

Aplicando a Metodologia Ágil Scrum para apoio ao Gerenciamento de Requisitos

Felipe Luiz Carnevali, Daniel Lucrédio

Resumo: Este trabalho apresenta um estudo de caso que aplica a metodologia ágil Scrum para o gerenciamento de requisitos. São levantados os dados das quais a empresa estudada usa para realizar o seu próprio gerenciamento de requisitos. Baseando nas práticas e regras do Scrum foi proposta uma implementação que se adeque à realidade da empresa, alterando a sua forma estrutural nas responsabilidades dos profissionais e também modificando o ambiente de trabalho aplicando as técnicas citadas neste estudo de caso, focando na qualidade, prazos, custo, agilidade e satisfação do cliente.

Palavras-Chave: Scrum, metodologia ágil, requisitos.

Applying Scrum agile methodology to support the requirements Management

Abstract: This paper presents a case study that applies Scrum agile methodology for requirements management. Data used by the studied company to carry out its own management requirements is raised. Based on Scrum practices and rules, an implementation that fits the reality of the company was proposed, changing its structural form in the responsibilities of professionals and also changing the working environment by applying the techniques mentioned in this case study, focusing on quality, deadlines, cost, agility and customer satisfaction.

Keywords: Scrum, Agile methodology, requirements.

I. INTRODUÇÃO

Diversos estudos (LEFFINGWELL et al., 2003); (SOMMERVILLE; SAWYER, 1997) levantaram e comprovaram que uma grande quantidade de projetos de software são cancelados ou simplesmente fracassam por não atenderem totalmente às necessidades dos clientes ou excederem prazos e orçamentos propostos no contrato. Esses problemas geralmente acontecem devido à constante mudança do mercado, o que faz com que os requisitos mudem frequentemente. Apesar das mudanças, há grande deficiência no levantamento dos requisitos, um dos principais contratempos no fracasso dos projetos de software.

Com essas análises alguns estudiosos passaram a considerar a engenharia de requisitos “incluindo todo o processo de elicitação (coleta de requisitos), especificação, viabilidade e validação” como um fator muito importante dentro da área de engenharia de software. Todavia a engenharia de software já vem sendo estudada no mundo acadêmico há mais de 40 anos (ESPINDOLA et al., 2004), e muitos destes sistemas vem sendo utilizados até hoje.

A engenharia de requisitos é uma forte candidata para ocupar-se de responsabilidades com relação à garantia, necessidades, qualidade, prazos e orçamentos que são estimados aos clientes, bem como lidar com as dificuldades que surgem no dia à dia do desenvolvimento de sistemas.

Porém esse processo costuma ser muito rigoroso e demorado. Há algumas décadas criaram-se algumas metodologias que propõem melhorar o andamento dos projetos, as quais são chamadas de metodologias ágeis. Entre elas existem várias metodologias tais como Extreme Programming (XP), Feature Driven Development (FDD), Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Scrum, Crystal Methods (CM) entre outras. Abrahamsson et al. (2002) confirma que quanto mais de uma metodologia ágil é utilizada ao mesmo tempo, melhora-se o processo de agilidade. Entretanto, de acordo com AGILE SURVEY (2012), duas das mais populares metodologias que são utilizadas no contexto de metodologias ágeis são Extreme Programming (XP) e o Scrum. Por esse motivo, decidiu-se focar o estudo de caso em uma dessas, que é a metodologia ágil Scrum.

O Scrum foi criado por Schwaber em 1995, com foco no processo de gestão de projetos de software complexos. Conforme (SCHWABER; BEEDLE, 2002), sua teoria é atender as necessidades do negócio baseando-se em sistemas adaptativos. Schwaber e Beedle (2002) afirmam ainda que a principal característica do Scrum é a comunicação entre as equipes que participam direta ou indiretamente e não há especificações referentes a práticas de construção, pois o método enfatiza que as equipes sejam auto-organizadas e os

processos são orientados a objetivos e resultados.

Este trabalho apresenta uma análise de como funciona a engenharia de requisitos, considerando seu processo básico, e as práticas e regras da metodologia ágil Scrum. O objetivo desse estudo é propor uma melhoria no processo de engenharia de requisitos no contexto de uma empresa de software localizada na cidade de Ribeirão Preto (SP), que possui um sistema já consolidado no mercado há mais de 20 anos com uma cartela de clientes em todo o país. Seus processos de organização de levantamento de requisitos foram estudados e foi proposta uma melhoria com base nas técnicas, práticas e regras acadêmicas citadas neste trabalho.

A seção II apresenta a composição da engenharia de requisitos e na seção III a metodologia ágil Scrum. A seção IV apresenta o estudo de caso utilizado neste trabalho, detalhando o contexto da empresa estudada, e a proposta realizada para a melhoria de desempenho, utilizando-se dos papéis e práticas do Scrum com os resultados preliminares e discussões de maior importância para o resultado esperado. A seção V apresenta os trabalhos relacionados e as conclusões e trabalhos futuros são descritos na seção VI.

II. ENGENHARIA DE REQUISITOS

Com a grande demanda e crescente complexidade de software requerida, à ausência de qualidade começa a ser percebida, pois os produtos não conseguem prover a capacidade necessária. Atualmente é utilizada a engenharia de requisitos para sistematizar o processo de definição de software. Essa sistematização é necessária, pois ajuda os engenheiros de requisitos a entenderem melhor o problema.

Uma boa definição de requisitos é dada por como sendo a "condição necessária para a obtenção de certo objetivo, ou

para o preenchimento de certo objetivo". Leite (1994).

A engenharia de requisitos foca no processo e se torna responsável pelas evidências que mais podem causar problemas sérios, podendo ser destacadas as mudanças de requisitos e especificações, requisitos e especificações incompletos e falta de engajamento do usuário (LEFFINGWELL et al., 2000).

