalho na mesma Sprint	23
2.5	Fluxo de trabalho Lean UX	24
3.1	Cronograma do projeto	27

Lista de Tabelas

2.1	Classificação de integração entre Experiência de Usuário e Scrum	22
4.1	Recursos de Software escolhidos para implementação do projeto.	29

Lista de Acrônimos

DAINF	Departamento Acadêmico de Informática
UTFPR	Universidade Tecnológica Federal do Paraná
UX	User Experience
Agile	Métodos Ágeis
XP	Extreme Programming
TI	Tecnologia da Informação
PO	Product Owner

Capítulo 1

Introdução

Atendendo à demanda global **exponencialmente** crescente de software, nas últimas décadas foram apresentadas diversas abordagens promissoras para auxiliar o processo e gestão em relação ao desenvolvimento de aplicações [2] que atendam às expectativas **do consumidor final e tenha foco no usuário** [11]. Naturalmente, com maiores procuras, surgiram também maiores desafios no âmbito tecnológico que levaram à uma grande troca [30] de metodologias convencionais, como o Cascata, pelas chamadas Metodologias Ágeis, que se destacam por - resumidamente - proporem adaptabilidade, entrega do produto com maior velocidade perante métodos mais convencionais e **versionamento mais frequente** [28]. Tal escopo e conceito ágil foi investigado profundamente por Boehm e Turner [8], que identificaram a base do *agile*, *i.e.* Métodos Ágeis. Os autores distinguiram quatro elementos: Aplicação, Gestão, Técnica e Pessoas, *i.e.* Recursos. Estes, segundo Boehm e Turner, concentram o que se necessita para o que faz tal conceito ser executado com êxito [7], [8], [9] .

Logo, o conceito de *agile* pode se manifestar em diversos métodos para o desenvolvimento de software, entre os quais o *SCRUM* é um dos mais maduros e amplamente utilizados [13], sendo também uma das poucas **metodologias** sobreviventes neste vasto mercado [2]. Este, para fins introdutórios, pode ser **dividido** em Eventos, Artefatos e Papéis/Time [31], [10]. A categoria Eventos inclui reuniões para conseguir resolver conflitos, realizar planejamentos, retrospectivas e acompanhamento; A categoria Artefatos contempla o **escopo e incremento** do produto; E a categoria Papéis/Time **retém** o dono do produto, o *Scrum Master* (cujo papel específico será mais discutido a frente neste texto) e o time de desenvolvimento.

Dessa forma, o Scrum mostra-se como uma forma de aplicação madura e concisa dos conceitos de Metodologias Ágeis para o desenvolvimento de software [13]. Contudo, mesmo abrangendo e guiando a implementação do produto final [31], métodos ágeis no âmbito computacional, em geral, tendem a ter maior foco no desdobramento do código e de funcionalidades em si (e.g. [6]) e não na experiência do usuário final [15].

Segundo [20], Usabilidade [1] e boa Experiência de Usuário são fatores chave para bons produtos e sistemas. Ambas focam no usuário final e em sua relação com o produto; No entanto, [24] afirma que as duas maiores escalas de medidas da Usabilidade são eficácia e eficiência [1], enquanto a Experiência do Usuário (*UX*) abrange muito mais que isso. Ela visa a satisfação do usuário em sua mais ampla versão: desde os sentimentos hedônicos de um produto antes mesmo de ele ser apresentado, até às expectativas levantadas durante a utilização do que quer que esteja sendo avaliado [18], que visa mais os sentimentos e emoções e vão muito além do foco de natureza orientado à tópicos e métricas da Usabilidade.

No âmbito de desenvolvimento computacional, enquanto o Scrum pode ser considerado um *framework* ágil com boa demanda e popularidade [19], a *UX* segue em ascensão neste mesmo cenário, sendo pauta de discussões [26] por existir um mercado cada vez mais competitivo no mundo tecnológico. Logo, cria-se a necessidade de ampliar cada vez mais o contexto de realidade de seus produtos e serviços. É assim que a *UX* entra para auxiliar o mercado de softwares, entendendo melhor novos escopos de produtos e/ou serviços [24].

Apesar de sua junção já ser aplicada muitas vezes no mercado [21] e ser, fundamentalmente, promissora, não há clareza ou padrões em como realizá-la [23], [15], [11]. Isso acaba gerando diversos questionamentos e incertezas quanto à integração dos dois elementos, e impede maior clareza e concisão nesse processo, como não conseguir envolver profissionais e usuários devido a alguns elementos de tempo estabelecidos no próprio Scrum [24].

1.1 Objetivo

Dado o cenário acima, torna-se objeto de pesquisa apontar *Qual a melhor maneira de conciliar Scrum com Experiência de Usuário (UX) para o escopo individual?*, ou seja, quando há apenas uma pessoa assumindo os papéis em ambos processos.

