Inicialmente o processamento de dados estavam ligados fielmente a parte física do computador, não existia programas instalados e cada necessidade demandada o computador devia ser alterado fisicamente. Em 1945 o matemático John von Neumann apresentou a ideia do primeiro computador com uma série de instruções armazenadas internamente que poderiam ser combinadas criando uma vasta quantidade de programas úteis, computador nomeado como EDVAC (Wilkes, 1956).

Inicialmente o processamento de dados estavam ligados fielmente a parte física do computador, não existia programas instalados e cada necessidade demandada o computador devia ser alterado fisicamente. Em 1945 o matemático John von Neumann apresentou a ideia do primeiro computador com uma série de instruções armazenadas internamente que poderiam ser combinadas criando uma vasta quantidade de programas úteis, computador nomeado como EDVAC (Wilkes, 1956).

Desde lá, a indústria de software tem crescido a se tornar umas das mais importantes da era moderna. Não existe forma de fugir do software, está presente em todas as atividades desde as mais simples como fazer compras ou até as mais complexar como dirigir carros. Essa demanda de todas as áreas por softwares que atendam as necessidades nas mais diferenciadas escalas, torna o mercado de software cada vez mais competitivos, se esforçando para trazer novas tecnologias, produtos e serviços. O sucesso de uma empresa pode está maioria das vezes entrelaçado com a frequência com o que elas lançam essas inovações diferenciando-as dos seus concorrentes (Prikladnicki, Miliani e Willi, 2014).

Neste contexto, na década de 1990, é introduzido o método novo de desenvolvimento de software chamado Métodos Ágeis (Prikladnicki, Miliani e Willi, 2014). Tal metodologia vinha com uma forma nova de pensamento sobre o desenvolvimento de software, que a diferenciava dos métodos nomeados como clássicos que tinham seu foco voltado a documentação, resultando num software “pesado” e de difícil sustentabilidade (Soares, 2004). O foco da metodologia ágil em pessoas e seu conjunto de valores, tornou ela uma ótima alternativa para respostas rápidas para mudanças de um mercado em constante alteração (Soares, 2004).

Dessas metodologias ágeis, o Scrum é uma das mais difundidas entre os projetos de desenvolvimento de software, o método foi apontado como sendo usado por 2 a cada 3 participantes da pesquisa realizada pela VersionOne em 2011 (VersionOne, 2011). Mesmo com uma aceitação mundial ótima e trazendo resultados para as empresas independente do porte que o adotaram (Sabbagh, 2014), não implica que ele seja perfeito e tenha todos os seus problemas resolvidos. Alguns atrelados a casos específicos da equipe de desenvolvimento, apontados por (Virgínia, 2010). Ou até mesmo no gerenciamento de riscos como elencado por (Soares, 2004) em seu estudo. Mesmo que essas metodologias prezem por um conjunto de valores, elas ainda estão ligadas a premissa de entregar o produto o mais rápido e com o menor custo possível com a melhor qualidade possível, esse ponto pode ser um calcanhar de Aquiles no desenvolvimento de software (Prikladnicki, Miliani e Willi, 2014).

Mas o que é essa tal qualidade? É válido, pensar o conceito de qualidade; A ideia de por muitas vezes pode ser vista como simples e facilmente definida, mas ao se parar para analisar ao longo do tempo se torna cada vez mais complexo se definir (Koscianski e Soares,2007). Segundo Herman G. Weinberg: “A qualidade é relativa. O que é qualidade para uma pessoa pode ser falta de qualidade para outra”. Trazendo essa discussão para o ambiente de desenvolvimento de software, não existe um consenso entre autores sobre o que é qualidade de software, apenas concordam que essa meta a ser perseguida (Rocha e Campos, 2012).

Segundo Stahl (1988), existe um ponto de convergência entre os autores, que um software de qualidade é aquele que atende as necessidades do usuário. Mas essa meta não surge do nada no produto final, ela é um compilado de várias etapas tanto de requisitos, codificação, avaliação e etc. Visando atingir essa qualidade a Experiência do Usuário (*UX*) tem como objetivo levar em consideração 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).

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 levantamento bibliográfico sobre trabalhos que conciliem o *Scrum* com a Experiência do usuário (*UX*) e assim responder à pergunta “O que é necessário adaptar no fluxo de desenvolvimento de software 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).

Wilkes, M. V. (1956). Automatic Digital Computers. New York: John Wiley & Sons. pp. 305 pages. QA76.W5 1956.

Prikladnicki, R, Miliani, F., Willi, R. Métodos Ágeis Para Desenvolvimento De Software. Bookman Companhia Ed, 2014. v. 1.

SOARES, M. S., Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software, Minas Gerais, 2004, p1-6.

|  |  |
| --- | --- |
| State of Agile Survey Results 2011 by Version One Inc. acessado 09 de maio de 2021 <https://stateofagile.com/#ufh-i-613555398-6th-annual-state-of-agile-report/7027494>  Sabbagh, R., Scrum: Gestão ágil para projetos de sucesso, Casa do código, 2014, v1 |  |

Virgínia C. Et al. Estudo de Caso da Utilização de Scrum no Desenvolvimento Distribuído de Software, Pernambuco, 2010, p 1 -11.

Koscianski, A. and dos Santos Soares, M., Qualidade de Software, 2007, Novatec, v 2.

STAHL, Marimar M. Avaliação da qualidade de software educacional: relatório técnico do Programa de Engenharia de Sistemas e Computação. Rio de Janeiro: UFRJ, COPPE, 1988.

Rocha, A. Campos, G. AVALIAÇÃO DA QUALIDADE DE SOFTWARE EDUCACIONAL, Brasília, ano 12, n.57, jan./mar. 2012.

Metodologia ágeis

Mesmo com o surgimento em na década de 1990, as metodologias ágeis só se popularizaram a partir de 2001, quando um grupo de 17 especialistas em desenvolvimento de software nas diversas metodologias se uniram e definiram princípios que seriam divididos entre esses métodos (Soares, 2004). Essa reunião resultou na Aliança Ágil e o estabelecimento do Manifesto Ágil.

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.

O manifesto ágil salienta tudo o que tem de mais valor nas metodologias ágeis. Burocracia tem seu valor para o desenvolvimento de software, mas não são mais importantes que saber se relacionar com pessoas e ter o cliente sempre presente para encontrar soluções de qualidade e se adaptar para mudanças. Essas metodologias propõem em um tempo menor a obtenção de resultados, para isso tirasse o foco dos processos e coloca no produto.

Scrum

O Scrum é um framework de gerenciamento de projetos, da organização ao desenvolvimento ágil de produtos complexos e adaptativos com o mais alto valor possível. O Scrum possui seu foco no gerenciamento e projeto da organização onde é difícil planejar à frente. Há maneiras de controle empírico, onde ciclos de feedback constituem o núcleo da técnica de gerenciamento que são usadas.

Schwaber, K. e Sutherland, J. The Scrum Guides.,2020, < https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>*. Consultado em 10 de maio de 2021*