Os usuários esperam um sistema que atenda as suas necessidades, tarefas e facilite o seu trabalho diário. Os financiadores aguardam um sistema que obtenha um retorno adequado ao investimento fornecido. A equipe de desenvolvimento cria expectativas em desenvolver um sistema com qualidade, prazo e custo estipulado atendendo todas as reais necessidades dos usuários (LEFFINGWELL et al., 2000), porém a realidade quase sempre não atende a esses quesitos, devido à grande abrangência em que os softwares vêm fazendo na sociedade.

A engenharia de requisitos é uma atividade responsável por criar e manter documentos de especificações de requisitos de sistemas chamados DERS segundo (KOTONYA; SOMMERVILLE, 1998). Este processo como um todo é composto por quatro grandes processos: o estudo de viabilidade, em qual aspecto o sistema será viável ao negócio a ser utilizado, a elicitação de requisitos, com a descoberta e levantamento dos requisitos, a especificação, onde a organização dos requisitos é apresentada em um padrão e a validação, com a descoberta e validação dos requisitos que foram registrados e definem a necessidade que o usuário deseja para o sistema.

Na figura 1 é possível visualizar o modelo de como funciona o processo da engenharia de requisitos:

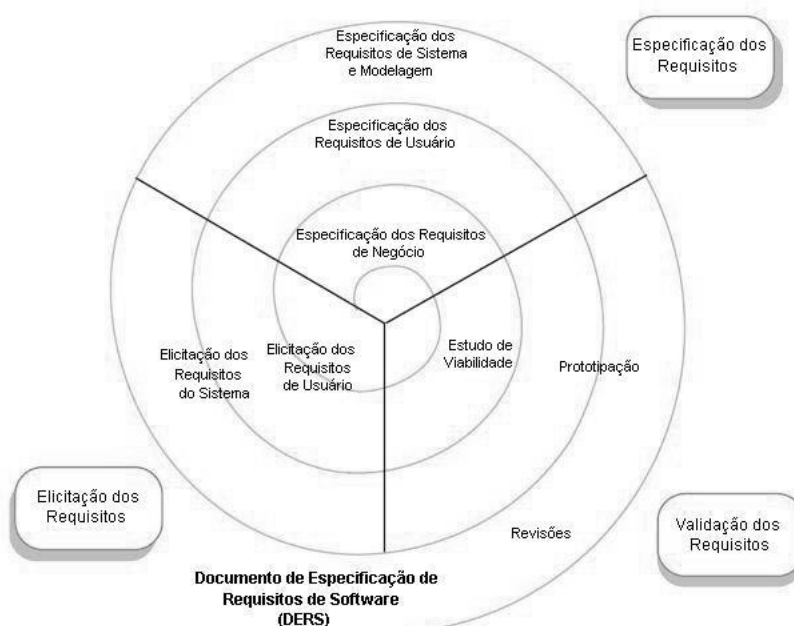


Figura 1 – O processo de Engenharia de Requisitos segundo (KOTONYA; SOMMERVILLE, 1998)

Segundo (KOTONYA; SOMMERVILLE, 1998), este processo é bem flexível, pois pode voltar às fases anteriores

que já foram realizadas como em um espiral. Após o estudo de viabilidade, temos as fases de elicitação, especificação e

validação, que podem ser divididas em subfases.

A elicitação é a fase que coleta os requisitos, é dividida em duas subfases chamadas Elicitação dos Requisitos do Sistema, que foca no levantamento das funções e restrições do sistema de uma forma mais detalhada, e Elicitação dos Requisitos do Usuário, onde é realizado o levantamento, em linguagem natural ou por diagramas, de quais funções que o sistema deve fornecer e quais restrições se devem operar.

A especificação é composta de três subfases: Especificação dos Requisitos de Sistema e Modelagem, que descreve as características de um sistema ou produto, Especificação dos Requisitos de Usuário, que consiste em descrever as atividades efetuadas pelos usuários na organização que irá utilizar o sistema, e Especificação dos Requisitos de Negócio, que descreve em termos de negócio o que deve ser entregue ou conseguido para fornecer valor ao sistema.

A validação é dividida em três subfases: Estudo de Viabilidade, onde se avalia do ponto de vista organizacional e tecnológico se o projeto é viável, Prototipação, que é uma apresentação visual final do produto que está sendo desenvolvido, e Revisões, onde analisam sistematicamente as especificações produzidas para garantir que o projeto corresponde ao que foi pretendido.

III. METODOLOGIAS ÁGEIS

Segundo (ZANATTA, 2004), em 2001, vários pesquisadores denominados de "Aliança Ágil" (Agile Alliance), dentre eles vários conhecidos internacionalmente no contexto da engenharia de software tais como Martin Fowler, Alistair Cockburn, Jim Highsmith, Kent Beck, Mike Beedle entre outros, com grande experiência no mundo do desenvolvimento de software, foram motivados a iniciar uma discussão de como ter uma forma eficaz e rápida para se obter produtos de software com qualidade e que atendessem as necessidades dos clientes de forma simples. Nesta ideia nasceu o chamado Manifesto Ágil que abrange essas características, com os dizeres (AGILE MANIFESTO, 2013):

“Gostaríamos de descobrir como desenvolver software e ajudar a outros a fazer software da melhor maneira possível seguindo os melhores caminhos; Queremos valorizar os seres humanos que participam desse processo com ferramentas eficazes; O software deve possuir uma documentação clara, compreensiva e simples; Queremos estar aptos às mudanças contando com a colaboração do cliente e obtendo um feedback através de uma ideia específica”.

Como qualquer processo, métodos e práticas no desenvolvimento de software, foram criados alguns princípios para os métodos ágeis:

- A nossa maior prioridade é satisfazer o cliente através da entrega rápida e contínua de uma versão do software sempre atualizada.
- Alterações sobre os requisitos mesmo sendo tardia, sempre serão bem vindas, pois isto pode ser um diferencial

competitivo ao cliente.

