

QUALIDADE DE SOFTWARE NA METODOLOGIA ÁGIL

Gabriel Linardi, Nilson Ribeiro Modro
Pós Graduação em Engenharia de Software – PGES
Centro de Educação Superior do Alto Vale do Itajaí – CEAVI
Universidade do Estado de Santa Catarina - UDESC
linardigabriel@gmail.com, nilson.modro@udesc.br

Resumo

Equipes de desenvolvimento que utilizam métodos ágeis podem produzir softwares de modo rápido, mas um fator que deve ser levado em conta é o atendimento aos requisitos de qualidade de software. Após contextualizar as principais metodologias de desenvolvimento ágil, este trabalho abordou os impactos que o desenvolvimento ágil tem sobre a qualidade dos produtos desenvolvidos. Alguns trabalhos correlatos foram estudados e foram analisadas diferentes estratégias que permitem conciliar qualidade de software no paradigma ágil.

Palavras-chave: Qualidade de software, Métodos Ágeis. Projetos de Software.

1. Introdução

Sistemas computacionais têm um papel cada vez mais protagonista em nossas atividades cotidianas. Como Pressman (1995) já mostrava, com o crescimento do setor de TI existe cada vez mais o aumento da demanda por sistemas com qualidade e baixo custo, porém isso aumenta a complexidade inerente ao desenvolvimento de sistemas computacionais.

A Engenharia de Software lida com esta complexidade administrando, de modo conveniente, os recursos disponíveis para construir um produto de software dentro dos prazos e custos estimados (MARCONDES et al., 2004), e como toda área tecnológica e científica, evolutiva, tal evolução da Engenharia de Software pode ser constatada a partir da existência de inúmeros métodos de desenvolvimento de software, que surgiram nas últimas duas décadas e que já são considerados métodos tradicionais e mais complexos em termos de documentação. Tal documentação, embora estabeleça uma sistemática visando obter qualidade no processo, demanda um tempo significativo, principalmente quando a equipe de desenvolvimento é pequena. Nesse caso, o tempo é um importante fator de competitividade dada a crescente necessidade de se executar e entregar sistemas de software com qualidade em prazos cada vez menores (BARCELOSS, 2013).

Desde que Kent Beck introduziu a Programação Extrema, o desenvolvimento ágil de software tem se tornado um tema controverso na engenharia de software. Profissionais e pesquisadores discutem sobre os benefícios disso, outros são rigorosamente contra os métodos ágeis, enquanto outros sugerem uma mistura de métodos ágeis e métodos tradicionais, No entanto, a realidade é que a metodologia ágil ganhou uma enorme aceitação no mercado comercial desde o final dos anos 90, porque deste modo é possível acomodar os requisitos voláteis que surgem durante o projeto, o foco em colaboração entre desenvolvedores e clientes, e apoiar a entrega do produto mais cedo (HUO et al.,2004).

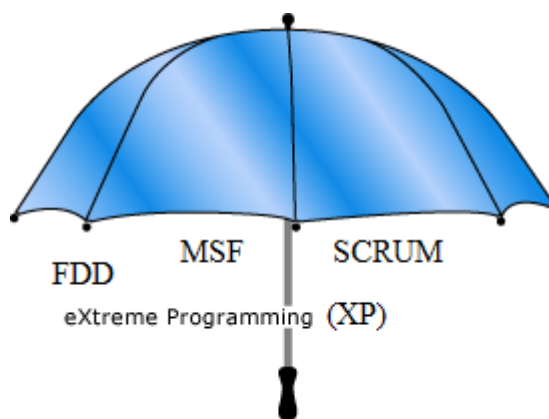
Pensando na qualidade de software utilizando as metodologias ágeis em equipes de desenvolvimento, existem alguns desafios para a utilização deste modelo. O curto ciclo de desenvolvimento apresenta naturalmente, um desafio sob a perspectiva de garantia da qualidade da construção e manutenção do software, pois dificilmente há tempo suficiente para testar tudo. Outro desafio é que equipes ágeis podem não incluir testadores em tempo integral para realizar um regime abrangente de testes. Tais problemas podem ser resolvidos com uma mistura de boas práticas de design e técnicas de testes ágeis, que, quando combinado com um curto ciclo de desenvolvimento, vai resultar em aumento da qualidade do produto, detecção de defeitos cedo, e custo reduzido (CHEN, 2015).

Neste contexto, este trabalho tem como objetivo trazer os fundamentos para que seja possível conciliar qualidade de software na metodologia ágil de um modo efetivo. Para tanto, este artigo está organizado em cinco seções: A segunda seção apresenta as metodologias Ágeis. Em seguida, são apresentados os conceitos básicos de qualidade de software. Já a quarta seção apresenta alguns estudos de situações de como conciliar a metodologia ágil e qualidade. Finalmente, são apresentadas as considerações finais do artigo.

2. Metodologia Ágil

A definição teórica de Metodologia Ágil é dada por (Lindvall et al., 2002), como um grupo de processos de desenvolvimento de software que são iterativo, incremental, auto-organizados e emergentes, conforme figura 2. Alguns dos métodos ágeis mais utilizados são XP (Extreme Programming), SCRUM (processo de desenvolvimento iterativo e incremental), MSF (Microsoft Solutions Framework), FDD (Feature Driven Development) conforme Figura 1.

