**Introdução**

Atendendo à demanda global intensamente crescente de software, várias técnicas de gestão de desenvolvimento de aplicações surgem com (Abrahamsson et al., 2017) o foco em atender as necessidades do consumidor final (Silva et al., 2011). Naturalmente com uma maior demanda, novos obstáculos surgiram no terreno da tecnologia tais mudanças resultaram na troca de metodologias mais convencionais como o Cascata (Safwan et al., 2013), pelas Metodologias Ágeis, que se destacam por - resumidamente - propõem adaptabilidade, entrega do produto final com maior velocidade diante os métodos convencionais e mais entregas durante o desenvolvimento (Melo et al., 2013). Tal escopo e conceito ágil foi estudado profundamente por (Boehm e Turner ,2004), que identificaram a baseado agile, i.e Métodos Ágeis. Os autores também elencaram quatro elementos: Aplicação, Gestão, Técnica e Pessoas, se. Tais elementos segundo os autores convergem as necessidades que fazem o conceito ser executado com êxito. (Boehm e Turner, 2003, 2004, 2005).

Segundo Jokela (2012), Usabilidade (ISO 9241-210, 2010) e Boa Experiência do Usuário são fatores centrais para classificar bons produtos e sistemas. Ambas têm com o foco no usuário final e sua relação com o produto. Porém (Lárusdóttir, Cajander e Gulliksen, 2012), diz que há duas maiores escalas para mensurar a Usabilidade são elas a Eficiência e a Eficácia (ISO 9241-210, 2010), enquanto a Experiência do Usuário (*UX*) abrange muito mais que isso. Ela tem como objetivo a satisfação do usuário na sua mais extensa pluralidade: desde os sentimentos hedônicos presentes antes da apresentação de um produto até as expectativas durante a utilização. (Hassenzahl, 2003), que transpõem o foco voltado a tópicos e métricas da Usabilidade e visa mais os sentimentos e emoções.

Apesar dos conceitos de Experiência do Usuário (*UX*) e *Scrum* serem muitas vezes aplicadas em conjunto no mercado de desenvolvimento de software (Kikitamara e Noviyanti, 2018) e ser em especial promissora a união. Não há clareza, padrões ou diretrizes a serem seguidos na hora de aplicá-las (Kuusinen, Mikkonen e Pakarinen, 2012), (Ferreira, Sharp e Robinson, 2011), (Silva etal., 2011). Isso causa um atrito, incertezas e questionamentos sobre a integração dos dois elementos, e impede clareza e coesão delas no processo de produção, como não conseguir professionais e usuários devido alguns elementos temporais estabelecidos pelo *Scrum*, como as durações dos ciclos de trabalho (Sprint) pode impedir trabalhos como a pesquisas com usuários ou escrever histórias de usuário (Lárusdóttir, Cajander e Gulliksen, 2012).

Tendo em vista esse panorama, este trabalho se propõe a realizar um estudo sobre abordagens para conciliar o *Scrum* com a Experiência do usuário (*UX*) e assim responder à pergunta “O que é necessário adaptar no fl*UX*o de desenvolvimento de software em para incluir a experiência do usuário?’.Este estudo visa assim desenvolver uma proposta para responder à questão de pesquisa utilizando a classificação em KIKITAMARA e NOVIYANTI (2018).

Para responder essa pergunta será utilizado a seguinte metodologia:

* Realizar um aprofundado estado da arte de trabalhos que conciliam as duas abordagens no cenário;
* avaliar as abordagens apresentadas nos trabalhos segundo KIKITAMARA e NOVIYANTI (2018);
* avaliar a utilização conjunta das duas metodologias em relação a alguns indicadores (Qualidade da aplicação, prazo de entrega e atendimento dos requisitos).

**Referência teórico**

Mesmo com a necessidade de atualização. As ideias de expandir os métodos de desenvolvimento de software vigentes na década de 1990 foram muito restritas e não houve grande crescimento dessas ideias (Agile 101). Porém em 2001 em Utah, um grupo se reuniu para debater ideias, durante a discussão de diferenças e o que de fato funcionava, o grupo criou o que hoje é nomeado como “Manifesto for *Agile Software Development*, que posteriormente daria origem a *Agile Alliance* (Agile 101)(Shwaber et al, 2001).

