

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-
GRANDENSE, CAMPUS PASSO FUNDO
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET**

LUÍS AUGUSTO DE RAMOS RODRIGUES

UMA ANÁLISE DE SCRUM

Carmen Vera Scorsatto

PASSO FUNDO(RS), 2014.

LUÍS AUGUSTO DE RAMOS RODRIGUES

UMA ANÁLISE DE SCRUM

Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-Rio-Grandense, *Campus* Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientadora: Carmen Vera Scorsatto

PASSO FUNDO(RS), 2014.

LUÍS AUGUSTO DE RAMOS RODRIGUES

UMA ANÁLISE DE SCRUM

Trabalho de Conclusão de Curso aprovado em ____/____/____ como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet

Banca Examinadora:

Prof. Esp. Carmen Vera Scorsatto

Prof. Me. André Fernando Rollwagen

Prof. Dr. Alexandre Tagliari Lazzaretti

Prof. Dr. Alexandre Tagliari Lazzaretti

(Coordenador do Curso)

PASSO FUNDO, 2014

*Aos meus pais,
pela compreensão e o estímulo
em todos os momentos.*

RESUMO

Este trabalho apresenta uma análise bibliográfica e estudos de caso tendo como base o processo ágil de produção de software *Scrum*, visando responder sobre os benefícios de seu uso para empresas desenvolvedoras de *software*. A questão principal do trabalho é analisar este processo através de estudo da bibliografia, aplicação de questionário e simulação de ciclos de *Scrum* dentro de um projeto de uma empresa.

Palavras-chave: *Scrum*; Engenharia de *software*; Desenvolvimento de produtos de *software*; Processo Ágil.

ABSTRACT

This work presents a literature review and case studies based on the Agile SCRUM process of producing software in order to answer about the benefits of its use for software development companies. The main question of the paper is to analyze this process by analyzing literature, a questionnaire and simulation of Scrum Sprints within a project in a company.

Key words: Scrum; Software engineering; Development of software products; Agile Process.

SUMÁRIO

1	INTRODUÇÃO	7
1.1	OBJETIVOS	8
1.1.1	Objetivo Geral	8
1.1.2	Objetivos Específicos	8
2	DESENVOLVIMENTO	8
2.1	QUANTO A NATUREZA DA PESQUISA	9
2.2	AS CONTRIBUIÇÕES TEÓRICAS	9
2.3	A IMPORTÂNCIA DO TEMA	9
2.4	A POSSIBILIDADE DE SUGERIR MUDANÇAS.....	10
2.5	PROBLEMA DA PESQUISA	10
2.6	HIPÓTESE BÁSICA	10
2.7	REFERENCIAL TEÓRICO SOBRE SCRUM	11
2.7.1	Sprint.....	11
2.7.2	Artefatos.....	12
2.7.3	Time-Boxes.....	14
2.7.4	Reuniões.....	15
2.7.4.1	Release Planning Meeting	15
2.7.4.2	Sprint Planning Meeting.....	15
2.7.4.3	Daily Scrum Meeting	16
2.7.4.4	Sprint Review Meeting.....	16
2.7.4.5	Sprint Retrospective	17
2.7.5	Papéis	17
2.7.6	Método.....	19
2.7.7	Sistemas que auxiliam o uso de Scrum	19
2.8	METODOLOGIA	24
2.8.1	Método de abordagem.....	24
2.8.2	Método de procedimento.....	24
2.8.3	Aplicação e resultados do questionário	25
2.9	TRABALHOS RELACIONADOS	26
2.10	SIMULAÇÃO DE SCRUM EM PARTE DE UM PROJETO DE UMA EMPRESA...	28

3	CONSIDERAÇÕES FINAIS.....	32
3.1	RECOMENDAÇÕES PARA A IMPLEMENTAÇÃO DE SCRUM.....	33
	REFERÊNCIAS	35
	ANEXO (A) QUESTIONÁRIO	37

1 INTRODUÇÃO

É cada vez maior o interesse empresarial, por menores que sejam as empresas, em instalar sistemas computacionais conhecidos como *softwares*, ou ainda aplicações *web* que aproximem e simplifiquem a relação entre empresa e cliente. Porém, o gerenciamento da produção destes sistemas, ao longo de sua história, tem se mostrado uma tarefa tão difícil quanto necessária para se assegurar a qualidade do produto em um tempo determinado (PRESSMAN, 2010).

O gerenciamento de um projeto de *software* tem como objetivo reger todo o processo de confecção de um sistema. Esse sistema pode ter efeitos profundos e inesperados em empresas comerciais, nas pessoas e até na cultura como um todo, no que conhecemos como “lei das conseqüências não pretendidas” (PRESSMAN, 2010), que acontece quando algo é criado e só pode ser mensurado o seu início, sendo seus resultados imprevisíveis.

