

Análise da Percepção de Importância de Requisitos de Usabilidade no Desenvolvimento de um Sistema Web com *Scrum*

Críssia Marcelino¹, Francisco Nascimento²

¹Universidade Federal de Pernambuco (UFPE)

²Instituto Federal de Pernambuco (IFPE)

crissiasantana@gmail.com, francisco.junior@jaboatao.ifpe.edu.br

Abstract. *The agile methodologies are used in software development projects, for their dynamic and efficient profile, to promote attending the customer satisfaction. However, in many cases, these methodologies seem to be used solely, without analyzing important aspects in the user-system relationship. Thus, this paper describes a case study about the perceived importance of usability in a Scrum team from a retail software company, and as a significant outcome of this work, it worths emphasizing that the team in question do not take into consideration the user study as essential practical task in order to address the usability requirements.*

Resumo. *As metodologias ágeis são utilizadas nos projetos de desenvolvimento de softwares, por seu perfil dinâmico e eficiente, visando atender a satisfação dos clientes. Porém, em muitos casos, essas metodologias são utilizadas isoladamente, sem considerar aspectos importantes na relação usuário-sistema. Dessa forma, o artigo descreve um caso de estudo sobre a importância percebida da usabilidade em um time Scrum de uma empresa de softwares para o varejo e, como significativo resultado deste trabalho, vale ressaltar que o time em questão não leva em consideração o estudo do usuário como prática essencial para atender requisitos de usabilidade do sistema.*

1. Introdução

A Engenharia de *Software* (ES) é caracterizada por padrões e métodos que norteiam o processo de desenvolvimento de *software*. Dentre os vários processos de desenvolvimento de *software* praticados pelo mercado de produção de sistemas, destaca-se a metodologia ágil. Segundo [Soares, 2004] essa escolha é baseada principalmente na versatilidade dos métodos ágeis, pois se adequam facilmente às mudanças, reduz a complexidade do planejamento, propõe um desenvolvimento rápido, aumentando o foco nas pessoas e não nos processos. Assim a equipe tende a ser mais colaborativa, visando entregar um sistema de alta performance.

Entende-se que é responsabilidade da equipe desenvolver um *software* usável e que atenda as expectativas dos clientes. Porém, apesar da interação com o cliente ser essencial no desenvolvimento ágil de *softwares*, pouco se fala da relação entre os métodos ágeis com aspectos que podem melhorar a qualidade do sistema para seu usuário final, como a usabilidade e o estudo do usuário.

Assim quando a aplicação de metodologias ágeis não leva em consideração aspectos de usabilidade, os sistemas, mesmo sendo planejados, desenvolvidos e entregues em concordância com os prazos, continuam correndo o risco de não serem aprovados por seus usuários. Seria natural então, em um cenário onde abordagens ágeis são interessantes alternativas para projetar um *software*, que pensássemos também em ações da área de

usabilidade, pois essas disciplinas se complementam em alguns fatores que influenciam no valor agregado do produto final, como a garantia de satisfação e qualidade [Oliveira, 2016].

Apesar da relevância de uma pesquisa que discuta e analise a observância conjunta de elementos de Metodologia Ágil e Usabilidade, são raros os estudos de casos mostrando essa interação. [Silva, 2010] iniciou uma discussão sobre a relação de métodos ágeis com Usabilidade, apresentando conceitos, as principais dificuldades em integrá-los e especificou as práticas ágeis mais adotadas (Sprints, Daily, User stories), técnicas de modelagens (protótipos, cenários de uso) e instrumentos (Designs Conceituais, Personas, testes rápidos);

[Cybis, 2009] implantou um estudo para avaliar a relação ‘abordagem ágil *versus* engenharia de usabilidade’ numa empresa de segurança da informação em Porto Alegre. O autor verificou as boas práticas que garantem o sucesso dessa interação, como a importância em fazer o *software* de acordo com protótipos; constatou também que as *Sprints* funcionam em vários ciclos do processo, como na análise, concepção e testes. Comprovou que testes rápidos alimentam as definições da próxima *Sprint*, o que garante a dinâmica ágil e produtiva. Porém, também verificou desafios a serem superados, como a concepção de interfaces em pares (programadores e designs). Cybis constatou que essa interação propõe uma maior análise dos processos de execução do sistema, mas também compromete a agilidade da entrega, necessitando assim de aperfeiçoamentos.