- Entrega de software com maior frequência, de preferência a cada semana ou mês, sempre visando o menor tempo possível.
- Os stakeholders, analista de requisitos e os desenvolvedores trabalham juntos diariamente no projeto.
- Manter uma equipe focada e motivada, aderindo confiança e ambiente necessário para o trabalho.
- A principal comunicação da equipe que participa do projeto é baseada em uma conversa face-a-face.
- A simplicidade é essência do negócio.
- Software funcionando é a medida principal do progresso.
- A atenção contínua a excelência técnica e um bom foco no projeto mantêm a agilidade.
- As melhores arquiteturas, requisitos, e projetos emergem de equipes auto-organizadas.
- A equipe em intervalos regulares reflete sobre como se tornar mais efetivo, por isso a mesma se ajusta e aperfeiçoa o seu comportamento de acordo com a necessidade.
- Promover a evolução do desenvolvimento sustentável. Todos stakeholders devem ser capazes de manter um ritmo constante visando sempre à melhoria.

Baseando-se nestes princípios do manifesto ágil, é notório que alguns valores existentes nos métodos atuais são e continuam sendo muito importantes, como por exemplo, a modelagem de dados e a documentação que são discutidas por Fowler (2013). Porém a preocupação é o excesso de papéis que são gerados com documentos que muitas vezes os mesmos nunca serão lidos. Outro fator é o planejamento, que é extremamente importante que seja utilizado, mas existe a consequência de limitações em termos de organizações, cultura do ambiente, comunicações entre pessoas, entre outras.

É possível considerar os métodos ágeis uma forma eficaz para realizar o gerenciamento de requisitos, na qual é subdividido em dois aspectos básicos (ZANATTA, 2004): ser adaptativo ao invés de preditivo e ser orientado a pessoas ao invés dos processos.

Existem algumas formas que se destacam nos métodos ágeis quando se fala em desenvolver um software, conforme (BECK et al., 2000): deve ser priorizado a satisfação do cliente, através da entrega rápida e contínua de software com qualidade, deve-se alterar os requisitos de forma rápida e eficaz, deve-se buscar um desenvolvimento sustentável e orientar todos que participam do projeto a praticar a simplicidade.

A) Scrum

Scrum (SCWABER; BEEDLE, 2002) é um processo para projetos que realizam desenvolvimento de software que seja focado nas pessoas no qual o ambiente tenha uma mudança constante nos requisitos. O Scrum pode ser também considerado uma técnica utilizada para realizar o gerenciamento de processo para o desenvolvimento de software.

Abrahamsson et al. (2002) relatam que a etapa de

desenvolvimento do Scrum inclui as fases tradicionais que existe no desenvolvimento de software como requisitos de usuário e requisitos de sistema; a análise; projeto e a entrega final do produto, ressaltando que todas essas questões precisam ser adaptáveis durante o seu processo.

O Scrum foi inicialmente desenvolvido pelas organizações de software Advanced Development Methods e VMARK Software, em meados dos anos 90, e os seus criadores foram Ken Schwaber e Jeff Sutherland com o intuito de gerenciar projetos de software (HIGHSMITH et al., 2002).

O nome Scrum nasceu por meio de uma jogada de rugby, que este nome é dado pela disputa da bola, onde todo o time avança, trabalhando em equipe com os seus jogadores como se fosse um só, passando sempre a bola pra um e para outro, para que marquem o máximo de pontos possíveis. Essa é a ideia no desenvolvimento de software, é justamente definir papéis bem específicos para todas as pessoas que estão envolvidas no projeto, com o intuito que a equipe realize o máximo de tarefas possíveis para ter uma nova interação (release) do produto final.

Sua característica básica são equipes pequenas entre 8 a 10 pessoas; os requisitos são desconhecidos ou mesmo instáveis e as interações são em um prazo curto de tempo. O desenvolvimento é realizado em intervalos de no máximo 30 dias, denominados de Sprints. Os requisitos levantados no Product Backlog são selecionados para serem feitos e quebrados em tarefas para ser realizadas no Sprint, porém essas tarefas não podem exceder o tempo de execução de uma semana. Caso isso ocorra é necessário que haja um novo desmembramento naquele item. É de extrema importância ressaltar que durante o Sprint, não há nenhuma mudança nas tarefas que estejam em desenvolvimento.

Outra característica importante é o encontro diário (Daily Meeting) que são reuniões curtas de quinze minutos realizadas diariamente para que a equipe do projeto se atualize de como está o andamento e que possa expor as dúvidas encontradas durante a realização das tarefas no Sprint.

Este método não possui ou impõe nenhuma técnica específica para que se realize o desenvolvimento do software, porém são estabelecidos conjuntos de práticas e regras para gerenciar e devem ser adotadas para que obtenha sucesso no projeto.

IV. APLICAÇÃO DA METODOLOGIA ÁGIL SCRUM NA ENGENHARIA DE REQUISITOS

O estudo de caso foi realizado em uma empresa situada em Ribeirão Preto (SP) que atua no mercado de desenvolvimento de software há mais de 20 anos na qual o seu produto atende clientes em todo o país e possui um processo de gerenciamento de requisitos que parte desde a sua entrada, manutenção e a saída que é o software pronto para o uso final. Conforme o levantamento, não existe nenhuma metodologia aplicada junto ao processo de gerenciamento de requisitos, e a empresa criou o seu próprio gerenciamento descrito mais adiante.

A empresa possui oitenta funcionários técnicos, seis

funcionários administrativos, um gerente e dois diretores.

O sistema comercializado pela empresa é único, que é alugado a clientes e cobrado por manutenções e customização específicas dos clientes, de acordo com as suas necessidades. As solicitações são implementadas e todos os clientes têm direito do uso das funcionalidades disponíveis independente se os mesmos foram solicitados por eles.

Nessa estrutura, o processo de gerenciamento de requisitos é realizado conforme descrição a seguir:

- A empresa possui um sistema pronto que é comercializado. O processo começa com a implantação desse sistema, com as pessoas responsáveis por realizar toda a instalação e treinamento do sistema para os novos clientes que irão utilizá-lo. Na maioria das vezes são necessárias várias adaptações decorrentes da implantação. Essas adaptações consistem nos primeiros requisitos a serem levantados. A equipe de implantação também é responsável por realizar visitas de pós-implantação e periódicas para saber como está a satisfação do cliente com o software utilizado. Nessas visitas também ocorrem solicitações de novos requisitos.