Figura 1- Guarda-Chuva Ágil

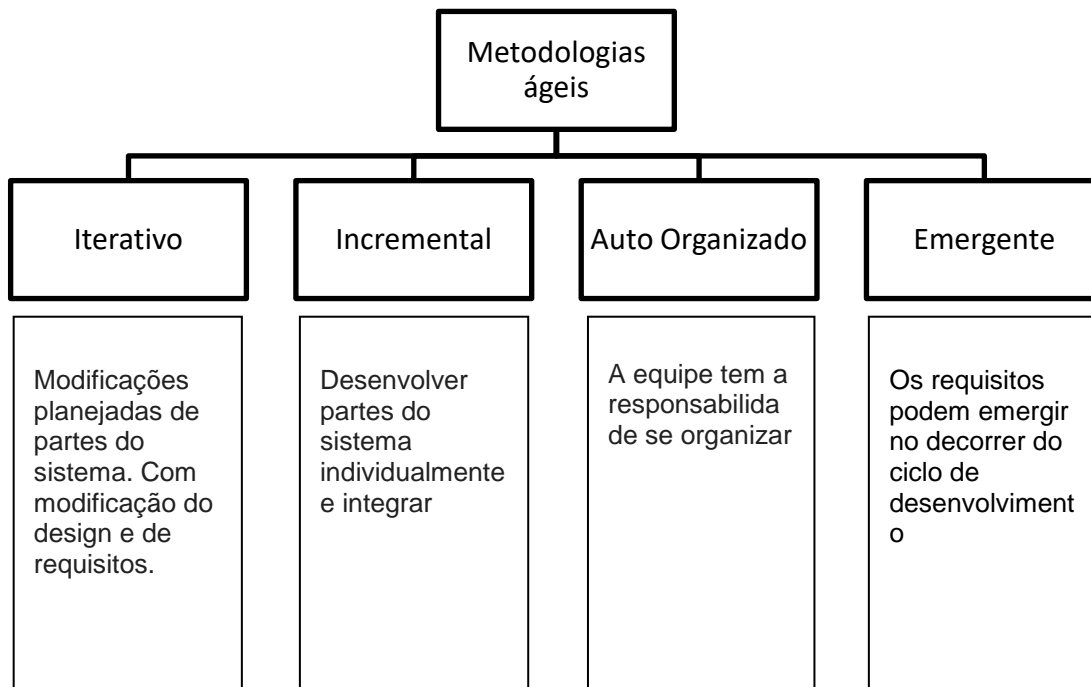


Fonte – (Adaptado de Projeto, 2013)

O Scrum emprega uma estrutura iterativa e incremental da seguinte maneira: no início de cada iteração, a equipe analisa o que deve ser feito e então seleciona aquilo que acreditam poder se tornar um incremento de valor ao produto ao final da iteração. A equipe então faz o seu melhor para realizar o desenvolvimento daquela iteração e ao final apresenta o incremento de

funcionalidade construído para que os stakeholders possam verificar e requisitar alterações no momento apropriado. A metodologia Extreme Programming (XP) enfatiza o desenvolvimento rápido do projeto e visa garantir a satisfação do cliente, além de favorecer o cumprimento das estimativas, já a metodologia MSF(Microsoft Solutions Framework) é uma abordagem adaptável que visa a entrega bem-sucedida de soluções de tecnologia com maior rapidez, número reduzido de pessoas e menor risco, possibilitando resultados de maior qualidade. Encerrando as metodologias mais utilizadas temos o FDD (Feature Driven Development) um método que prega a visibilidade do estado projeto de forma consistente e honesta. É possível saber quantas funcionalidades já foram desenvolvidas e quantas faltam ser desenvolvidas porque tudo é orientado às funcionalidades (LIBARDI; BARBOSA, 2010).

Figura 1- Definição de Metodologias



Fonte – (Adaptado de Ernest Mnkandla, 2008)

As Metodologias Ágeis são uma alternativa sobre as formas tradicionais de desenvolvimento de software. Na implementação de métodos tradicionais, o trabalho começa com o levantamento e documentação de um conjunto "completo" de requisitos, seguido por alto nível de design, desenvolvimento e inspeção. A partir de meados da Década de 1990, alguns pesquisadores acharam estes passos de desenvolvimento iniciais frustrantes e talvez, impossíveis (HIGHSMITH, 2002). Highsmith (2002), mostra ainda que a indústria e tecnologia se move muito rápido, os requisitos dos clientes alteram muito no decorrer do projeto, complicando o desenvolvimento quando utilizado os métodos tradicionais e os clientes tornaram-se cada vez mais incapaz de afirmar definitivamente as suas necessidades e, ao mesmo tempo, esperam mais do seu software.

A metodologia de desenvolvimento ágil oferece oportunidades para avaliar o andamento de um projeto durante todo o ciclo de desenvolvimento. Isto é conseguido através de iterações,

no final dos quais as equipes devem apresentar um incremento de produto potencialmente utilizável. Centrando-se sobre a repetição de ciclos de trabalhos, bem como o produto funcional que produzam, metodologia ágil é descrita como "iterativo" e "incremental." Utilizando a metodologia tradicional como cascata, as equipes de desenvolvimento só tem uma chance de obter cada aspecto de um projeto. Em um paradigma ágil, cada aspecto do desenvolvimento, requisitos, design etc, são continuamente revisto ao longo do ciclo de vida do projeto. Quando uma equipe para e reavalia a direção de um projeto a cada duas semanas, há sempre tempo para conduzi-la em outra direção.