[Leite, 2013] aplicou um estudo de caso, avaliando duas equipes (uma ágil, denominada como equipe colaborativa e uma equipe tradicional) em uma empresa de desenvolvimento de *software*. As duas equipes possuíam experiências equivalentes em usabilidade. O estudo propôs uma análise das técnicas utilizadas pelas equipes e comprovou que a equipe colaborativa encontrou mais problemas de usabilidade no sistema; apresentou maior produtividade e mais compartilhamento de conhecimentos; demonstrou maior alinhamento para integração em processos e ciclos iterativos. O artigo comprovou no fim, a maior eficiência das equipes ágeis.

[Leite, 2013] corrobora que alguns desses poucos estudos mostraram que em ambientes onde são utilizados métodos ágeis ainda há muitos problemas a serem resolvidos quanto à integração com práticas de usabilidade. Diante da necessidade de maior explanação da temática, justifica-se o artigo aqui proposto, pois se verificou que essa discussão ainda é emergente, assim como observou-se, em um determinado ambiente profissional, a necessidade de analisar a importância que os membros da equipe fornecem à usabilidade no processo de construção de uma sistema.

Diante deste cenário, esse artigo tem como objetivo geral analisar a relação da Metodologia Ágil com aspectos de Usabilidade no processo de desenvolvimento de *softwares*. E como objetivos específicos (1) mapear como são tratados e definidos os aspectos de usabilidade no processo de planejamento e desenvolvimento de *softwares* construídos a partir de metodologias ágeis; e (2) verificar se a equipe que define o processo e desenvolve o sistema considera o usuário como parte essencial no processo de aceitação do produto.

Para tentar alcançar esses objetivos pretende-se entrar no universo de um projeto desenvolvido sob a vertente da metodologia ágil *Scrum*. A equipe estudada faz parte de uma empresa de desenvolvimento de *software* para o varejo. Os resultados obtidos e aqui apresentados podem contribuir como incentivo para exploração/criação de pesquisas mais abrangentes sobre a temática.

2. Métodos Ágeis

O conceito de Método Ágil tem como prioridade a “satisfação do cliente através de entregas antecipadas e contínuas de *software* de valor [Silva, 2010]”. A partir desta premissa muitas organizações adotam processos ágeis no desenvolvimento de seus projetos com o intuito de oferecer o melhor produto no menor tempo.

Os princípios dos modelos ágeis deram origem ao Manifesto Ágil em 2001. O Manifesto surgiu estabelecendo as seguintes premissas:

- Indivíduos e interações mais que processos e ferramentas.
- *Software* em funcionamento mais que documentação abrangente.
- Colaboração com o cliente mais que negociação de contratos.
- Responder a mudanças mais que seguir um plano.

Entende-se assim, que as principais características da aplicação de métodos ágeis são: mais produtividade e menos custos, mais empenho e satisfação por parte dos colaboradores, entregas de *software* mais rápidas, maior qualidade no produto final e maior satisfação dos *stakeholders*¹.

Metodologias ágeis atuam no refinamento de metodologias interativas, tirando o foco do processo em si e dando mais ênfase para a contribuição das pessoas, assim a integração da equipe é essencial para entender como o desenvolvimento funciona e para definição da melhor abordagem para execução do trabalho. Para [Leite 2013] as reuniões diárias, prática comum aos modelos ágeis, são o principal momento onde as informações são trocadas, a equipe fica sabendo das atividades e dificuldades uns dos outros, assim, podem ajudar e aprender. Estas reuniões também são uma forma de incentivar o relacionamento entre a equipe.

Uma das metodologias ágeis mais difundidas da Engenharia de *Software* é a *Scrum*. Uma das características principais do método *Scrum* é se preocupar com a gestão do projeto e não com processos e controles. Assim com o *Scrum* é possível descobrir pontos falhos nos processos e como corrigi-los. O *Scrum* será detalhado a seguir.