- A equipe de atendimento (help desk) realiza o suporte, sanando dúvidas e realizando configurações que às vezes são necessárias para o cliente utilizar o software. Essas solicitações podem também incluir novos requisitos, que precisam ser tratados. Na maioria das vezes em que um novo requisito entra por esta equipe trata-se de correções de problemas ou erros inesperados no sistema.

- A equipe de qualidade também é responsável pelo levantamento de requisitos. Na realização dos testes de validação para a liberação de uma nova versão, muitas vezes são encontrados erros pré-existentes ou falhas no processo que podem ser relacionados com requisitos para melhoria no sistema.

Os tipos de requisitos são avaliados de três maneiras:

- Erro: quando o sistema não funciona como esperado, potencialmente causando problemas.
- Customização: novos requisitos que são solicitados a pedido do cliente, geralmente para adaptação.
- Melhoria: são novas funcionalidades que agregam valor ao sistema sem gerar nenhum custo a quem realizou a solicitação.

Esses requisitos são registrados em uma ferramenta própria chamada Diário. Essa ferramenta permite que as diferentes pessoas envolvidas no processo, tais como atendentes de help desk, implantadores, programadores, testadores, analista de negócios e a gerência, que participam desde a entrada, manutenção e saída desse requisito, consigam acompanhar como está o andamento do mesmo. Dessa forma, todos podem ficar cientes de quando sairá uma próxima versão para satisfazer as pendências que são aguardadas pelos clientes.

São definidos três tipos de requisitos, porém os requisitos de erros não precisam passar pelo processo de engenharia de requisitos. Esses três itens são tratados de forma conjunta, pois os problemas são comuns a todos. Os erros são classificados nesse contexto para chegarem até desenvolvimento, teste e liberação da correção do software.

Quando um novo requisito é solicitado, existem duas

peças responsáveis por avaliá-lo. Essas pessoas, que são analistas de negócio, verificam, por exemplo, se os erros realmente são erros ou se são apenas uma falta de entendimento do sistema. Também verificam se é necessário realizar uma customização solicitada ou se existe outra opção já disponível no sistema. E decidem se melhorias solicitadas realmente irão agregar valor para o sistema. Porém, quando os analistas de negócios realizam esse processo são encontradas algumas falhas.

Uma das principais falhas é a falta de comunicação, que é observada atualmente na empresa. A falta de comunicação com os clientes é um dos principais problemas, podendo resultar no levantamento de requisitos desnecessários, e muitas vezes causando desconforto no momento da implementação do software. A comunicação interna também é escassa, pois a equipe de atendimento e implantadores, que são os que têm maior interação com os clientes, não têm a prática e regra de conversarem com os analistas de negócios ou mesmo programadores.

Os analistas de negócios conversam entre si, porém nem sempre ambos têm o conhecimento de toda a ferramenta completa devido à falta de documentação, que é outro problema observado. Assim, quando os requisitos são validados por eles, é comum serem registrados requisitos de melhorias, customizações e, principalmente, correções de erros, que não são realmente necessários para o sistema. Se houver uma melhor interação entre a equipe do desenvolvimento junto aos analistas de negócio este problema pode ser resolvido.

Além disso, há o problema de falta de qualificação das pessoas que levantam os requisitos. Atendentes de help desk nem sempre são pessoas qualificadas para fazer esse tipo de trabalho por falta de conhecimento técnico e de negócio, o que dificulta um pouco o processo.

Cita-se também o fato de que os implantadores dificilmente ficam na empresa, pois os mesmos estão constantemente realizando viagens de instalações e treinamentos. Eles são os maiores responsáveis por trazerem requisitos de customizações, que são normalmente aqueles que causam maior impacto. Como resultado, a comunicação com esses profissionais também é dificultada.

Por último, a forma de conduzir os requisitos que estão sendo implementados atualmente é falha. Muitas vezes, devido ao tempo escasso e a necessidade de alterar os requisitos que estão sendo implementados, muitas tarefas são abortadas e novos requisitos são implementados sem nenhum planejamento e isso é um fator prejudicial em termos da qualidade, pois com isso podem ser gerados repetidos requisitos de erros.

Nota-se, portanto, que há uma série de problemas no processo da empresa. Para tentar resolvê-los, propõe-se aqui a utilização da metodologia ágil Scrum que, com a sua facilidade de uso pode ser adaptável a realidade da empresa. A ideia é propor soluções para se obter um melhor resultado a cada liberação de um incremento (versão release) do software proprietário.

A) Proposta de melhoria utilizando o Scrum

Seguindo as regras e práticas do Scrum, a principal característica é a comunicação da equipe. Esta prática se for aplicada no caso estudado, reduziria drasticamente esta falha.

Mas também é notória a falta de papéis dentro da empresa. Tanto a equipe de atendimento como implantadores fazem um mesmo trabalho de levantamento de requisitos, o que pode ocasionar problemas de responsabilidade e hierarquia. Assim sendo, se forem atribuídos os papéis descritos no Scrum há maiores chances de respeito à hierarquia e senso de responsabilidade. Isso pode também tornar a comunicação mais eficaz. Muitas vezes a própria gerência e diretoria têm influência nesse tipo de problema, querendo passar requisitos que não foram analisados como prioritários para adquirir novos clientes.

A.1) Uso de papéis Scrum

Na metodologia ágil Scrum existem papéis e responsabilidades diferentes na qual o objetivo é executar os trabalhos durante a utilização das práticas. Abaixo serão descritos esses papéis (SCHWABER; BEEDLE, 2002):

Product Owner - É o dono do produto, geralmente é um cliente ou especialista do negócio o qual está sendo desenvolvido. Ele defende os interesses do negócio e tem como objetivo e responsabilidade de priorizar e manter os itens do Product Backlog. No caso da empresa estudada, o Product Backlog consiste no conjunto de requisitos constantemente levantados para o sistema. Na empresa analisada, propõe-se que esse papel seja desempenhado pelos analistas de negócios, pois são eles quem mais sabem sobre os interesses dos requisitos e se os mesmos irão agregar valor ao sistema.