1.2 Objetivos Específicos

- Eleger uma maneira de conciliar Scrum e Experiência do Usuário para o desenvolvimento de uma aplicação
- Empregar Scrum durante o desenvolvimento como forma de gerenciamento do projeto, de forma a adequar a metodologia para desenvolvimento individual
- Empregar Experiência de Usuário no desenvolvimento para entender as necessidades de quem irá utilizar a aplicação
- Avaliar a forma de aplicação conjunta do Scrum e Experiência de Usuário utilizada neste estudo, no que diz a respeito a prazo de entrega do produto, qualidade e atendimento dos requisitos

Baseando-se na análise de casos e levantamento bibliográfico, afim de adquirir um rico estado da arte sobre o objeto de pesquisa, este estudo visa assim desenvolver uma proposta para responder à questão de pesquisa utilizando a classificação em [21]. Após esse levantamento, será estabelecido um cronograma e a forma de estruturação e andamento do projeto.

O tema, escopo, desenvolvimento e estudo do projeto será baseado em uma aplicação para Hemobancos.

Capítulo 2

Levantamento Bibliográfico e Estado da Arte

2.1 Metodologias Ágeis em Desenvolvimento de Software

Na década de 90, o que ainda era referência para muitos desenvolvedores eram métodos mais tradicionais, como o *Waterfall* e o Espiral. Conforme o mercado crescia e as necessidades se expandiam, alguns desenvolvedores perceberam que tais metodologias não eram mais ideais para seu contexto; Assim, eles acabaram criando novos princípios para o processo de desenvolvimento, misturado com antigos. Esses novos princípios acabaram trazendo mais próximos o desenvolvimento dos *stakeholders* e também satisfazendo mais ambos. Com isso, as pessoas que criaram esses princípios sentiram a necessidade de expandir suas ideias e seu *framework* para o mundo. Ainda assim, essa expansão foi muito restrita e não houve grande crescimento dessas ideias [4].

Um grupo de 17 pessoas, então, em 2001 em Utah, se reuniu para debater seus princípios e ideias, trazendo semelhanças e diferenças e o que de fato funcionava. Houve diversos pontos em que essas pessoas discordaram, porém nos pontos que houveram assentimentos, este grupo pôde criar o que hoje é conhecido como Manifesto for Agile Software Development, i.e. *Agile Manifesto* [4], [14] e, posteriormente, criaram a *Agile Alliance*.

O Manifesto Agile [14], então, tornou-se um modelo para boas práticas de software a serem seguidas, que tem como foco um conjunto de valores discutidos por seus idealizadores. São eles:

- Indivíduos e iterações acima de processos e ferramentas
- Software em funcionamento acima de documentação abrangente
- Colaboração com o cliente acima de negociação de contratos
- Responder à mudanças antes de seguir o plano

Além disso, o Manifesto contém 12 princípios que refletem os mencionados valores acima. São eles:

- Nossa maior prioridade é satisfazer o cliente através de entrega antecipada e contínua de software.
- Receber mudanças de requisitos, mesmo tardiamente no projeto.
- Entregar software estável com frequência, de em algumas semanas a alguns meses, com preferência à escala de tempo mais curta.
- A equipe de negócios e a de desenvolvimento devem trabalhar juntos diariamente durante o projeto
- Construa projetos em torno de indivíduos motivados.
- O método mais eficiente e eficaz de transmitir informações para e dentro de um desenvolvimento a equipe é conversa pessoal.
- A versão estável do software é a medida prioritária de progresso.
- Processos ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem poder manter um ritmo constante indefinidamente.
- Atenção contínua à excelência técnica e um bom design aumenta a agilidade.
- Simplicidade - a arte de maximizar a quantidade de trabalho não feito - é essencial.
- As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizadas.

- Em intervalos regulares, a equipe reflete sobre **como** para se tornar mais eficaz, então afina e ajusta seu comportamento de acordo.



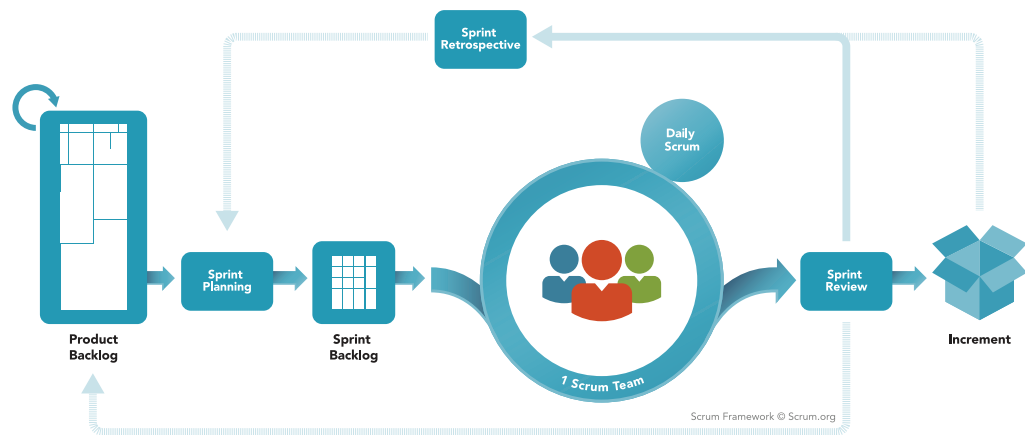
Figura 2.1: 12 princípios Ágeis.