2.1 Scrum

O *Scrum* é um método de gerenciamento que busca a flexibilidade dos prazos e resultados, utilizando times pequenos que colaboram entre si. Esse método não dispõe de técnicas ou processos específicos para o desenvolvimento do projeto, apenas estabelece conjunto de regras e práticas gerenciais que devem ser adotadas para o sucesso de um projeto [Carvalho & Melo, 2009]. Assim, fica a cargo da equipe definir a organização do processo, a execução das atividades, buscando sempre o sucesso na entrega do projeto.

O *Scrum* possui responsabilidades (*Product Owner, Scrum Master, Team*), atividades (*Sprint planning Daily Scrum, Sprint Review*) e artefatos (*Product Backlog, Sprint Backlog*). Onde os responsáveis planejam a construção do produto. As *Sprints* são iterações do projeto, que podem durar de duas a quatro semanas, e *Backlog* consiste em uma lista das atividades a serem realizadas durante o projeto. No início de um *Sprint* é feita uma reunião de planejamento, na qual são selecionados determinados itens do *Backlog* que serão executados no período [Leite, 2013].

No *Scrum* é empregado o conceito de *time-boxes*, no qual cada atividade tem seu tempo estipulado, criando um controle na qualidade das funcionalidades e na produção.

¹ Público estratégico. Grupo ou pessoa que possui participação, investimento ou ações e que apresenta interesse em uma determinada empresa ou negócio. (BEZERRA, 2014).

Essa autonomia faz com que a equipe se torne mais interativa e confiante, promovendo assim um desenvolvimento mais conciso.

A interação entre as pessoas que compõem a equipe *Scrum* é primordial, assim como a interação do usuário com o sistema. Para melhorar a relação usuário-sistema costuma-se aplicar aspectos de usabilidade.

3. Usabilidade

O termo usabilidade está relacionado com a facilidade de aprendizado e uso do sistema, bem como a satisfação do usuário em decorrência desse uso. Para [Nielsen e Loranger 2007, p.xvi] a usabilidade é um atributo de qualidade relacionado à facilidade do uso de algo. Mais especificamente, refere-se à rapidez com que os usuários podem aprender a usar alguma coisa, a eficiência deles ao usá-la, o quanto lembram daquilo, seu grau de propensão a erros e o quanto gostam de utilizá-la. Assim, [Siebra, 2011] afirma que especialmente no contexto de *software*, a interface deve ser configurada de forma a não exigir sempre dos usuários um novo aprendizado para associar comandos e ações, tornando a interação e uso mais “intuitivos”, facilitando assim a ação e minimizando a probabilidade de erros.

Pensar na usabilidade de determinado sistema é sinônimo de pensar no usuário, em qualquer etapa do projeto, seja no início, meio ou no fim. Quando não se pensa na usabilidade desde o princípio do projeto pode-se perder tempo e recursos para, posteriormente, corrigir os eventuais erros e inconsistências que emergirão [Siebra, 2011]. Dessa maneira:

a concepção de sistemas muitas vezes prioriza as exigências da informática antes de responder àquelas relacionadas ao usuário. Na maioria das vezes os profissionais de informática (projetistas) se empenham antes de tudo em definir as funções lógicas de um sistema sem de fato se preocuparem com as necessidades e habilidades físicas e cognitivas do usuário. [Silva, 2007 apud Ignácio; Carvalho, 2008].

No contexto do desenvolvimento ágil, a usabilidade ainda é tratada como algo complementar. Sendo apenas modificada quando há uma solicitação por parte do cliente, que em muitas vezes também não tem conhecimento sobre a importância da usabilidade no desempenho do trabalho do usuário final.

Existem várias estratégias para realizar a avaliação de usabilidade de um sistema, entre elas: os requisitos não funcionais de usabilidade [Pressman, 1992; Ferreira e Leite, 2003], Critérios Ergonômicos de [Scapin e Bastien 1993], Heurísticas de Usabilidade de [Nielsen 1994], e os princípios de Diálogo da ISO 9241:10, entre outros.