Na prática do Scrum, a ideia é que o Product Owner seja uma única pessoa. Porém na empresa estudada existem claramente dois profissionais com essas atribuições. Portanto a proposta é que o papel seja atribuído a apenas um deles por cliente, e o outro atuará em seu auxílio. Essa pessoa representará o cliente, sempre realizando os contatos periódicos com a maioria deles, pois a empresa possui vários clientes e nem sempre os mesmos solicitam ou necessitam de algo nas mesmas circunstâncias.

Scrum Master - Este papel é responsável pela aplicação de valores, práticas e pelo objetivo concretizado no final do Scrum. Ele é considerado como um "facilitador" das reuniões removendo os obstáculos ou qualquer impedimento que comprometa o projeto. Também protege a equipe fazendo com que eles não se comprometam a realizar tarefas que o time não pode sanar. Ele é a principal chave que une o negócio e as tecnologias e segundo (ZANATTA, 2004) suas principais atividades são organizar as Daily Meetings, representar o gerente do projeto e o técnico, gerenciar todo processo Scrum dentro da organização, e remover os impedimentos que comprometam o projeto.

Com o conceito definido deste papel, foi analisado que a melhor pessoa por esse papel seria uma pessoa da gerência ou diretoria, pois na empresa existem dois sócios e o gerente de

projeto, o qual também faz o papel de analista de negócio. Ambos são divididos na seguinte questão: Um diretor é responsável pela implantação e outro é responsável pelo desenvolvimento. O gerente de projetos faz o papel de gerência e de analista de negócio.

Analisando o papel do Scrum Master o que melhor se adaptaria seria o gerente de projetos, pois ele é o que tem maior afinidade e contato com os clientes, e também por possuir um vasto conhecimento no negócio ele também é capaz de sanar as dúvidas técnicas. Mesmo sabendo que os diretores são os que participam mais no gerenciamento de novos clientes, nem sempre será possível encontrá-lo presente na empresa, assim dificultado o processo do Scrum.

Scrum Team - Por meio de encontros com o Scrum Master, o Scrum Team é representado pela equipe de desenvolvimento. Não há necessidade de uma divisão entre os integrantes desta equipe como, por exemplo, o programador, o analista de banco de dados, o testador, entre outros. O foco principal é que todos trabalhem juntos no projeto para entregarem o conjunto de trabalho a qual foi prometido durante a Sprint. A equipe típica costuma-se ter entre 8 a 10 integrantes, podendo-se variar dependendo do ambiente em que se aplica. Pode existir mais de um Scrum Team dentro de uma organização, e eles irão trabalhar separadamente, porém quando se tem grandes equipes e existe a necessidade de criar outras, é usado o conceito de "Scrum of Scrums" (DESENVOLVIMENTO ÁGIL, 2013). Em cada Scrum Team será necessário que um integrante frequente o Scrum of Scrums Meeting, para que eles conversem entre si para que as mesmas possam coordenar o trabalho de múltiplas equipes.

Atualmente na empresa a equipe de desenvolvimento é composta por programadores, testadores e analistas de banco de dados, totalizando 21 integrantes. Baseando-se nas recomendações do Scrum propõe-se criar três equipes, cada uma composta por cinco desenvolvedores, um analista de banco de dados e um testador. Essa divisão foi realizada na tentativa de incluir as três principais categorias de profissionais necessários para realizar implementações completas em todos os aspectos do sistema. A divisão também tenta manter em uma mesma equipe profissionais especializados nos mesmos módulos do sistema, mantendo assim as equipes mais coesas. Por existir mais de uma equipe, existiu a necessidade de deixar uma responsabilidade maior para um dos integrantes realizar o Scrum of Scrums Meeting. Esta questão não foi trivial, pois dentro da empresa já existia um conceito de que uma pessoa de cada módulo fosse líder de um determinado módulo. Esse perfil é de um desenvolvedor com maior experiência no módulo e conseqüentemente com mais tempo de colaboração.

A.2) Práticas do Scrum

Além da definição dos papéis descrita anteriormente, propõe-se a utilização de algumas práticas para realizar o gerenciamento com o Scrum tais como o Product Backlog, Daily Scrum, Sprint, Sprint Planning Meeting, Sprint Backlog e Sprint Review Meeting.

Product Backlog é na onde se inicia todo o processo do Scrum, sendo considerada a prática responsável pela parte da elicitação da área de engenharia de requisitos onde os analistas de requisitos realizam as coletas dos requisitos (SCHWABER; BEEDLE, 2002).

Realizando reuniões e técnicas de levantamento de requisitos com todos stakeholders do projeto, são levantados e registrados os itens com todas as necessidades dos requisitos de negócios e técnicos para serem desenvolvidos, padrões, tecnologia entre outros fatores. Em outras palavras, para a empresa em questão, o Product Backlog é a lista de atividades que serão desenvolvidas no projeto.

Esta prática pode ser realizada usando a ferramenta proprietária Diário, pois ela contém todos os requisitos que são necessários para o desenvolvimento do produto e o seu término seria o software completo conforme solicitado. Um aspecto interessante dessa ferramenta é que é possível saber quem fez uma solicitação, quem a realizou, e quem a aprovou. Apesar de que Scrum o foco é o resultado e não a rastreabilidade (ZANATTA, 2004), acreditamos que isto pode contribuir para a melhoria no processo.