Fonte: <https://claudemirgarcia.wordpress.com/2017/04/05/os-12-principios-ageis>

2.2 Scrum

Seguindo os valores e princípios **acima citados**, o Scrum é uma **metodologia** ágil criada por dois dos idealizadores do Manifesto, Ken Schwaber e Jeff Sutherland [5], [6]. **Tendo** seu nascimento anterior ao próprio Manifesto, o Scrum é hoje uma das metodologias ágeis mais maduras e mais aplicadas no mercado [13]. Mais de um terço (1/3) dos profissionais de T.I. em países europeus desenvolvidos, como a Islândia, usavam esse processo em seus projetos [25] já em 2009, e essa popularidade continua em ascensão no mercado tecnológico [12]. **Este estudo** irá se basear no Guia Oficial do Scrum, desenvolvido e mantido pelos idealizadores do Scrum, mencionados acima e referenciado em [31]. O **scrum**, segundo o Guia, é dividido em três **grupos**: Papéis/Time, Eventos e Artefatos.

SCRUM FRAMEWORK



Scrum.org

Figura 2.2: "The Scrum Framework Poster"[33]

2.2.1 Papéis / Time

O time consiste de um *Product Owner*, Scrum Master e o Time de Desenvolvimento. Aqui, o time é auto gerenciável e multi funcional. Feedback também é um item muito importante ao time de Scrum, cuja aquisição é conquistada com a entrega incremental e iterativa do produto pelo time.

Product Owner

Também chamado de Dono do Produto, é o responsável por gerenciar o backlog do produto de forma clara, ordenando seus itens e garantindo que esse seja transparente e visível para todos os interessados. Dessa forma, esse perfil guia o que necessita ser feito a seguir e maximiza o resultado e valor do produto. A execução desse papel varia de organização para organização, não sendo estabelecidas outras boas práticas para tal função no Guia Oficial do Scrum.

Scrum Master

Como função principal, o Scrum Master é responsável por colocar em prática de forma efetiva o Scrum como definido no Guia Scrum. Para quem não está dentro do time, ele

é reponsável por fazer com que haja interação dessas pessoas com o projeto da maneira mais eficiente possível.

Seu trabalho para com o Product Owner garante que todos os objetivos sejam entendidos por todos do projeto, além do escopo do produto e o seu domínio. Ele também deve garantir que o PO saiba gerenciar o backlog de forma clara e efetiva.

Já com o Time de Desenvolvimento, o Scrum Master fica responsável por auxiliar a destrinchar quaisquer impedimentos que surjam no decorrer do processo, garantir sua aplicação de forma eficaz, facilitar eventos Scrum, treinar o time para alcançar autogerenciamento e interdisciplinaridade e também facilitar a clara comunicação entre o PO e o Time de Desenvolvimento.

Time de Desenvolvimento

São profissionais que realizam o trabalho para entregar, a cada iteração, um incremento do produto considerado "Pronto" para aquele momento. No Scrum, esse time é responsável por organizar e gerenciar seu próprio trabalho, de forma auto organizada para transformar os itens de *Backlog* em Produto. O Scrum não reconhece ou estabelece subtítulos e subtimes dentro do time em seu guia. O Time, para a metodologia, é aquela que, como um todo, independente da subtitularidade de seus integrantes, deve transformar o *Backlog* do Produto em incremento.

Segundo o Guia, o tamanho ideal do Time de Desenvolvimento é *"pequeno o suficiente para se manter ágil e grande o suficiente para completar um trabalho significativo dentro da Sprint"*.

2.2.2 Eventos

São chamados Eventos em Scrum certos eventos *time-boxed*, com duração máxima e prescritas, realizadas com pré-determinada regularidade. Essas, foram criadas para diminuir a necessidade de reuniões não previstas. Todo evento contém duração máxima.

Sprint

A Sprint é o coração do Scrum; Ela é um *container time-boxed* de um mês ou menos, com duração fixa, para outros eventos. A partir dela, são realizadas planejamentos para o alcance de objetivos, reuniões diárias, estabelecidos os trabalhos feitos de forma flexível

pelo Time Scrum, a revisão da Sprint e retrospectiva. Cada Sprint tem uma meta e é considerada um pequeno projeto, onde no final o Time chega a um subproduto estável e entregável.

Seu **cancelamento** pode ser realizado única e exclusivamente pelo Product Owner. Quando isso ocorre, qualquer subproduto deve ser revisado e readequado ao gerenciamento do projeto.