Para esta pesquisa abordaremos os requisitos não funcionais de usabilidade, por serem mais adequados para o contexto estudado.

3.1 Requisitos Não-funcionais (RNF)

Os RNFs, para qualquer tipo de sistema, estão relacionados a aspectos de *software*, *hardware* ou fatores externos, que determinem condições ou restrições ao comportamento do sistema pretendido [Sommerville, 1997, 2004] e [Pressman, 2004]. Entre estes requisitos é possível citar: desempenho, segurança, portabilidade, confiabilidade, manutenibilidade, acessibilidade, usabilidade, etc. Assim como os demais sistemas, os sistemas de *softwares* para varejo necessitam atender aos RNFs, os quais devem ser especificados com clareza.

De acordo com Pressman, os requisitos não funcionais de usabilidade da interface podem ser agrupados em duas categorias: os relacionados à entrada de dados e os

relacionados à exibição da informação. [Pressman, 1992 *apud* Ferreira; Leite, 2003, p. 117]. Os requisitos relacionados à exibição da informação são:

Quadro 1: Requisitos não funcionais de exibição da informação.

Requisito	Descrição
Consistência	Uniformidade dos comandos, funções, padrões.
Feedback	Resposta adequada do sistema ao usuário a toda e qualquer atividade.
Níveis de habilidade e comportamento humano	Garantia de acomodação de personalidades distintas, aplicando conhecimentos individuais.
Percepção humana	Preocupação com a qualidade de captação da informação.
Uso de metáforas	Facilitação da explicação e/ou compreensão de interfaces combinando conhecimento familiar e novos conceitos.
Minimização de carga de memória	Disponibilização de recursos que remetam ao cotidiano dos indivíduos
Eficiência do diálogo, movimento e pensamento	Simplicidade, clareza e objetividade na definição dos passos para realizar tarefas.
Classificação funcional dos comandos	Agrupamento das funcionalidades no menu do sistema de acordo com critérios conhecidos pelo usuário.
Exibição exclusiva de informação relevante	Eliminação de excesso informacional e de redundância de acessos a operações.
Uso adequado de janelas	Cautela na utilização de janelas simultâneas, a fim de evitar sobrecarga de informações.
Manipulação direta	Objetos visuais devem estar nítidos durante sua utilização.
Uso de rótulos, abreviações e mensagens indicativas	Clareza, objetividade e significância na elaboração destes elementos.
Projeto independente da resolução do monitor	Definição da aparência dos componentes utilizando proporcionalidade em relação ao espaço disponível.

Categoria Entrada de Dados

Muito tempo de trabalho do usuário é gasto com a escolha de comandos, digitação de dados e outros *inputs*. Um bom sistema deve otimizar ao máximo o tempo que o usuário gasta com essas tarefas. As diretrizes apresentadas a seguir tornam o sistema mais poderoso no que diz respeito à entrada de dados [Pressman, 2004].

Quadro 2: Requisitos não funcionais de Entradas de Dados.

Requisito	Descrição
Mecanismos de ajuda	Disponibilização de informação de ajuda para toda ação de entrada.
Prevenção de erros	Desativação de caminhos inválidos para evitar erros do usuário.
Tratamento de erros	Recursos que permitam a correção de erros de forma ágil.

4. Metodologia

A pesquisa é de natureza exploratória, pois se pretende levantar e analisar opiniões e práticas ainda desconhecidas sobre um grupo específico. Quanto aos objetivos, esta pesquisa tem caráter descritivo. Segundo [Gil 2008], as pesquisas descritivas possuem como objetivo a descrição das características de uma população, fenômeno ou de uma experiência. Quanto à fonte de dados pode-se considerar esta pesquisa como bibliográfica. Para [Marconi e Lakatos 2007], a pesquisa bibliográfica oferece meios para definir, resolver, não somente problemas já conhecidos, como também explorar novas áreas onde os problemas não se cristalizam suficientemente, tendo como objetivo permitir

ao pesquisador um reforço na análise de suas pesquisas ou manipulação de suas informações. Considerando as abordagens de análise dos dados, esta pesquisa se caracteriza como quantitativa e qualitativa [Gil, 2008]. Com objetivo de derivar conclusões de forma clara e sistemática, evidenciada a partir dos dados coletados.