Linda e Norman (2000) explicam que Daily Scrum são as reuniões realizadas dentro do Scrum durante o desenvolvimento do projeto. Na maioria das vezes são feitas diariamente, com duração de quinze minutos aproximadamente, lembrando que esse tempo não pode exceder o máximo de trinta minutos. Essa reunião é realizada para que a equipe Scrum possa expor suas dificuldades ou impedimentos que estão tendo durante a implementação do projeto. É importante ressaltar que essas reuniões não são utilizadas para a solução do problema e sim, para expor os problemas. É de responsabilidade do Scrum Master encontrar uma melhor maneira de solucionar esses problemas. O Scrum Master, durante as reuniões, deve levar à equipe três importantes perguntas (ZANATTA, 2004) APUD (SCHWABER; BEEDLE, 2002): o que já foi finalizado desde a nossa última reunião, existe alguma dificuldade encontrada nos trabalhos que estão sendo realizados, e quais são as próximas atividades que serão entregues no nosso próximo encontro.

No estudo de caso realizado essas reuniões seriam uma prática muito eficaz, pois a falta de comunicação interna é evidente e isso contribuiria com o alinhamento de todas as pessoas que fazem parte do processo. Apesar dessa prática não incidir tanto no processo dos requisitos de customização, ela pode agregar valor no momento em que o desenvolvimento encontra falha no processo de requisitos de melhorias e erros.

Com relação a Sprint pode-se dizer que é considerada a principal parte do Scrum, pois é nela que são executados os itens que foram adicionados no Product Backlog pelos Product Owner e os stakeholders do projeto. Esta etapa não pode exceder 30 dias, mas em média pode durar de uma a quatro semanas. No Sprint normalmente adota-se a mesma divisão tradicional no processo do desenvolvimento do software: requisitos, análise, projeto e entrega. Ao término de cada Sprint a equipe Scrum cria uma demonstração simples para ilustrar ao cliente como está o andamento do projeto, e

com isso é gerada uma sensação que o desenvolvedor executou sua tarefa com êxito (BEEDLE et al., 1999). O teste é um fator muito importante a ser realizado dentro desta etapa, pois assim é possível garantir a qualidade e validar se os requisitos foram implementados de acordo com o especificado, significando menos problemas e reduções na lista do Product Backlog. Caso seja encontrada alguma falha e a Sprint ainda não tenha terminado, o mesmo deve ser corrigido, ou gerado um novo item no Product Backlog como déficit técnico para ser realizado no próximo Sprint.

Para alinhamento dessa prática no presente estudo de caso, uma das maiores contribuições seria a organização nos requisitos em andamento, evitando assim o problema de duplicação de requisitos descrito anteriormente, pois a forma de conduzir os requisitos que estão sendo implementados é falha. Muitas vezes devido ao tempo escasso e a necessidade de alterar os requisitos que estão em andamento, muitas tarefas são abortadas e novos requisitos são implementados sem nenhum planejamento e isso é o principal fator que é prejudicial nos termos de qualidade. Essa é a principal questão levantada que fazem gerar erros inesperados. A ideia da Sprint ter uma duração de no máximo 30 dias atualmente já é suprida, pois nos dados levantados a empresa já trabalha com prazos curtos de desenvolvimento. O acompanhamento de como está a implementação dos requisitos também já existe de uma forma simples. A única alteração proposta é a implementação de uma taskboard dentro da ferramenta Diário, para que os desenvolvedores e testadores consigam alterar o andamento de como está o desenvolvimento de um determinado requisito, o que hoje não é possível.

O Sprint Planning Meeting é a reunião que é realizada na qual devem estar presentes o Product Owner, o Scrum Master e todo a equipe do Scrum (DESENVOLVIMENTO ÁGIL, 2013). É nesta etapa que o Product Owner descreve quais as funcionalidades são de maior importância e que têm maior relevância para serem desenvolvida primeiro. Não é necessário que o Product Owner descreva todas as prioridades e sim, de preferência a quais irão participar daquele Sprint. Nesse momento o Scrum Team irá realizar perguntas de modo que seja possível o refinamento dos Backlog Items em Task (tarefas) técnicas logo após a reunião. Esses conjuntos de Tasks darão a origem ao Sprint Backlog. Após a criação do Sprint Backlog o Scrum Team e o Product Owner irão definir o objetivo daquele Sprint, que será uma breve descrição do que irá ser entregue e qual será o objetivo dos requisitos implementados. Muitas vezes esses requisitos podem ser implementações implícitas que pode obter nenhum resultado final palpável pelo cliente, porém isso é discutido e explicado durante o Sprint Planning Meeting. Após toda a definição e o Sprint Backlog aprovado pela equipe Scrum é possível se iniciar a Sprint.

Essa prática, se introduzida na empresa estudada, apesar de aparentemente ocupar tempo produtivo, faria com que os desenvolvedores e analistas de negócios expusessem mais as ideias do que já existe na ferramenta, e o que realmente existe a necessidade de implementar, pois conforme levantado muitos requisitos eram solicitados e os mesmo já

existiam ou havia uma outra forma de utilização no sistema.

Essa proposta tem como objetivo reduzir consideravelmente grande parte de requisitos ambíguos e desnecessários. Outro fator a ser considerado é a maior facilidade em se medir os esforços e tempos para a entrega de um requisito durante as reuniões. Existem muitos requisitos solicitados com prazos escassos, e promessas feitas ao cliente sem a consulta prévia à equipe sobre o impacto que o mesmo poderia causar. A redução de estresse na equipe de desenvolvimento e a pressão dos clientes serão reduzidas notavelmente, pois as negociações dos requisitos são melhores planejadas.

O Sprint Backlog como mencionado acima, é uma lista de tarefas que é gerada a cada Sprint Planning Meeting, a qual a equipe Scrum se compromete a cumprir durante a Sprint corrente. Esses itens são extraídos do Product Backlog, os quais foram priorizados pelo Product Owner, baseando-se na complexidade e tempo que será necessário para se realizar aqueles requisitos. Cabe ao Scrum Team determinar a quantidade de itens que serão implementados do Product Backlog, pois o Sprint Backlog já é a total responsabilidade do time com relação a uma Sprint (DESENVOLVIMENTO ÁGIL, 2013). O papel do Scrum Master nesta etapa é manter o Sprint Backlog atualizado, para que toda a equipe e ele possam refletir sobre quais as tarefas foram completadas e qual será o tempo necessário para poder terminar as demais que ainda estão na lista para serem feitas. Na prática pode acontecer de alguns dos itens que foram selecionados no Sprint Backlog não poderem ser implementados, por isso cabe ao integrante do Scrum Team expor isso nas Daily Scrum para que o Scrum Master possa verificar o que pode ser feito, como por exemplo, passar a Task para outro integrante do time, ou mesmo negociar com Product Owner.