O Manifesto Ágil virou referência de boas práticas em desenvolvimento de software, que tem como base um conjunto valores a serem seguidos, sendo 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 às mudanças antes de seguir o plano.

Baseado nesses valores o Manifesto (Shwaber et al, 2001) apresenta 12 princípios, 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, entre algumas semanas e 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 em 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 aumentam 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 o que fazer para se tornar mais eficaz, então afina e ajusta seu comportamento de acordo.

Seguindo tais valores e princípios listados anteriormente, o *Scrum* é um método ágil criado, precedendo o surgimento do próprio manifesto, o *Scrum* é uma das metodologias ágeis mais coesas para as demandas de desenvolvimento de software e uma das mais aplicadas no mercado (Dingsøyr,2012). Cerca de um terço (1/3) dos desenvolvedores da Europa desenvolvida, usavam esses processos em seus projetos em 2009(Larusdottir, Haraldsdottir, Mikkelsen, 2009), e essa popularidade tem seguindo uma continua ascensão no mercado da tecnologia (Dingsøyr et al, 2018).

Não cabe ao escopo do trabalho explicar detalhadamente todas as partes componentes do *Scrum*, mas podemos – resumidamente - dizer que a estrutura não é um processo linear, mas um conjunto de conceitos e técnicas em que você pode usar vários outros processos ou técnicas. Analise a eficácia relativa das práticas de gerenciamento e desenvolvimento para que você possa melhorá-las. Este não é um processo prescrito, ou seja, não descreve como lidar com todas as situações, porque é impossível prever o que acontecerá em um trabalho complexo. É um conjunto de valores, princípios e práticas que fornecem a base para a organização agregar práticas específicas de engenharia e gestão, e estão relacionados com a situação real da empresa. O resultado será sua própria versão do *Scrum*.

O *Scrum* tem seu foco no gerenciamento e organização em projetos onde é difícil planificar o futuro. O mecanismo de controle é empírico, onde ciclos de feedback constituem a área central da técnica que é oposta ao tradicional gerenciamento de comando e controle. Este é um método de planejamento e gerenciamento de projetos que pode aumentar a autoridade de tomada de decisão para níveis das propriedades operacionais e determinísticos.

A Experiência do usuário – Do inglês User Experience (*UX*) - é a junção de fatores e elementos ligados à interação de um usuário com um determinado produto, cujo o resultado resulta um sentimento, negativo ou positivo. O termo surge a primeira vez por Donald Norman em 1990, vice-presidente do Grupo de Tecnologia Avançada na Apple (Agni, 2016). Para Norman *UX* é mais que apenas design, mas sim abraçar aspectos ligados aos sentimentos e experiências. Por se tratar de percepção pode-se dizer que *UX* é subjetiva, pois nasce de um crescimento inerente a cada ser. Ela pode ser dinâmica por se molda ao longo do tempo com as evoluções apresentadas (Interaction Design Foundation, 2021).

O design de Experiência de Usuário, tem como alvo modelar essa experiência do usuário de forma que seja a melhor possível para a utilização de um produto ou serviço. Para fazer essa modelagem designer *UX* considera uso de certas questões a serem feitas “Por que?”, “O que” e “Como, que trazem como reposta as motivações, as utilizações do produto – Funcionalidade – e qual seria o caminha mais agradável para atingir o objetivo (Interaction Design Foundation, 2018).

Desde de 2005, a complexidade dos softwares vem aumentando de forma considerável. Dessa forma, a interatividade com tecnologias e o que elas podem oferecer ao usuário (Interaction Design Foundation, 2021). Essa demanda por novas funcionalidades se apresentou cada vez mais. Mas para que se possa resumir o sucesso de um produto de software precisamos saber como o usuário se sente em relação ao produto. (jokela, 2012)(Interaction Design Foundation, 2021)

Para isso, levasse em consideração a Experiência de Usuário durante no esqueleto do software, e recorrentemente são levantadas as seguintes questões:

* 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 que a algum software atingiu uma “Boa” Experiência do Usuário, ela deve responder “Sim” para as três perguntas acima.