Também foi utilizado na pesquisa o método indutivo de abordagem. Justifica-se a utilização do método indutivo, pois a partir de dados coletados de um grupo de participantes, deseja-se chegar a uma teoria fundamentada, a partir de interpretações e abstrações, objetivando identificar e analisar fatores e aspectos [Felix, 2011].

Quanto ao procedimento adotado nesta pesquisa, destaca-se o estudo de caso holístico de caso único, onde um estudo de caso estuda um fenômeno dentro do seu contexto, principalmente quando os limites entre o fenômeno e o contexto não estão claramente definidos [Yin, 2005 *apud* Felix, 2011]; holístico porque a unidade de análise são as práticas aplicadas ao desenvolvimento do projeto pelo engenheiro de software (desenvolvedores, arquitetos de softwares, analista de testes), pois o objetivo da pesquisa é analisar os aspectos de atuação destes indivíduos ligados ao projeto; e único porque considera a organização estudada como caso representativo, que se deseja investigar a utilização das práticas de usabilidade com o método *Scrum*.

O universo da pesquisa baseia-se em uma equipe que desenvolve *softwares* direcionada pelo método *Scrum*. Esta equipe faz parte de uma empresa do segmento de *softwares* para o varejo, atuando há mais de 30 anos no mercado. Sua sede está situada em São Paulo (SP), mas possui filiais espalhadas pelo Brasil, em cidades como Belo Horizonte (BH), Rio de Janeiro (RJ), Manaus (AM), Recife (PE), Franca (SP), Porto Alegre (RS), entre outras unidades, possuindo mais de 2.000 funcionários.

A equipe analisada faz parte das filiais Recife (PE) e Franca (SP) e contém 11 participantes, sendo 1 representante do cliente, 1 gerente do projeto, 1 analista de serviços, 2 analistas de suporte, 4 desenvolvedores e 2 analistas de testes.

Assim, os participantes foram contatados por e-mail e após aceitarem o convite foi enviado o link de endereço do questionário. Foi aplicado à equipe 12 questões envolvendo os RNFs de usabilidade relacionados ao desenvolvimento via metodologia *Scrum*. Essa medida visa levantar/analisar a percepção dos envolvidos em relação à interação do método ágil com os RNFs.

Para a aplicação do questionário foi utilizada a ferramenta aberta *Google Forms*, com o objetivo de melhor estruturar as respostas obtidas. Foi dado o prazo de cinco dias (de 03/07/2017 a 07/07/2017) para o preenchimento do questionário. Após esta etapa, os dados foram compilados e apresentados em figuras geradas pela própria ferramenta. A análise dos dados foi feita de forma qualitativa, a fim de fundamentar os resultados obtidos e evidenciados a partir dos dados.

Vale salientar que o cenário estudado é restrito e os resultados não devem ser generalizados como práticas comuns às demais equipes *Scrum* espalhadas pelas diversas empresas de softwares existentes.

5. Resultados

A priori foi feita uma análise dos aspectos intrínsecos à equipe estudada. Assim, percebeu-se que o projeto apresenta um escopo definido, com etapas de planejamento, definição das atividades, delegação das tarefas e prazos a serem atendidos. Dentro desse contexto, a equipe trabalha de forma dinâmica com *sprints* quinzenais.

Ressalta-se que dos 11 integrantes da equipe, dois não responderam a pesquisa, inclusive o representante do cliente, que não respondeu ao *e-mail* de solicitação de

participação na pesquisa e um analista de suporte, que estava de férias no período de vigência do questionário. Salienta-se também que nesse estudo não foi feito um levantamento sobre o nível de conhecimento dos participantes sobre usabilidade

5.1 Usabilidade aplicada à metodologia Scrum

Ao serem questionados na pergunta 1 se há aplicação do estudo de usuário no planejamento do projeto, a maioria, 87% informou que sim, que há levantamento de necessidades e opiniões dos clientes. Vale salientar que no cenário estudado o cliente não é o usuário final do sistema, mas sim os operadores de frente de caixa das lojas. Estes não são consultados quando algum projeto é criado. Então, pode-se observar e presumir que no contexto do estudo o cliente pode não ser o melhor ponto focal para, de forma contínua, obter informações sobre as reais necessidades de quem opera o sistema, comprometendo seu desempenho e eficiência.