Realizando a adaptação dessa prática para a empresa estudada, a sobrecarga de trabalho no Scrum Team, caso exista, passará a ficar explícita, pois depois de definidos os itens que serão entregues em cima dos prazos comprometidos, são evitados os requisitos passados sem nenhum planejamento. Também fica notório para os diretores e gerente que todos os colaboradores da equipe possuem tarefas e que ambos estão se empenhando no projeto.

Na finalização de cada Sprint é realizada a Sprint Review Meeting. Nesta reunião, o Scrum Team mostra o que foi realizado durante a Sprint. Formalmente, isso pode ser apresentado em uma demonstração ou uma prévia descrição das novas funcionalidades. Essa reunião é realizada junto à equipe do Scrum, Scrum Master, Product Owner e os demais stakeholders que compõem o projeto. O Scrum Master junto ao Product Owner mais os stakeholders irão analisar se o incremento satisfaz o que foi comprometido na Sprint Planning Meeting. Mesmo que todos ou alguns itens não tenham sido implementados no Sprint Backlog, a ideia principal é que a equipe atinja o objetivo final daquela Sprint.

Adotando-se esta prática no estudo de caso, a proposta é haver um aumento da satisfação do cliente e conseqüentemente a satisfação da equipe de desenvolvimento, pois nessas reuniões, conforme a

apresentações das demonstrações os clientes podem ter a confiança da utilização da nova funcionalidade produzida em um pequeno espaço de tempo.

B) Resultados preliminares e discussão

A proposta de melhoria está em estágio avançado de implementação, tendo já implementado a taskboard e todos os papéis e algumas práticas do Scrum tais como o Product Backlog, Daily Scrum, a Sprint, a Sprint Backlog e o Sprint Review Meeting. Com os papéis bem definidos foi simples organizar algumas práticas dentro da organização, e com a implementação da taskboard no aplicativo Diário a equipe pode acompanhar como está o andamento dos requisitos. Uma prática que ainda falta ser implementada é Scrum of Scrums Meeting, pois está sendo necessário realizar algumas adaptações de horários e pessoas entre as reuniões diárias, para que depois implemente as reuniões entre as equipes. O Sprint Planning Meeting foi parcialmente implementado pois existem trabalhos atrasados e precisam ser realizados devido aos prazos prometidos anteriormente.

Analisando os resultados ficou explícita a questão de como a metodologia ágil melhorou o processo da engenharia de requisitos. Focando nos pontos principais que são a elicitação, especificação e validação, a utilização das práticas e regras do Scrum, junto com a definição de papéis aplicado dentro da empresa, obteve-se um resultado satisfatório. A redução de requisitos desnecessários e redundantes foi efetivamente observada, causando grande impacto em termos de agilidade.

Outro ponto positivo é que, mesmo que a utilização de algumas práticas provoque alguma perda de tempo produtivo com reuniões, este tempo é sutilmente pequeno analisando o tempo que se perdia implementando requisitos sem necessidade ou mesmo o tempo que era gasto para se fazer o levantamento do mesmo. Observou-se alguma reclamação, mas principalmente por parte da gerência. No geral, a equipe e os clientes mostraram-se satisfeitos com o resultado.

Com as tarefas sendo registradas, e com o explicativo de que aquele requisito faz, foi possível aumentar a capacidade dos envolvidos em saber como está o software atualmente.

Com relação ao foco na comunicação, a utilização de papéis mais bem definidos levou a elicitação a um nível mais produtivo. Foi possível aproximar não só os clientes e analistas de requisitos, mas todos os integrantes da equipe, tais como programadores, testadores e outros, que passaram a ter uma melhor visão do negócio e do produto a ser implementado. Dar essa importância a todos da equipe emotiva e aumenta as ideias para encontrar uma melhor solução para o incremento final do produto.

A especificação também tornou-se mais clara, sendo que o Scrum Team já obteve o conhecimento do contexto e o objetivo final do que tem que ser feito, pois as conversas realizadas com o Product Owner e os stakeholders fizeram com que eles entendessem melhor o dia a dia e a dificuldade que os usuários tem em manusear o software, assim obtendo uma ideia e planejamento melhor na hora de realizar a implementação para que satisfaça o cliente da maneira que

foi solicitada.

A validação passou a ser mais sucinta, pois com os clientes presentes a cada levantamento e a cada apresentação do incremento, já era realizada a aprovação se o que foi realizado abrange a necessidade acordadas no início de cada Sprint. Realizando a apresentação formal, foi possível analisar a satisfação do cliente quanto ao que foi prometido. A satisfação é um ponto muito importante no Scrum quando se refere ao cliente, mas isto foi recíproco a equipe também, pois ver o resultado positivo do produto fez com que a equipe se motivasse mais fazendo com que haja harmonia e um bom relacionamento entre eles. Este ponto, aliás, é uma das características que a metodologia ágil foca: “A valorização das pessoas que fazem parte do projeto”.

Em resumo, com a realização das três principais fases da engenharia de requisitos aplicados junto à metodologia ágil Scrum foi possível obter um documento de todos os itens que foram e serão implementados de forma simples, clara e objetiva sem muita burocracia. Foi também adquirida a tão importante comunicação que fez acontecer um melhor planejamento do produto a ser implementado realizando a integração do time como se fosse um só com o principal objetivo de obter o que foi acordado com qualidade, agilidade e a satisfação do cliente.