Geralmente, esses métodos dividem o desenvolvimento em diversas iterações de ciclos mais curtos (no caso do Scrum estes ciclos são chamados de Sprints) e realizam entregas ao final de cada uma delas, de forma que o cliente (interno ou externo) receba uma versão que agregue valor ao seu negócio (DANTAS, 2003). Assim, as mudanças de requisitos podem ser acompanhadas pelos desenvolvedores no início de cada ciclo. E existe uma retroalimentação por parte do cliente para a equipe de desenvolvimento, o que reduz o risco do projeto.

Os métodos tradicionais de desenvolvimento têm o foco na geração de documentação sobre o projeto e no cumprimento rígido de processos. Já os métodos ágeis concentram as atenções na constante entrega do produto (MUNDIM et al., 2002) e nas interações entre os indivíduos. Nestes métodos, é minimizada a fase de planejamento inicial, de modo que os desenvolvedores se concentram em entregar o produto ao fim de cada iteração, ao invés de traçar diretrizes e planejamentos para o projeto como um todo.

Como as equipes podem desenvolver software ao mesmo tempo em que eles realizam a coleta de requisitos, o fenômeno conhecido como "paralisia da análise" é menos susceptível de travar uma equipe de progredir. E como o ciclo de trabalho de uma equipe é limitada a duas semanas, ele dá as partes interessadas oportunidades recorrentes para calibrar lançamentos para o sucesso no mundo real. Deste modo, a metodologia de desenvolvimento ágil ajuda as empresas a construir o produto mais adequado, capacitando equipes para replanejar continuamente e agregando competitividade, podendo preservar a relevância crítica do mercado de um produto e assegura o trabalho de um time não acabar em uma prateleira, nunca lançado, (METHODOLOGY, 2008).

3. Qualidade de Software

A qualidade sempre foi um conceito inerente à produtos e serviços, porém em relação a qualidade de software, até pouco tempo, o usuário / público em geral não conseguia fazer uma clara distinção entre hardware e software. Em pouco tempo, pode se perceber uma crescente preocupação pública sobre a difusão do software, não só nos serviços mas também em produtos de consumo como automóveis, máquinas de lavar, telefones e máquinas de barbear. Consequentemente, os profissionais de software precisavam se preocupar com a qualidade de todos esses produtos - desde sistemas grandes e complexos, até para pequenos softwares embarcados (PFLEEGER; KITCHENHAM, 1996).

Neste sentido, surge a seguinte questão, como deve se avaliar "adequadamente" a qualidade em um produto de software? O contexto é importante. Erros tolerados no software de processamento de texto de simples podem não ser aceitáveis no controle software para uma usina de energia nuclear. Assim, deve-se reexaminar o significado de "segurança crítica" e "Missão-crítica", bem como a contribuição de software para melhor funcionalidade e qualidade de produtos e empresas. Ao mesmo tempo, é necessário perguntar quem é responsável pela

definição de qualidade, das metas e ter certeza de que eles estão sendo realizadas de acordo (PFLEEGER; KITCHENHAM, 1996).

Na medida em que cresce a demanda por sistemas complexos, onde os mesmos possuem grande responsabilidade no contexto das organizações, a qualidade é um fator essencial no desenvolvimento de software. Sendo assim, cada vez mais, há uma necessidade para se investir em qualidade. Contudo, uma das primeiras dificuldades encontradas na definição e implantação de um programa de qualidade está em compreender o que, de fato, significa qualidade de software.

Para auxiliar os profissionais da área de qualidade existe a ISO (25030:2008) - Square responsável pela especificação e avaliação da qualidade de produto de software, propondo Atributos de Qualidade, distribuídos em seis características principais, com cada uma delas divididas em sub-características: Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade. A ISO (25030:2008) - Square fornece requisitos e recomendações para a especificação de requisitos de qualidade de software (COLOMBO, 2013).

No contexto de desenvolvimento de software, a qualidade pode ser entendida como um conjunto de características a serem atendidas em um determinado grau, de modo que o software atenda os requisitos explícitos e implícitos de seus usuários (ROCHA et al., 1994). Porém, não se obtém qualidade do produto de forma simples. Ela tem de ser construída. Assim, a qualidade do produto depende fortemente da qualidade de seu processo de desenvolvimento.

No processo de qualidade de software é necessário que se tenham testes dos produtos desenvolvidos para que o processo de qualidade e entrega do produto fiquem de acordo com os requisitos do cliente, claro que utilizando a atividade de testes por si só, não é capaz de tornar o software livre defeitos. Contudo, quando formalmente feita pode reduzir o número de não conformidades e ainda fornecer dados importantes para uma estimativa da confiabilidade do software. Sua implementação exige, além da capacitação técnica da equipe de desenvolvimento para torná-la capaz de gerar casos de testes consistentes e de alta probabilidade de evidenciar erros, uma estrutura organizacional que permita incorporar a atividade de testes às atividades do ciclo de desenvolvimento e a participação de pessoas isentas na execução dos testes.