Já a pergunta 2, refere-se aos RNFs de usabilidade relatados nas Especificações, assim 45% falaram que as especificações abordam questões de usabilidade, 44% disseram que às vezes abordam e 11% informaram que não abordam. Compreende-se assim, que apesar da necessidade de clareza que a documentação deve conter, ainda é falha a questão da especificação dos RNFs nos projetos desenvolvidos na empresa. Provavelmente esse fato está ligado à cultura implantada, de que esses aspectos são considerados intrínsecos ao profissional que desenvolve o *software*. Normalmente o especificado refere-se à estrutura do código e ao retorno obtido quando determinadas funções são acionadas. Porém, em muitas situações, a não contemplação dos RNFs nas especificações compromete, consideravelmente, a utilização do sistema.

Essa suposição pode ser confirmada com os dados levantados na pergunta 3. Pois quando questionados se as instruções de usabilidade especificadas são suficientes para o entendimento e utilização do sistema por parte dos usuários, obtivemos o seguinte resultado: 43% afirmaram que às vezes as instruções de usabilidade são suficientes, 29% acreditam que as instruções não são suficientes e 28% acreditam que as instruções atendem completamente as necessidades dos usuários.

Entende-se, assim, uma divergência na opinião dos entrevistados, pois mesmo acreditando que as especificações contemplam os RNFs de usabilidade, os entrevistados alegaram também que nem sempre o que é especificado, é de fato entendido pelo usuário. O reconhecimento dessas dificuldades por parte do time faz-se entender que em muitas ocasiões quem está desenvolvendo tem problemas para traduzir ao cliente como a ferramenta deve ser utilizada.

Outro ponto referente aos RNFs foi o tratamento de mensagens de erros. Na Pergunta 4 todos os entrevistados responderam que o sistema contempla mensagens de erros. Porém, ao serem questionados sobre a clareza dessas mensagens na pergunta 5, 63% acreditam que nem sempre as mensagens retornadas informam claramente aos usuários o motivo exato que causou o erro no sistema; já 25% acreditam que as mensagens exibidas informam adequadamente ao usuário sobre os erros ocorridos e 12% acreditam que as mensagens não traduzem o ocorrido. Esse fato corrobora possíveis problemas de interação entre usuário e *software*. Pois para ser considerado fácil de usar, o sistema deve ser intuitivo, o usuário deve se sentir familiarizado com termos e funções. A falta de interação entre usuário e sistema acarreta retrabalhos e solicitações de mudanças no sistema, como mostra a questão seguinte.

Já quando se refere a mudanças no projeto (Pergunta 6), 56% informaram que sempre há mudanças e outros 44% informaram que as mudanças ocorrem ocasionalmente. Porém quando questionados em que momento as solicitações de mudanças ocorrem (Pergunta 7), todos afirmaram que as mudanças ocorrem durante o

desenvolvimento. E dentre os motivos mencionados para tais mudanças (Pergunta 8), 80% informaram que as mudanças derivam de fatores externos, como solicitações de clientes, dificuldades na compreensão do sistema, mudanças de *softwares/hardwares*; já 20% acreditam que as mudanças ocorrem pela falta de conhecimento/entendimento da equipe desenvolvedora. Esse cenário denota possíveis problemas de comunicação entre o solicitado pelo cliente, e o que é entendido pelo time que desenvolve o projeto. Seria interessante, que a equipe buscasse formas mais objetivas de levantar e identificar as reais necessidades dos clientes, sem espaço para dúvidas.