Planejamento da Sprint

Como mencionada anteriormente, a partir da Sprint é necessária a elaboração de outro evento: o Planejamento da Sprint. Ela é nada mais do que um evento de no máximo oito horas, realizado antes do início da Sprint pelo Time Scrum, onde duas questões devem ser respondidas: *O que pode ser entregue como resultado do incremento da próxima Sprint?* e *Como o trabalho necessário para entregar o incremento será realizado?*

A partir do Backlog do Produto, é necessário o Time de Desenvolvimento prever as funcionalidades a serem desenvolvidas durante esse período. A partir da capacidade do Time, então, é estimado esforços, previstas funcionalidades e avaliado a viabilidade de um entregável. Neste evento, também, é decidido de forma clara e concisa como o Time de Desenvolvimento pretende trabalhar para completar o objetivo da Sprint e realizar o entregável previsto.

Reunião Diária

Também chamada de *Daily*, o evento acontece internamente entre o Time de Desenvolvimento e tem como propósito informar a todos dentro dele sobre o status e andamento do projeto, de forma a deixar transparente quaisquer dificuldades, pretensões e trabalho realizado. Sua estrutura é estabelecida pelo time e realizada, como sugere o nome, diariamente. A duração desta reunião deve ser curta (máximo de 15 minutos), de forma a não atrapalhar o desenvolvimento do produto, e sim somar.

Revisão da Sprint

Ao final da Sprint, é realizada a sua revisão. Esse evento inspeciona o incremento e adapta o Backlog do Produto caso necessário. Aqui, as partes interessadas e o Time Scrum trabalham para alinhar seus interesses e objetivos. Esta é uma reunião de no máximo

4 horas de duração, para uma Sprint de um mês; aqui é exposto quais itens do Backlog foram "Prontos" e quais não foram durante a Sprint, esclarecido dúvidas sobre o incremento, a revisão de problemas durante a Sprint e revisão da linha do tempo, orçamento e outros aspectos do projeto como um todo.

O resultado desse evento é um Backlog do Produto revisado, que servirá de horizonte para definição de itens a serem feitos na próxima Sprint.

Retrospectiva da Sprint

Após a Revisão da Sprint e antes do Planjamento da Sprint, em no máximo três horas, o Time Scrum inspeciona seu próprio desempenho para com o projeto, identifica e ordena os principais itens que foram bem e suas potenciais melhorias e, principalmente, cria um plano de ação para sempre melhorar seu desempenho.

2.2.3 Artefatos

Os artefatos são informações chaves sobre os elementos do Scrum, para que todos tenham entendimento esclarecido e transparente sobre alguns aspectos já mencionados neste artigo. São eles:

Backlog do Produto

De forma simples, é uma lista ordenada de tudo que julga-se necessário pelo Product Owner para a implementação e entrega produto final. Tal artefato pode ser mudado e reorganizado diversas vezes durante o projeto, de forma que o topo da lista sejam coisas prioritárias para o desenvolvimento do produto. O Time de Desenvolvimento e o Scrum Master podem ajudar a apontar aspectos importantes para elaboração do Backlog do Produto, mas apenas o Product Owner julga o que realmente é necessário para tal.

Backlog da Sprint

A partir da definição de uma meta para a Sprint, é filtrado itens do Backlog do Produto pelo Time de Desenvolvimento e transformados em Backlog da Sprint. Assim como o artefato anterior, este é flexível e pode ser modificado no decorrer da Sprint, o evento time-boxed que este artefato está contido. Esse Backlog é rastreado inteiramente pela

equipe de desenvolvimento, e deve ser adequada às necessidades da equipe de acordo com a meta estabelecida para a Sprint.

Incremento

Atendendo de forma transparente ao Backlog da Sprint em execução, o Incremento é um subproduto, um entregável que atende ao objetivo da Sprint. No decorrer do projeto, representa cada vez mais um subproduto perto do objetivo final.

Definição de "Pronto"

Para qualquer item do Backlog do Produto, incremento ou qualquer outro artefato do Scrum, essa definição é atribuída a ele de forma concisa e de entendimento / significado idênticos a todos os integrantes do Time Scrum. Tal definição deve ser estabelecida no início do projeto, de forma a não impactar posteriormente sobre o produto.

2.3 Experiência de Usuário - UX

2.3.1 Origem e Conceito

A relação do produto com o usuário tem sido estudada por designers e desenvolvedores há anos. Vários conceitos, como a Usabilidade, contém diversas boas práticas [1] a fim de proporcionar efetividade, eficiência e satisfação em um contexto específico para quem utilizar o artefato. Contudo, a relação desses conceitos de forma a engloba-los não existia.

Durante seu período como vice presidente do Grupo de Tecnologia Avançada na Apple, Donald A. Norman, co-fundador do Nielsen Norman Group - renomeou o grupo para "*User Experience Architect Group*", por achar que definições como Usabilidade e Experiência de Usuário limitavam muito o escopo da função de seu setor dentro da empresa na qual trabalhava [3]. A partir disso, o termo UX - diminutivo de User eXperience - tomou popularidade internacionalmente.