Para Chappell (2013) a qualidade de software possui três aspectos principais a serem seguidos, são eles: qualidade funcional, qualidade estrutural e processos de qualidade, Chappell (2013) cita que existem muitas ligações entre estes três aspectos de qualidade de software. Por exemplo, melhorar o processo de qualidade com métodos ágeis de desenvolvimento aumenta as chances de conseguir os requisitos do projeto, que também melhora a qualidade funcional do produto, segundo Chappell (2013) fornecer igual ênfase na qualidade funcional, qualidade estrutural e qualidade do processo, podem ampliar a visão de um projeto para incluir as necessidades que são importantes para todas as três partes interessadas: os usuários, as equipes de desenvolvimento e Patrocinadores dos produtos. Para avaliar a qualidade, é preciso existir formas de medi-la. Ou seja, é preciso obter uma medida que possa calcular o grau de alcance de uma característica de qualidade. Deste modo, para computar uma característica de qualidade, é necessário estabelecer uma métrica capaz de calcular e fazer uma medição para determinar a medida, resultado da aplicação da métrica escolhida.

4. Como Conciliar Métodos Ágeis e Qualidade de Software.

O desenvolvimento de software tradicional, que muitas vezes utiliza o modelo cascata, emprega um processo sequencial que consiste tipicamente de definição de requisitos, especificação,

concepção, construção, testes unitários, testes de integração e testes do sistema. Onde a equipe de trabalho e usuários do sistema pode ter uma idéia do produto apenas no final deste processo, isso pode trazer muitos problemas de rendimento do projeto, tempo, elevação de custos e oportunidades de mercado. Por outro lado, temos a metodologia ágil um modelo de desenvolvimento de software altamente iterativo onde cria pequenos pedaços de software, que em consequência consegue eliminar os problemas mencionados acima convertendo isso em qualidade do produto.

De acordo com os estudos apresentados a seguir, realizadas por profissionais da área para que se fosse possível achar um modo de unir as Metodologias ágeis e Qualidade de software. Como poderá ser visto nesta seção, algumas maneiras para realizar essa conciliação foram encontradas em estudos de casos e pesquisas por diferentes autores, nesses estudos serão citados os pontos fortes e fracos para a implantação destas metodologias.

São muitas razões para que as equipes de qualidade não consigam acompanhar as equipes de desenvolvimento quando utilizada as metodologias ágeis, essas dificuldades podem ser vistas no cotidiano de muitas empresas e equipes de desenvolvimento: em quase todos os artigos ou revistas sobre qualidade descrevem-se que todos os membros das equipes são responsáveis pela qualidade do projeto, não apenas os analistas de qualidade. Em especial, a qualidade do próprio processo de desenvolvimento adotado contribui para a qualidade do produto final. Porém, muitas vezes não existe uma definição clara de papéis e responsabilidades para cada pessoa no time a respeito da qualidade do produto, que atividades e artefatos e ferramentas estão sob sua responsabilidade e como compartilham os mesmos (NEGRELLO, 2013).

Mas a resistência da utilização das metodologias ágeis existem por diversos fatores, primeiramente porque ainda existe empresas que esperam os produtos ficarem prontos para posteriormente iniciar o processo de qualidade. E também pelo fato das aplicações atuais serem muito mais complexas, onde as mesmas fazem muitas integrações com softwares distintos ao projeto, por este motivo criar um ambiente de testes ficou muito mais custoso e complexo, não apenas a necessidade de investimentos de maquinas e softwares, mas também de pessoas especializadas na área de qualidade e testes e com conhecimento nas varias aplicações envolvidas com autonomia para validar os softwares.

Para driblar estes pontos negativos e desafiadores na utilização de metodologias ágeis. A maioria das equipes utiliza o Scrum utilizando Sprint ou as iterações e ao final de cada um desses ciclos realizam a qualidade do produto desenvolvido. Porém isso foge dos princípios e definições de como utilizar a metodologia ágil. O Sprint pode ser considerada o coração do Scrum, é durante sua existência que o produto é implementado pela equipe de desenvolvimento, e no final dela é entregue o incremento utilizável do produto. Cada Sprint possui uma definição do que deve ser entregue, junto com o prazo, que geralmente varia entre 2 a 4 semanas, não podendo passar disso. Caso haja alteração no escopo, é necessário que o Time de Desenvolvimento e o ProductOwner se reúnam para negociarem prazos e alterações na Sprint. (BONFIM, 2014).

Para que estes problemas de como realizar qualidade e quando será realizada não será respondido de uma forma tão simples e também não será resolvido com apenas uma solução, pois será necessário utilizar componentes básicos para um projeto: processos e ferramentas (NEGRELLO, 2013). O primeiro ponto a ser verificado é realizar o compartilhamento dos objetivos e informações dos projetos com ambas as equipes sejam elas de Qualidade ou de Desenvolvimento, também é necessário sincronizar estas equipes para que elas não atrapalhem o serviço uma da outra.

Para entender melhor como essas técnicas podem ser aplicadas em conjunto nas empresas, alguns estudos de caso e artigos foram realizados com entrevistas e questionários aplicados em empresas por profissionais da área. Um desses trabalhos foi realizado por (OLIVEIRA, 2013), que foi motivado por um estudo sobre qualidade de software e processos de desenvolvimento. No qual definiu um cronograma de um mês entre maio e junho de 2013 para

realizar as pesquisas e aplicação de questionários, esses questionários foram enviadas para quatro empresas conhecidas por trabalharem com métodos ágeis, estes questionários também foram publicados em redes sociais com foco em métodos ágeis.