Ao serem questionados sobre a utilização de práticas de usabilidade adotadas individualmente ou pela equipe (Pergunta 9), 63% informaram que não adotam nenhuma prática de usabilidade e 37% afirmaram a adoção. A respeito das práticas adotadas, foi mencionado o seguinte: ***“A prática adotada é de paralelismo entre Certificação e Homologação do produto. Este fator é utilizado devido à complexidade das metodologias ágeis e busca da maior qualidade do produto tendo em vista que a carga de testes pode ser compartilhada entre a equipe.”***; entende-se aqui que os testes são divididos entre a equipe de desenvolvimento e a equipe de testes, para que assim haja uma cobertura maior dos testes; já outro entrevistado respondeu: ***“Destaque de funções mais utilizadas no sistema pelo usuário, afim de facilitar e agilizar o uso do mesmo. A ideia principal ao utilizar esta prática foi de ajudar o usuário durante a sua navegação, evitando que ele se desgaste durante qualquer percurso no sistema, tornando-o mais agradável. Além disso, estudos acerca do design, cores, estilos de botões, foram práticas de usabilidade que também são levadas em consideração na equipe de projeto o qual participei”***. Entende-se aqui que a estratégia é testar e aperfeiçoar as funções mais utilizadas pelos usuários.

A utilização de práticas isoladas de usabilidade denuncia uma possível lacuna na descrição desses aspectos na documentação, ocasionando a interpretação subjetiva por parte dos desenvolvedores, ou seja, o desenvolvedor presume e decide quais aspectos podem melhorar a utilização e compreensão do usuário. O que certamente causa divergências de compreensão e opinião entre a equipe, onde para um indivíduo uma certa funcionalidade deve ser desenvolvida sob um determinado ponto de vista, para outro indivíduo a mesma funcionalidade deve apresentar outro comportamento.

Quando questionados (Pergunta 10) se a solicitação de *feedback* é uma prática da equipe após a entrega das *Sprints Backlogs*, 78% disseram que sim, pois consideram importante o retorno dado pelos clientes durante essa fase, assim podem aperfeiçoar até a entrega final; já 22% afirmaram que às vezes esse *feedback* é necessário, depende da complexidade do projeto ou da necessidade de aprovação do cliente a cada etapa desenvolvida. Já sobre a satisfação do cliente (Pergunta 11), ao receber o produto, 75% dos entrevistados afirmaram que os clientes sempre estão satisfeitos, já 25% informaram que às vezes os clientes ficam satisfeitos. Essas opiniões são da equipe e não a opinião do cliente em si. Essa insatisfação pode ser ocasionada por problemas de comunicação entre cliente e time, onde o time não entregou exatamente o que o cliente solicitou ou pela não adequação do usuário ao sistema.

Por fim foi questionado (Pergunta 12) se o time acredita que o estudo do usuário é considerado relevante no planejamento do projeto. 56% informaram que sim, 33% às vezes e 11% que não. Esse cenário mostra que é comum a equipe atuar como ‘intérprete de usuários’, ou seja, supõem que seus costumes enviesados contemplarão todas as expectativas dos usuários, dispensando estudos e levantamentos sobre suas necessidades.

Outro ponto identificado é que a equipe ‘*Scrum*’ apresenta divergências de opiniões e atuações, assim o entendimento sobre as necessidades do usuário é falho.

Verificou-se que em várias situações a dinâmica *Scrum* não é completamente utilizada, comprometendo o resultado final do projeto.

6. Considerações Finais

Mesmo com os visíveis benefícios ofertados pela metodologia ágil *Scrum* no desenvolvimento de *software*, percebeu-se que no projeto estudado sua aplicação acontece, predominantemente, de forma isolada, sem interação com os usuários ou técnicas que tornariam o processo de desenvolvimento do *software* mais completo. A usabilidade, por exemplo, possui aspectos essenciais no desenvolvimento de *software* orientado ao usuário, porém pouco disseminado na cultura da equipe. O estudo de caso aplicado mostrou que a usabilidade é empregada de forma parcial, baseando-se nas percepções e definições da equipe desenvolvedora.