O Design de Experiência de Usuário [16], visa modelar a experiência de usuário ideal ao utilizar um determinado serviço ou produto. Dessa forma, pode-se observar a amplitude e complexidade [24] do termo: ele retém diversos outros campos de estudo, desde acessibilidade a prototipação. Por isso, surge a importância de atribuir um significado comum a todos em determinado conceito.

Para o *Interaction Design Foundation*, que também baseia-se nos fundamentos explicados por Donald A. Norman, mencionado **acima** [3], [16], UX performa e engloba tudo o que concerne o usuário sobre os sentimentos e sensações que eles **tem** sobre determinado artefato.

“Humans have always been emotional and have always reacted to the artifacts in their world emotionally.” (Humanos sempre foram emocionais e sempre reagem aos artefatos em seu mundo de forma emocional)

- Alan Cooper, presidente da Cooper



Figura 2.3: "UX Design, um "guarda-chuva" entre Usabilidade e Design de Interação. [29], [16]

Especialistas na área também complementam a definição **acima**, apontando-no frequentemente como um "guarda-chuva" para os clássicos **Design de Interação** - parte do Design que foca a implementação a partir da observação e teste com o usuário [32] - e **Usabilidade** - medida **pela qual um artefato pode alcançar** ao ser utilizado, levando em conta sua eficiência, efetividade e satisfação [1] -, (Ver figura 2.3) **onde olha-se as limitações humanas no que diz respeito de entender, responder e lembrar**. E vai além do início da

experiência. Ela abrange o durante e até o depois do uso do artefato, em níveis físicos e mentais, em diferentes níveis, como psicológicos e sócio econômicos. [29]

2.3.2 Experiência de Usuário em Softwares

A partir de meados de 2005, a complexidade dos softwares vêm aumentando de forma considerável. Com isso, tornou-se real a maior interatividade com sites e o escopo do que eles podem oferecer ao internauta [16]. A demanda por essas funcionalidades expandiu-se de forma crescente; Contudo, o objetivo e o sucesso do software muitas vezes se resume, simplesmente, em uma sentença: como os usuários se sentem em relação ao artefato/software [20], [16].

Para levar em consideração a Experiência de Usuário no escopo de Software, frequentemente Designers levantam perguntas como:

- O Software dá valor ao usuário?
- O usuário considera a aplicação simples para navegar e utilizar?
- O usuário, ao usar o software, aprecia sua utilização?

Para considerar **boa** Experiência de Usuário de uma determinada aplicação, a resposta seria "Sim" aos três itens acima.

Portanto, para atingir o objetivo acima, é importante "desenhar" o software centrado no **usuário**; Desde o levantamento de requisitos até a entrega final do projeto, colocando e avaliando suas necessidades implícitas e explícitas para o projeto, de forma a guiar o desenvolvimento a fim de sempre **enalter** as sensações de quem irá utilizar a aplicação.

2.4 Experiência de Usuário e Scrum

Ao mesmo tempo em que o Scrum ganha visibilidade nas últimas décadas para um eficiente e eficaz desenvolvimento, a UX torna-se um sinalizador de qualidade no produto [24]. Um dos maiores desafios na indústria de Software é a integração de ambos os conceitos, pois não há maneira padrão de uni-los [21], [25], [24].

Em 2018, Sesaria Kikitamara e Anastasia Ary Noviyanti, da *Universitas Prasetiya Mulya*, realizaram um modelo conceitual do uso de Experiência do Usuário no Scrum [21], levantando uma rica revisão bibliográfica de estudos que utilizaram o método a partir de 2005.

Este trabalho terá como base o estudo realizado por Kikitamara e Noviyanti mencionado anteriormente, que classifica, basicamente, essa integração em três categorias:

Tabela 2.1: Classificação de integração entre Experiência de Usuário e Scrum

Abordagem	Descrição/Conceito
Trabalho Paralelo	O time de desenvolvimento e o time de UX trabalham após alinharem e planejarem suas atividades. O time de UX fica até 2 Sprints a frente do time de desenvolvimento, para coletar dados do usuário, prototipar telas e propor soluções embasadas no usuário, ao projeto.
Trabalho na mesma Sprint	O(s) responsável(is) pela UX trabalham simultaneamente na mesma sprint que o time de desenvolvimento durante todo o projeto.
Lean UX	Aqui, há maior envolvimento do usuário durante o desenvolvimento; O time de UX ficará focado no MVP (<i>Minimal Viable Product - Produto Mínimo Viável</i>), para deliberar algo menor, porém similar, ao produto final (sem funcionalidades propriamente ditas, porém, sempre claras suas pretensões futuras.), com objetivo de sempre receber feedback sobre a aplicação e alinhar as expectativas com o usuário.

A partir do levantamento, avaliação de qualidade e leitura de diversos estudos de caso e relatos dessa integração, as autoras puderam classifica-las em uma das três abordagens citadas acima, descritas em seu trabalho.