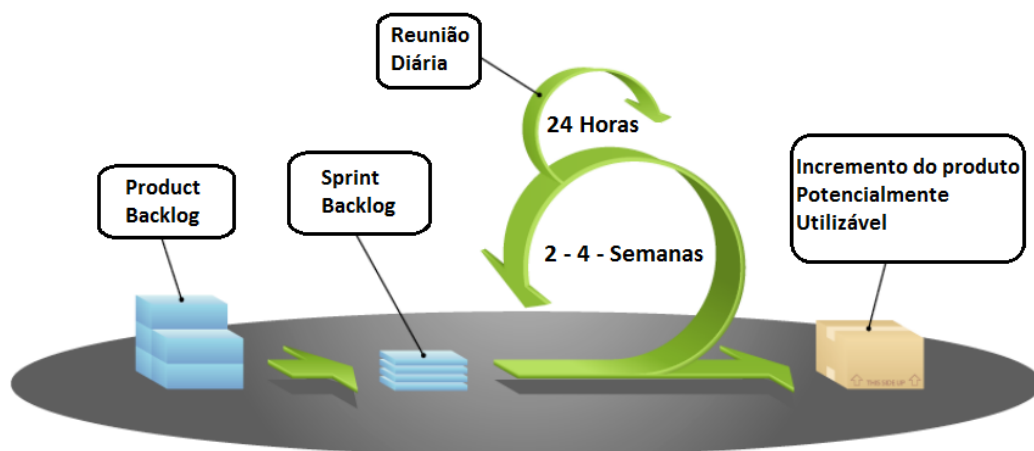
Este questionário foi realizado em cinco etapas, onde a primeira parte é focada em conhecer e classificar os participantes. Já na segunda parte abordou o processo de desenvolvimento, para identificar os métodos utilizados pelo profissional, as terceira e quarta etapa foram abordadas especificamente atividades de qualidade e teste de software. A ultima etapa foram questões abertas sobre o método de desenvolvimento utilizado e qualidade de software. O questionário foi respondido por 28 profissionais de 6 empresa distintas. Empresas na sua maioria de pequeno porte.

Os resultados obtidos pela pesquisa de (OLIVEIRA, 2013), foram conforme a suspeita inicial, onde existe uma identificação muito grande entre os desenvolvedores e métodos ágeis. As maiorias dos participantes mostraram conhecimento no processo utilizado e identificaram em suas respostas a grande importância de atividade de garantia de qualidade e em especial testes de software. Apesar da pouca quantidade de documentação foi verificado que as equipes possuem meios de comunicações muito eficientes e as informações do projeto são repassadas entre a equipe de forma natural.

Desenvolvedores acreditam que a utilização de metodologias ágeis é uma das melhores formas de criar um produto de qualidade. A utilização de metodologias ágeis muda a postura dos agentes do processo e indica que pode ser uma das principais razões pela qual os métodos ágeis criam projetos de software com alta qualidade (OLIVEIRA, 2013).

Outro estudo foi realizado por (VIJAY; GANAPATHY, 2014) em um dos módulos de um projeto de uma empresa de software que utilizava a metodologia ágil por mais de 3 anos. o sistema estudado e um sistema de serviços de viagens. A equipe do estudo era formada de uma forma diferente das tradicionais foi montado utilizando o modelo Onsite-Offshore, era composta por 16 pessoas 3 pessoas alocadas na empresa e 13 trabalhando externamente onde a mesma seguiu o ciclo scrum para organização das atividades conforme figura abaixo.

Figura 2- Ciclo Scrum



Fonte:(Adaptado de Vijay; Ganapathy, 2014)

Estes ciclos trabalhavam com um período de 2 semanas. Todos os requisitos eram estudados na reunião de planejamento do sprint que era prevista para ser realizada antes de cada sprint. Somente os membros da equipe alocada na empresa faziam parte da reunião de planejamento do sprint na reunião o Product Owner do produto prioriza os itens de sprint e a equipe externa fornece estimativas para cada história.

Uma vez que o sprintbacklog está pronto, a equipe começa a trabalhar nos itens de sprint como Design, desenvolvimento e atividades de qualidade Todos os dias reuniões diárias são agendada entre os membros da equipe no local e externa através de telefone e discutem o que tem sido feito, o que será feito e os bloqueadores. esta reunião tem a duração de 15 a 30 minutos. No fim de cada iteração do sprint é repassado aos clientes para feedback e também reuniões de retrospectiva estão agendadas para discutir o que ocorreu bem dentro do Sprint e o que não ocorreu bem as lições aprendidas e melhorias a serem realizadas.

Utilizando este método de desenvolvimento algumas situações negativas foram encontradas podendo perceber claramente que o método utilizado não trouxe a qualidade no projeto conforme era esperado, alguns destes problemas podem ser vistos na tabela abaixo.