Desde modo, para atingir esse objetivo proposto, é importante projetar o software com um foco especial no usuário, em todo o ciclo de vida do projeto, desta forma ela servirá como farol e guiará do desenvolvimento sempre evidenciando quem irá utilizar o produto final.

**Experiência do Usuário e *Scrum***

Enquanto de um lado temos o *Scrum* ganhando cada vez mais os holofotes do desenvolvimento pode ser uma metodologia eficiente e eficaz, do outro temos a Experiência de Usuário que se torna cada vez um bom indicador de qualidade no produto (Lárusdóttir, Cajander e Gulliksen, 2012). O grande obstáculo é a junção desses conceitos dentro da indústria de software, pois não existem padrões e para uni-los de maneira clara e coesa.(Kikitamara, Noviyanti,2018), (Lárusdóttir, Cajander e Gulliksen, 2009),(Lárusdóttir, Cajander e Gulliksen, 2012).

No seu trabalho de 2018, Sesaria Kikitamara e Anastasia Ary Noviyanti, da Universitas Prasetiya Mulya, produziram um modelo conceitual para a junção de Experiência do usuário e a metodologia *Scrum* (Kikitamara, Noviyanti,2018), expondo um rico estado da arte de estudos que utilizaram o método a partir de 2005.

Este trabalho se baseia no estudo realizado por Kikitamara e Noviyanti mencionado anteriormente, que classifica, basicamente, essa junção em três categorias:

* Trabalho Paralelo;
* Trabalho na mesma Sprint;
* *Lean UX*;

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.

Na abordagem **“Trabalho Paralelo"**, os times de desenvolvimento e *UX* trabalham após sincronizarem e planejarem suas atividades. O time *UX* pode avançar até 2 (dois) ciclos para ter dados do usuário e prototipar telas e propor agregadores baseados no usuário ao projeto.

Quando surge novas funcionalidades, pode ocorrer que a prototipação não seja terminada, tal problema seria respondido pelo trabalho paralelo. Geralmente o time de *UX* avança nos ciclos de desenvolvimentos para ter vantagem na análise dos dados e propor soluções.

Na abordagem **“Trabalho na mesma *Sprint*”** – Considera-se Sprint um ciclo de desenvolvimento na metodologia *Scrum* –, o time de *UX* trabalha simultaneamente com o time de desenvolvimento durante todo o projeto.

Ao mesmo tempo que o time de desenvolvimento está realizando as atividades de realiza os levantamentos necessários para o ciclo de desenvolvimento o time de *UX* realiza os levantamentos para o design da mesma e dá início a coleta de dados do próximo ciclo.

Na abordagem **“Lean *UX*”** – Lean vem do inglês e significa “Enxuto” -, o usuário tem um maior envolvimento durante o desenvolvimento participando várias vezes dos ciclos. O Time de *UX* ficará focado no MVP (*Minimal Viable Product* - Produto Mínimo Viável).

Afim de diminuir o trabalho desnecessário e o desperdício de recursos, que não atendem as necessidades propostas para o produto final, essa abordagem utiliza o MVP como uma representação do produto final, que deixa claro quais as funcionalidades o mesmo deve ter, sempre procurando a validação do usuário que estará recorrentemente presente durante as etapas e desenvolvimento.

Também é importante que toda equipe trabalhe em conjunto, focando em sua área, mas dando suporte para todos caso seja necessário. Aqui, há foco no *feedback* do usuário e, a partir disso, as ações são tomadas.

**Metodologia**

A partir de alguns artigos que mostraram relevância, mencionados no estudo Kikitamara e Noviyanti (2018), será feita uma análise e dada uma classificação referente as abordagens citadas nos parágrafos anteriores.

Este estudo tentará obter por revisão bibliográfica pelo menos 2 estudos mais recentes quanto possível que se encaixem em cada abordagem.