2.4.1 Trabalho Paralelo

Durante o desenvolvimento, quando há novas funcionalidades, pode ocorrer da prototipação de interface **podem não serem terminados** a tempo [17]. Em casos como esse, o trabalho paralelo resolveria o problema.

Nesta abordagem, o time de desenvolvimento e o time de UX trabalham de forma paralela após planejarem e alinharem suas atividades - geralmente na Sprint 0 [22], **onde** não ocorre desenvolvimento de fato - e estabelece um padrão de forma de trabalho, a fim de sincroniza-los. Nesse planejamento, os times podem realizar em conjunto atividades como análise de tarefas, entrevistas, questionários e protótipos no papel.

Geralmente, o time de UX se mantém de uma a duas Sprints de vantagem para ter tempo o suficiente para coletar dados de usuários, analisar dados coletados e outros dados e propor soluções para o projeto, centrados no usuário. Para a atual sprint, o time de UX pode auxiliar desenvolvedores na implementação e criação do design do projeto.

2.4.2 Trabalho na mesma Sprint

Nesta abordagem, enquanto no início do projeto o time de desenvolvimento realiza o levantamento do Backlog para a primeira Sprint, o time de UX realiza o design para - igualmente - a primeira Sprint.

Ao mesmo tempo em que coletam dados para a próxima Sprint, o time UX inspeciona o trabalho realizado na Sprint passada e faz pesquisa para a Sprint $i+2$. Dessa forma, o trabalho em conjunto com o PO torna-se mais necessário dependendo da Sprint, a fim de inspecionar continuamente o produto.

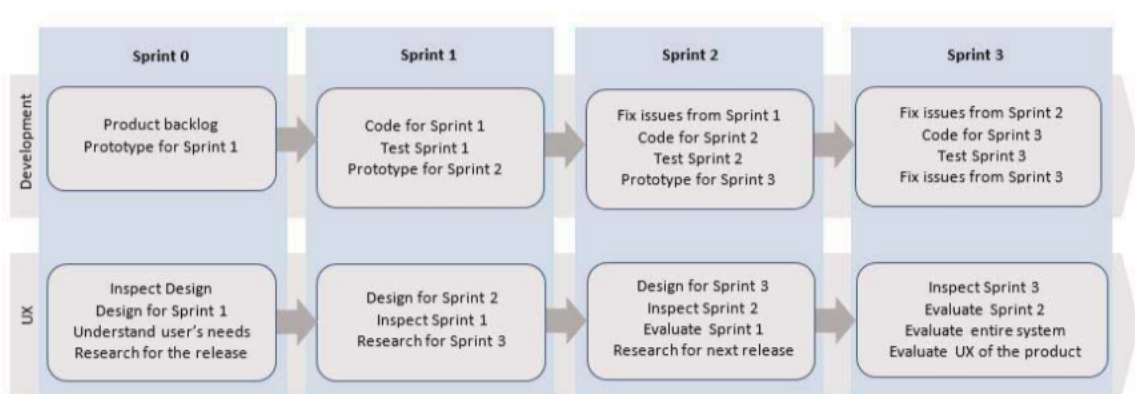


Figura 2.4: Fluxo de trabalho na mesma Sprint

2.4.3 Lean UX

A fim de minimizar o risco de desperdiçar recursos implementando um design que não atende às necessidades do usuário, esta abordagem visa focar na implementação de um MVP - *Minimal Viable Product*, Produto Mínimo Viável, um produto semelhante ao projeto final - porém menor - deixando claro quais serão as funcionalidades e focando no design para validar se o mesmo atende a contempla as necessidades do usuário.

Para isso ser realizado de maneira bem sucedida, o usuário deverá participar de forma interna. Ou seja, durante uma Sprint de no mínimo duas semanas, sua validação será necessária. (Ver figura 2.5)



Figura 2.5: Fluxo de trabalho Lean UX

Fonte: [21]

Nesta abordagem, também é importante trabalho conjunto em todo o Time Scrum. Cada membro do time irá focar em sua área, mas irá dar suporte para todos, se necessário. Aqui, há foco no *feedback* do usuário, e, a partir disso, ações são tomadas. Conceitos como *Design Thinking* e *Lean Production* também podem ser aplicados nesta metodologia de trabalho de maneira explícita, já que sua filosofia deriva de ambos, juntamente com a Metodologia Ágil [27].

Capítulo 3

Metodologia

Este capítulo traz a descrição dos processos a serem realizados neste trabalho. Também descreve ferramentas, tecnologias utilizadas e traz um cronograma para realização das atividades.

3.1 Pesquisa e Revisão Bibliográfica

Com base no estudo realizado por Kikitamara e Noviyanti [21] e de outros artigos que se mostrarem relevantes, este trabalho irá trazer uma revisão bibliográfica dos artigos estudados nele, a fim de levantar similaridades com o contexto em que se encontra. Após a releitura e mantendo a classificação composta pelas autoras, este estudo poderá seguir para o próximo passo.