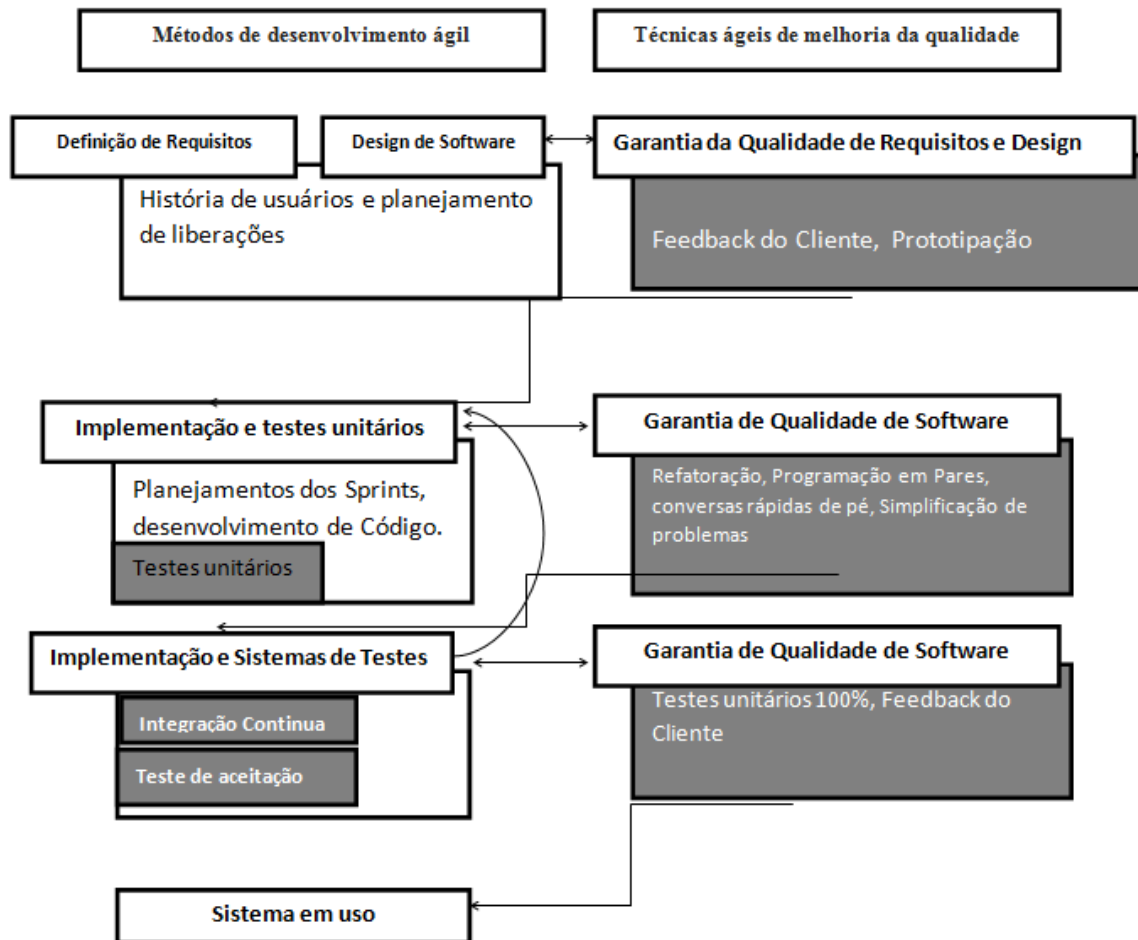
Tabela 1 – Pontos Negativos encontrados.

<i>Pontos Negativos encontrados</i>
A equipe externa ficou sobrecarregada e não conseguiu entregar o planejado.
Os membros das equipes precisaram esticar os sprint e horas de desenvolvimento, isso resultou na impossibilidade de entregar os produtos com qualidade.
O custo com qualidade ficou muito alto.
Utilizando este método, mais defeitos foram encontrados na ultima iteração.
Foi produzida uma quantidade reduzida de novas funcionalidades.
O produto final não foi entregue de acordo com as necessidades reais do cliente.
Atrasos na entrega do produto.
Ocorreu perda de cliente.

Um terceiro estudo de caso feito por (HOSSAIN; KASHEM; SULTANA, 2013) onde foi estudado algumas técnicas ágeis para a melhoria da qualidade de software, neste estudo foram considerados 14 fatores de qualidade alguns deles como: Exatidão, Medida em que o software está em conformidade com as suas especificações; verificabilidade, Verificar os recursos de software e desempenho com base em seus objetivos declarados; Eficiência, Medida em que o software é capaz de fazer mais com menos; Integridade, quando software é capaz de resistir a intrusão de usuários não autorizados ou software dentro de um período de tempo especificado, essas são alguns dos fatores estudados.

O estudo foi centralizado na figura 3, onde do lado esquerdo é exibido a sequência principal do processo ágil e do lado direito é exibido algumas técnicas ágeis que melhoram a capacidade de controle de qualidade unida a cada ciclo de desenvolvimento ágil.

Figura 3 - Técnicas ágeis e garantia de qualidade



Fonte:(Adaptado de HOSSAIN; KASHEM; SULTANA, 2013)

Segundo Hossain, Kashem e Sultana (2013) este método foi utilizado para melhor comunicação de problemas, soluções e promover foco da equipe. Dentro deste ciclo umas das técnicas utilizadas é o Stand-up-Meeting todos da equipe se levantavam em um círculo para evitar longas discussões. Durante essas rápidas conversas desenvolvedores relatam pelo menos três coisas; o que foi realizado ontem, o que vai ser realizado hoje, e os problemas que estão causando atrasos. Essa conversa rápida de pé pode melhorar os fatores de qualidade de software, tais como confiabilidade e flexibilidade do projeto que esta sendo desenvolvido. Depois dos estudos realizados, Hossain, Kashem e Sultana (2013) chegaram a conclusão que as metodologias ágeis e qualidade do software podem sim trabalhar em conjunto, bem como melhorar a qualidade do software durante todo o desenvolvimento do software, pois os defeitos encontrados e melhorias solicitadas pelo cliente podem ser realizadas durante o projeto sem prejudicar a qualidade do produto, também comentam que trabalhando desta forma, a satisfação do cliente em termos de tempo, orçamento, usabilidade, exigência e de segurança utilizável é muito maior.