Verificou-se também que é prática comum, aos envolvidos no planejamento e execução do projeto, exercerem a função de Design de Interface de Usuário, aplicando conceitos de usabilidade limitados a sua experiência profissional e acreditando que estas são suficientes, considerando assim o estudo do usuário dispensável.

Essa visão distorcida é causa recorrente das alterações e revisões sofridas pelo projeto, tendo funções excluídas/inseridas pela não adaptação de seus usuários. Porém, mesmo com a identificação de problemas de interação entre desenvolvedores-clientes-usuários, o time ágil ainda duvida da importância do levantamento de perfil e necessidades do público alvo. O valor da usabilidade não está internalizado no time.

Pretende-se ainda estender essa pesquisa, através de uma proposta de mestrado, a várias equipes ágeis, em diferentes contextos organizacionais. Para assim, se ter uma análise mais profunda do cenário *Scrum* nas empresas de softwares. Esta pesquisa buscou apenas conhecer um cenário específico, necessitando de estudos mais amplos.

7. Referências

- Bezerra, F. (2014). O que é um Stakeholder?. *Portal Administração – Tudo sobre administração*. <http://www.portal-administracao.com/2014/07/stakeholders-significado-classificacao.html>.
- Carvalho, B.V. and Mello, C.H.P. (2009). Revisão, Análise e Classificação da Literatura sobre o Método de Desenvolvimento de Produtos Ágil Scrum. In *XII Simpósio de Administração da Produção, Logística e Operações Internacionais*. São Paulo, Brasil: SIMPOI.
- Felix, A. de L. C. (2011). *Um Estudo de Caso Sobre Motivação em Integrantes de Equipes de Desenvolvimento de Software no Contexto de uma Organização Pública*. Dissertação de mestrado. UFPE, Universidade Federal de Pernambuco, Recife, Brasil.
- Ferreira, S. B. L. and Leite, J. C. S. do P. (2003). Avaliação da usabilidade em sistemas de informação: o caso do sistema submarino. *Revista de Administração Contemporânea – RAC*, 7(2), 115-137. http://www.anpad.org.br/rac/vol_07/dwn/rac-v7-n2-sbf.pdf.
- Gil, Antonio Carlos. (2008). *Como elaborar projetos de pesquisa* (5th. ed.). São Paulo, Brasil: Atlas.
- Leite, S. F. C. (2013). *Inspeção de Usabilidade Aplicada a Métodos Ágeis: Um estudo de caso*. Monografia de Bacharelado. UFL, Universidade Federal de Lavras, Lavras, Brasil.

- Marconi, M. de A. and Lakatos, E. M. (2004). *Metodologia Científica*. (3th. ed.). São Paulo, Brasil: Atlas.
- Nielsen, J. and Loranger, H. (2007). *Usabilidade da Web*. Rio de Janeiro, Brasil: Elsevier.
- Oliveira, G. R. (2016). *Integração de Práticas de Engenharia de Usabilidade em uma Abordagem Ágil de Desenvolvimento de Software*. Trabalho de conclusão de curso. UFSC, Universidade Federal de Santa Catarina, Florianópolis, Brasil.
- Pressman, R. S. (2004). *Software Engineering – A Practioner’s Approach*. (6th. ed.). Columbus, USA: McGraw-Hill education.
- Siebra, S. A. and Santana, J. F. and Silveira, D. S. (2011). Analisando as Questões de Usabilidade e Acessibilidade do Portal de Periódicos da Capes. *Encontro Nacional de Pesquisa em Ciência da Informação*. Brasília, Brasil.
- Silva, T. S. and Silveira, M. S. (2010). Integrando Avaliação de Usabilidade e Métodos Ágeis. In *V Mostra de Pesquisa da Pós-Graduação*. Rio Grande do Sul, Brasil: PUCRS.
- Soares, M. S. (2004). Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. Universidade Presidente Antônio Carlos. Minas Gerais, Brasil.
- Sommerville, I. (2004). *Software Engineering* (7th. ed.). Harlow, England: Addison-Wesley.
- Sommerville, I. and Sawyer, P. (1997). *Requirements Engineering: A Good Practice Guide*. Nova Jersey, USA: John Wiley & Sons.