3.2 Elegendo uma Metodologia

Após a Revisão Bibliográfica, serão levantadas similaridades dos artigos analisados com o cenário e contexto deste estudo para, enfim, eleger uma das três abordagens propostas pelas autoras (Ver tabela 4.1). A escolha de uma das abordagens será definida respondendo às seguintes perguntas e/ou atendendo aos seguintes critérios:

- 1) A abordagem é factível no contexto de um desenvolvedor?
- 2) A abordagem consegue atender ao cronograma proposto no capítulo?
- 3) A abordagem irá trazer novos conhecimentos e conceitos ao projeto?

- 4) A abordagem apresenta casos de sucesso?
- 5) A abordagem apresentou problemas durante o desenvolvimento? Caso sim, quantos e quais?
- 6) Classificar criticidade dos erros da pergunta anterior.

Respondendo à essas perguntas, com ênfase em (1), (5) e (6), pretende-se eleger de forma justa a melhor abordagem para integrar Experiência de Usuário com a metodologia ágil Scrum para o contexto deste trabalho.

3.3 Aplicação da abordagem escolhida

Após a escolha da abordagem em 3.2, será realizada desenvolvida uma aplicação, de forma a seguir tal abordagem escolhida e abrangendo os seguintes conceitos de Experiência de Usuário: aplicação de questionário, Mapeamento de Experiências e Jornada de Usuário, com pessoas da área de Design e Desenvolvimento. Já a codificação será feita individualmente, com elementos de back e front-end.

3.4 Fundamentos

Para realizar o Design de Experiência de Usuário, será utilizado elementos da UX, como aplicação de questionário em uma empresa de tecnologia, a fim de levantar dados de possíveis usuários da aplicação; **Jornada de Usuário**, para entender o que levaria o usuário a utilizar a ferramenta em desenvolvimento; **Mapeamento de Experiências**, para ajudar a levantar funcionalidades na hora da codificação. O primeiro elemento a ser aplicado, será o questionário. Os outros dependem da eleição de uma abordagem para serem inseridos de forma a respeitar a classificação de integração escolhida.

3.5 Tecnologias

Para o desenvolvimento do back-end, será utilizado o PHP por ser uma linguagem com documentação estável e madura, sendo referência para o desenvolvimento web atualmente. Da mesma forma, o framework Angular foi eleito para desenvolvimento front-end. A

arquitetura REST será utilizada para atender às necessidades de comunicação durante o desenvolvimento da aplicação, de forma segura.

3.6 Cronograma

O cronograma do projeto é apresentado abaixo (Ver figura 3.1).



Figura 3.1: Cronograma do projeto

Capítulo 4

Recursos de Hardware e Software

Este capítulo traz recursos de Hardware e Software que são considerados importantes e / ou essenciais para a conclusão deste projeto.

4.1 Recursos de Hardware

Requisitos mínimos para o desenvolvimento são:

Processador Intel Core i5 1,4 GHz, RAM 4 GB 1600 MHz DDR3 e Intel HD Graphics 5000 1536 MB.

4.2 Recursos de Software

Para o desenvolvimento do projeto, foram escolhidos os seguintes recursos de Software:

Tabela 4.1: Recursos de Software escolhidos para implementação do projeto.

Nome	Versão	Função
macOS	10.14.4	Sistema Operacional
Windows Server	2010	Sistema Operacional
PHP	7	Linguagem de Programação Back-end
Angular	7.1	Framework de Programação Front-end
Webstorm	2019.1.2	IDE Front-end
PHP Storm	2019.1	IDE Back-end
XAMPP	7.1.30	Servidor

Capítulo 5

Considerações Finais

O presente documento visa apresentar uma proposta para estudar, eleger e avaliar a aplicação de uma abordagem para integrar o framework de projetos Scrum com a Experiência de Usuário. Para isto, deve ocorrer a escolha de uma das três abordagens classificadas referenciadas em capítulos anteriores e em [21]. Após, aplicação do elemento da UX, e, em seguida, seguir a abordagem eleita no desenvolvimento do estudo, de forma a implementar uma aplicação seguindo princípios do Scrum e do Design de Experiência de Usuário. Por fim, será documentada Conclusões acerca do estudo e a abordagem escolhida, como também possíveis propostas para continuidade do trabalho.