Encerrando os estudos de caso, temos o trabalho realizado por (JINZENJI et al., 2013) onde trabalharam em cima da hipótese de que o desenvolvimento ágil não produz qualidade inferior de que quando utilizado a metodologia sequencial, O estudo foi realizado com o objetivo de examinar essa hipótese e detectar os fatores e condições para que a hipótese fosse confirmada, para isso estudos foram realizados utilizando testes unitários, integração, sistema e regressão em três projetos, onde utilizam metodologias de desenvolvimento ágeis e tradicionais. A conclusão deste estudo foi, que pode se ter mais maturidade e qualidade utilizando a metodologia ágil do que o desenvolvimento sequencial, quando a qualidade e testes são feitos adequadamente e suficientemente.

Após analisar os principais estudos de caso na utilização de metodologias ágeis e qualidade de software, foi obtida uma tabela com os resultados destes trabalhos. Pode se perceber que dependendo do método de trabalho utilizado as equipes de desenvolvimento conseguem conciliar desenvolvimento ágil com qualidade.

Figura 4 – Tabela de estudos de caso.

Autores	Ano	Método(s) Agil(ei)	Ferramentas da qualidade	Resultado do Estudo
Bruno Henrique Oliveira	2014	Não especificado	Não especificado	Utilizou pesquisas e aplicações de questionários nas redes sociais e enviado para 4 empresas. Os praticantes de métodos ágeis acreditam que o processo utilizado é uma das melhores formas de criar um produto de qualidade. A utilização de metodologias ágeis muda a postura dos agentes do processo e indica que pode ser uma das principais razões pela qual os métodos ágeis criam projetos de software com alta qualidade.
Deepa vijay e Gopinath GANAPATHY	2014	Scrum	Não especificado	Realizado um estudo dentro de uma empresa utilizando a metodologia ágil em uma equipe Onsite-Offshore, onde o estudo apontou muitos problemas ocorrem utilizando as metodologias ágeis(Scrum) neste tipo de
Amran Hossain, Dr. Md. Abul Kashem, Sahelee Sultana	2013	Não especificado	Qualidade de Design, Qualidade de Desempenho, Qualidade e de Adaptação	Utilizaram 14 técnicas de qualidade de software juntamente com o ciclo principal da metodologia ágil unindo com algumas técnicas ágeis que melhoram a capacidade de controle de qualidade unidas a cada ciclo de desenvolvimento ágil.
Kumi Jinzenji, Takashi Hoshino, Laurie Williams, Kenji Takahashi	2013	Não especificado	Testes unitários, testes de integração, testes de regressão, testes de sistema	Conseguiram provar a hipótese de que sistemas que utilizam o desenvolvimento ágil podem sim ter a mesma qualidade ou maior de que sistemas que utilizam metodologias tradicionais, se a qualidade e testes foram feitas de maneira correta.

Fonte: elaborada pelo próprio autor

5. Considerações Finais.

Este trabalho teve como principal objetivo estudar e analisar o tema qualidade de software no contexto ágil. Métodos ágeis estão a cada dia sendo mais empregados nas empresas de pequeno e médio porte, porém, pode-se observar que a forma de avaliar a qualidade nesse tipo de processo não pode ser o mesmo empregado em outras metodologias tradicionais.

Este estudo foi realizado para tentar entender um pouco mais a diferença entre a utilização dos métodos mais tradicionais de desenvolvimento e as metodologias ágeis com atividades de controle e garantia de qualidade, e pode ser observado que pode sim dar certo,

dependo do contexto e formato que as equipes realizam essa conciliação entre desenvolvimento e qualidade.

Baseado nos estudos realizados pelos autores citados na seção anterior, percebeu-se que ainda existe uma dificuldade no processo de qualidade de software dentro de um projeto ágil, uma vez que as entregas são realizadas em um curto período de tempo, porém não é impossível a implantação destas técnicas. O conjunto de problemas da qualidade realizados por todos os integrantes da equipe também é um fator relevante, pois a visão dos analistas de qualidade é muito mais criteriosa comparada a análise de qualidade que é realizada por outro membro da equipe de desenvolvimento.

Através das atividades realizadas pela equipe de qualidade podem-se dar condições ao gerente do projeto de manter os artefatos produzidos sob controle, pois eles estão sempre sendo avaliados no nível de atendimento aos requisitos funcionais e de qualidade, mantendo em alta a confiabilidade da equipe em relação ao projeto e ao seu prazo de entrega, já que os erros vão sendo identificados durante todo o ciclo de desenvolvimento. Esse é um dos fatores fundamentais para o sucesso do projeto.

Após a análise de 4 estudos de casos citados a cima referente a aplicação de metodologias ágeis e garantia de qualidade, pode se perceber que muito depende do modelo em que as metodologias são aplicadas e também no formato de trabalho das equipes, como pode ser notado, em equipes que trabalham distantes uma das outras, não se teve sucesso, pois pode se perceber que a conversa diária referente a problemas e sucessos do projetos são muito importantes, porem para equipes que trabalham em conjunto e ambas tem comunicação no dia a dia do projeto, a qualidade do produto se torna um requisito de extrema importância, onde todos zelam para que tudo de certo.