V. TRABALHOS RELACIONADOS

Na metodologia Scrum a simplicidade e a agilidade são os principais aspectos as quais suas práticas induzem realizar o gerenciamento baseado principalmente nas pessoas e nos processos. Diferente de outras práticas de gerenciamento de requisitos que envolvem regras e técnicas na qual o principal processo é realizar um gerenciamento mais burocrático perde-se a eficiência da praticidade. Para o Scrum esses quesitos são diferenciais, pois o seu principal objetivo é a satisfação do cliente, e para se garantir isso é necessário que haja qualidade e gerenciamento eficaz. Na literatura, é possível encontrar alguns esforços em se aplicar métodos ágeis e métodos de gerenciamento de requisitos.

Zanatta (2004) propõe a utilização da metodologia ágil Scrum para a gerência de requisitos utilizando as práticas do CMMI e fazendo uma adaptação. O resultado foi satisfatório, mas nem todas as práticas abordadas pelo Scrum são atendidas pelo conceito do CMMI. Segundo o autor, isso se deve ao fato de que o CMMI introduz práticas de gerenciamento burocráticas, e por isso muitos de seus itens CMMI não seriam contemplados.

Espindola et al. (2004) realizam uma análise crítica do mau gerenciamento dos requisitos e enfatiza o impacto das negociações dos requisitos. Os autores afirmam que a análise é responsável por identificar requisitos conflitantes, esquecidos, ambíguos ou irreais. Entretanto esta atividade realizada com os interessados do projeto pode ser prejudicada devida à falta de interação entre os stakeholders e o analista de requisitos. O Scrum propõe que a constante interação entre os interessados do projeto é uma das principais características para o sucesso do negócio, e é um dos princípios baseados na metodologia.

O presente estudo de caso vem corroborar outras observações da literatura, como por exemplo, a praticidade do Scrum e a importância da comunicação. Com isso aumenta-se a evidência em favor dessa prática ágil já bastante difundida.

VI. CONCLUSÕES E TRABALHOS FUTUROS

Concluimos com este trabalho que é importante ressaltar que quando os requisitos não são bem definidos, a obtenção da qualidade é dificultada, assim como cumprimento de prazos e custo dentro do contexto do desenvolvimento de software. Problemas similares foram relatados com base em um estudo de caso em uma empresa de software, que vivencia na prática essa realidade.

Com as melhorias propostas, observou-se que a metodologia ágil Scrum pode proporcionar esse fator de qualidade, prazo e custo como as demais metodologias convencionais, porém visando a agilidade e a satisfação do cliente, obtendo softwares mais rápidos e pertinentes conforme a solicitação do cliente. Com os papéis, práticas e regras do Scrum, são explícitas as melhorias que foram aplicadas na utilização desta metodologia.

Como trabalhos futuros, sugere-se a realizar a utilização de mais de uma metodologia ágil em conjunto, como por exemplo, o Scrum e o Extreme Programming (XP), pois de acordo com Abrahamsson et al. (2002) quando mais de uma metodologia ágil é utilizada ao mesmo tempo, melhora-se o processo de agilidade. Por exemplo, um ponto forte da utilização do XP seria o apoio na melhoria no fator da utilização das práticas da engenharia de software tais como Test Driven Development (TDD), Refactoring, Pair Programming entre outras. Como o Scrum não define o uso desse tipo de técnica, o XP seria um bom complemento ao processo, proporcionando maior agilidade também no âmbito da programação.

REFERÊNCIAS

- LEITE, J.C.S.P. Engenharia de Requisitos- Notas de Aula., 1994.
- KOTONYA, G.; SOMMERVILLE, I. Requirements Engineering: Processes and Techniques., John Wiley & Sons Ltd., 1998.
- LEFFINGWELL, D.; WIDRIG, D. Managing Software Requirements: a unified approach., Addison-Wesley, 2000.
- LEFFINGWELL, D.; WIDRIG, Don. Managing Software Requirements – A Use Case Approach. Addison-Wesley, 2003, 492p.
- SOMMERVILLE, I.; SAWYER, P. Requirements Engineering – a good practice guide. New York: John Wiley & Sons Ltd, 1997, 391p.
- ESPINDOLA, R. S.; MAJDENBAUM, A.; AUDY, J. L. N. Uma Análise Crítica dos Desafios para Engenharia de Requisitos em Manutenção de Software. In: VII Workshop on Requirements Engineering, 2004, Tandil, Argentina, 2004.
- AGILE SURVEY; 7th Annual Survey: 2012 The State of Agile Development Conducted: June-July, 2012 - Disponível em: <http://www.versionone.com/> Acessado em: 10 agosto 2013.
- BEEDLE, M.; DEVOS, M.: SCRUM: A Pattern Language for Hyperproductive Software Development. In Pattern Languages of Program Design 4, editado por N. Harrison, B. Foote, e H. Rohnert. Addison-Wesley (1999).
- SCHWABER, K.; BEEDLE M.: Agile Software Development with Scrum, Upper Saddle River, NJ, Prentice Hall. (2002).
- ZANATTA L. A. xScrum: uma proposta de extensão de um Método Ágil para Gerência e Desenvolvimento de Requisitos visando adequação ao CMMI. Florianópolis, 2004. 180 páginas. Dissertação (Mestrado em Ciência da Computação) - Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina.
- FOWLER, M. The New Methodology. Disponível em: <<http://www.martinfowler.com/articles/newMethodology.html>>. Acesso em : set 2013.
- HIGHSMITH, J.; Agile Software Development Ecosystems. Addison Wesley, 2002.
- LINDA R.; NORMAN J.,: The SCRUM Software Development Process for Small Teams, IEEE Software, July-August 2000.
- DESENVOLVIMENTO ÁGIL; Disponível em: <<http://desenvolvimentoagil.com.br/>>. Acesso em 26 set, 2013.
- ÁGILE MANIFESTO; Disponível em: <<http://www.agilemanifesto.org/principles.html>>. Acesso em 23 set, 2013.
- ABRAHAMSSON, P.; SALO, O.; RONKAINEN, J.; WARSTA, J. Agile Software Development Methods: Review and Analysis. Espoo, VTT Electronics, 107 p. Publicação VTT 478. (2002).
- BECK, K. Extreme Programming: Embrace Change. Addison Wesley. 2000.