Referências Bibliográficas

- [1] ISO/IEC 9241-210. *Human-Centered Design Processes for Interactive Systems*. ISO/IEC 9241-210, 2010.
- [2] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen, and Juhani Warsta. Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*, 2017.
- [3] Edu Agni. Don norman e o termo “ux”. <https://uxdesign.blog.br/don-norman-e-o-termo-ux-6dffb3f8d218>. Acessado em 19/06/2019.
- [4] Agile Alliance. Agile 101. <https://www.agilealliance.org/agile101/>. Acessado em 30/03/2019.
- [5] Kent Beck, Mike Beedle, Arie van Bennekun, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for agile software development. <http://agilemanifesto.org/>. Acessado em 30/03/2019.
- [6] Kent Beck and Erich Gamma. *Extreme programming explained: embrace change*. addison-wesley professional, 2000.
- [7] Barry Boehm and Richard Turner. Using risk to balance agile and plan-driven methods. *Computer*, 36(6):57–66, 2003.
- [8] Barry Boehm and Richard Turner. Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In *Proceedings. 26th International Conference on Software Engineering*, pages 718–719. IEEE, 2004.
- [9] Barry Boehm and Richard Turner. Management challenges to implementing agile processes in traditional development organizations. *IEEE software*, 22(5):30–39, 2005.

- [10] Robert G Cooper and Anita F Sommer. The agile–stage-gate hybrid model: a promising new approach and a new research opportunity. *Journal of Product Innovation Management*, 33(5):513–526, 2016.
- [11] Tiago Silva Da Silva, Angela Martin, Frank Maurer, and Milene Silveira. User-centered design and agile methods: a systematic review. In *2011 AGILE conference*, pages 77–86. IEEE, 2011.
- [12] Torgeir Dingsøy, Nils Brede Moe, Tor Erlend Fægri, and Eva Amdahl Seim. Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*, 23(1):490–520, 2018.
- [13] Torgeir Dingsøy, Sridhar Nerur, VenuGopal Balijepally, and Nils Brede Moe. A decade of agile methodologies: Towards explaining agile software development, 2012.
- [14] Ken Schwaber et al. Agile manifesto. In *12 principles*, 2001.
- [15] Jennifer Ferreira, Helen Sharp, and Hugh Robinson. User experience design and agile development: managing cooperation through articulation work. *Software: Practice and Experience*, 41(9):963–974, 2011.
- [16] Interaction Design Foundation. *The Basics of User Experience Design*. Interaction Design Foundation, 2018.
- [17] Stephen Griffiths. Mobile app ux principles. *Retrieved September, 26:2015*, 2015.
- [18] Marc Hassenzahl. The thing and i: understanding the relationship between user and product. In *Funology*, pages 31–42. Springer, 2003.
- [19] Tabea Ramírez Hernández, Melanie Kreye, and Steven Eppinger. Applicability of agile and scrum to product-service systems. In *EurOMA Conference*, 2019.
- [20] Timo Jokela. The early phases of ux: Why they are important (more than evaluation), and what they are. In *I-UxSED*, pages 49–51, 2012.
- [21] Sesaria Kikitamara and Anastasia Ary Noviyanti. A conceptual model of user experience in scrum practice. In *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 581–586. IEEE, 2018.

- [22] L. Klein. *UX for lean startups: Faster, smarter user experience research and design*. O'Reilly Media, Inc., 2013.
- [23] Kati Kuusinen, Tommi Mikkonen, and Santtu Pakarinen. Agile user experience development in a large software organization: good expertise but limited impact. In *International Conference on Human-Centred Software Engineering*, pages 94–111. Springer, 2012.
- [24] Marta Kristín Lárusdóttir, Åsa Cajander, and Jan Gulliksen. The big picture of ux is missing in scrum projects. In *Proceedings of the International Workshop on the Interplay between User Experience (UX) Evaluation and System Development (I-UxSED)*, pages 49–54, 2012.
- [25] Marta Kristin Larusdottir, Olöf Haraldsdottir, and Bengt Mikkelsen. User involvement in icelandic software industry. In *Proceedings of the INTERACT*, pages 1–2. Citeseer, 2009.
- [26] Effie Lai-Chong Law. The measurability and predictability of user experience. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, pages 1–10. ACM, 2011.
- [27] Paul McInerney and Frank Maurer. Ucd in agile projects: dream team or odd couple? *interactions*, 12(6):19–23, 2005.
- [28] Claudia de O Melo, Viviane Santos, Eduardo Katayama, Hugo Corbucci, Rafael Prikladnicki, Alfredo Goldman, and Fabio Kon. *Journal of the Brazilian Computer Society*, 19(4):523, 2013.
- [29] Tina Øvad and Lars Bo Larsen. The prevalence of ux design in agile development processes in industry. In *2015 Agile Conference*, pages 40–49. IEEE, 2015.
- [30] MMM Safwan, G Thavarajah, N Vijayarajah, K Senduran, and CD Manawadu. An empirical study of agile software development methodologies: A sri lankan perspective. *International Journal of Computer Applications*, 84(8):1–7, 2013.
- [31] Ken Schwaber and Jeff Sutherland. The scrum guide: The rules of the game. <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>, Novembro 2017. Acessado em 30/03/2019.

- [32] Helen Sharp, Jennifer Preece, and Yvonne Rogers. *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 2019.
- [33] Scrum Official Website. The scrum framework poster. <https://www.scrum.org/resources/scrum-framework-poster>. Acessado em 30/05/2019.