Referências

CHAPPELL, David. **THE THREE ASPECTS OF SOFTWARE QUALITY: FUNCTIONAL, STRUCTURAL, AND PROCESS**. 2013. Disponível em: <http://www.davidchappell.com/writing/white_papers/The_Three_Aspects_of_Software_Quality_v1.0-Chappell.pdf>. Acesso em: 25 set. 2016.

CHEN, Frank. **How Do Agile Rapid Iterations Improve Software Quality?** 2015. Disponível em: <<https://www.linkedin.com/pulse/how-do-agile-rapid-iterations-improve-software-quality-frank-chen>>. Acesso em: 07 ago. 2016.

DANTAS, V. F. **Uma metodologia para o desenvolvimento de aplicações Web num cenário global**. 2003. Dissertação (Mestrado)-Centro de Ciências e Tecnologia, Universidade Federal de Campina Grande, Campina Grande, 2003

DYBA, T.; DINGSOYR, T. **Empirical studies of agile software development: a system atic review**. *Information and Software Technology*, v. 50, n. 9-10, p. 833-859, 2008.

HIGHSMITH, J., **Agile Software Development Ecosystems**, Boston, MA: Addison Wesley, 2002.

HOSSAIN, Amran; KASHEM, Dr. Md. Abul; SULTANA, Sahelee. **Enhancing Software Quality Using Agile Techniques**. 2013. 7 f. Tese (Doutorado) - Curso de Computer Science And Engineering Department, Dhaka University Of Engineering & Technology, Bangladesh, 2013.

HUO, Ming; VERNER, June; ZHU, Liming; BABAR, Muhammad. **Software Quality and Agile Methods.**, 2004.

JINZENJI, Kumi et al. **Empirical study of Software Quality Evaluation in Agile Methodology Using Traditional Metrics.** Tokyo: ., 2013.

LINDVALL, Mikael et al. **Empirical Findings in Agile Methods.** Londres: , 2002.

MARCONDES, F. S. et al. **Agilidade em desenvolvimento de Software.** Anais do 6º Simpósio de Iniciação Científica e Tecnológica da FATEC-SP, SP. 2004.

METHODOLOGY, Agile. **The Agile Movement.** 2008. Disponível em: <<http://agilemethodology.org/>>. Acesso em: 10 set. 2016.

MNKANDLA, Ernest. **A SELECTION FRAMEWORK FOR AGILE METHODOLOGY PRACTICES:: A Family of Methodologies Approach.** 2008. 247 f. Tese (Doutorado) - Curso de ., Faculty Of Engineering And The Built Environment, Johannesburg, 2008.

MUNDIM, A. P. F. et al. **Aplicando o cenário dedesenvolvimento de produtos em um caso prático decapacitação profissional.** *Gestão & Produção*, v. 9,n. 1, p. 1-16, 2002.

NEGRELLO, Ana. **Métodos Ágeis e Qualidade: Como Conciliar?** 2013. Disponível em:<https://www.ibm.com/developerworks/community/blogs/rationalbrasil/entry/m_c3_a9todos__c3_a1geis_e_qualidade_como_conciliar2?lang=en>. Acesso em: 03 jun. 2016.

OLIVEIRA,B,H. **Qualidade de software no desenvolvimento com métodos ágeis,** 2014.85 f. Dissertação (Mestrado em Ciências da Computação e Matemática) Universidade de São Paulo, São Carlos, 2014.

PFLEEGER, Shari Lawrence; KITCHENHAM, Barbara. **Software Quality: The elusive target.** :lee Software, 1996.

PRESSMAN, R. S., 1995. **Engenharia de Software.** Makron Books. São Paulo. Brasil.

ROCHA, A.Ret al. **Uma experiência na definição do processo de desenvolvimento e avaliação de software segundo as Normas ISO.** Curitiba:, 1994. 93 p

SOFTWARE, **Manifesto For Agile. Manifesto for Agile Software Development. Web Site.** Disponível em: <[Http://agilemanifesto.org](http://agilemanifesto.org)>. Acesso em: 05 jan. 2016.

SCHWABER, Ken; BEEDLE, Mike. **Agile Software DevelopmentwithScrum.** Saddle River: Prentice Hall, 2001. 158 p.

VIJAY, Deepa; GANAPATHY, Gopinath. **GUIDELINES TO MINIMIZE THE COST OF SOFTWARE QUALITY IN AGILE SCRUM PROCESS.** 2014. 9 f., DepartmentOf Computer Science &Engineering, BharathidasanUniversity, India, 2014.

LIBARDI, Paula L.o.; BARBOSA, Vladimir. **Métodos Ágeis.** 2010. 35 f. Monografia (Especialização) - Curso de Faculdade de Tecnologia, Unicamp, Limeira, 2010.

COLOMBO, Regina Maria Thienne. **Proposta de uma metodologia de medição e priorização de segurança de acesso para aplicações web.** 2013. 233 f. Tese (Doutorado) - Curso de., Usp, São Paulo, 2013.

PROJETO, Universo. **Os 12 princípios do Manifesto Ágil.** 2013. Disponível em: <<https://universoprojeto.wordpress.com/tag/guarda-chuva-agil/>>. Acesso em: 05 nov. 2016.