Devido a esse contexto, faz-se necessário um processo de produção de *software* que possibilite como produto um sistema de qualidade, que tenha sido feito dentro dos prazos estipulados e, principalmente, atenda às necessidades que o cliente solicitou, uma vez que entre vários fatores, o retorno financeiro da empresa desenvolvedora do software está comprometido.

Um processo conhecido, porém relativamente novo chamado *Scrum* que foi desenvolvido em 1990, tornou-se uma importante ferramenta na produção de *softwares* de projetos com prazos apertados, requisitos mutantes e criticidade de negócio (PRESSMAN, 2010).

Processos tradicionais de produção de sistemas, como Cascata, que data de 1970, por muitos anos atenderam às necessidades dos sistemas desenvolvidos na época, porém, há pelo menos duas décadas, este modelo vem sofrendo críticas sobre sua eficácia frente a novos desafios em *softwares* com diferentes necessidades, com novas tecnologias e, principalmente, com o advento da internet.

Torna-se necessária a pesquisa em questão, tanto do ponto de vista teórico quanto prático.

Suponha que para atender uma nova oportunidade de mercado foi realizado um projeto de 12 meses de trabalho para prover uma solução pioneira em aplicação *web*. Porém ao lançá-lo, se descobre que há pelo menos 11 meses outra empresa de mesmo porte já havia lançado um produto para esse mesmo setor do mercado e já conquistou 70% deste nicho. E o mais grave, o produto se tornará apenas mais um no mercado e, não se beneficiará do impacto do pioneirismo e inovação (MEDEIROS, 2009).

Procura-se entender como essa outra organização conseguiu fazer um lançamento tão rápido. Então descobre-se que a empresa em questão não começou a desenvolver a solução *web* antes das concorrentes.

Nessa incursão descobre-se que o segredo dessa equipe foi ter lançado pequenas partes usáveis desse sistema ao mercado. Dessa forma a empresa em questão recebeu muito cedo o *feedback* dos clientes o que possibilitou adaptar rapidamente o produto de forma que as próximas entregas estivessem sincronizadas ao desejo do mercado consumidor.

Devido a grande confiança que o concorrente conseguiu junto aos clientes, as outras organizações não tiveram a oportunidade de recuperar nenhum cliente perdido.

Então depois de muito *marketing* e ações promocionais, muito dinheiro investido, decide-se desistir do produto, assumir o prejuízo e fechar o negócio.

Esta pequena estória já indica uma justificativa para se pesquisar o tema *Scrum* e perceber possíveis benefícios no processo de desenvolvimento de sistemas (MEDEIROS, 2009).

1.1 OBJETIVO

1.1.1 Objetivo geral

Analisar a bibliografia sobre o *Scrum*, para perceber possíveis benefícios na sua utilização.

1.1.2 Objetivos específicos

- enumerar e analisar bibliografias que tratem da metodologia *Scrum*;
- aplicar questionário em empresa que utilize o *Scrum*;
- elaborar uma lista de recomendações da implantação do *Scrum* para aplicação em uma empresa;

2 DESENVOLVIMENTO

Conhecer a velocidade de uma equipe, talvez seja a chave para o sucesso ou o fracasso de um projeto de desenvolvimento de *software*, e, neste ponto em particular segundo (KNIBERG, 2007, p. 11):

“Se as equipes não conhecem sua própria velocidade, o *Product Owner* não pode criar um guia de produto com datas de *releases* com credibilidade. E sem datas de *release* interdependentes, a empresa poderá falhar e os investidores poderão perder seu dinheiro!”

Deve ser observado antes de começar de fato a pesquisar a bibliografia de *Scrum* que este é um processo de desenvolvimento de sistemas que deve ser variado conforme o problema que se deseja resolver, ou seja, não há uma única maneira exata de se usar o *Scrum*, o que se verá a seguir são os pontos cruciais que toda a utilização de *Scrum* adequada deve conter, como lembra (KNIBERG, 2007).

2.1 QUANTO A NATUREZA DA PESQUISA

Quanto à natureza, este estudo se enquadra como pesquisa aplicada, uma vez que gera conhecimento através de uma análise bibliográfica que pode ser utilizado para uma melhor produção de *softwares* (UNISC, 2012).

2.2 AS CONTRIBUIÇÕES TEÓRICAS

O estudo contempla contribuições teóricas, pois há uma grande carência de material bibliográfico que abranja o tema *Scrum*, como esclarece Carvalho e Mello (2009), pois embora este quadro esteja mudando lentamente, o número de publicações que discorrem sobre o tema ainda são reduzidos, principalmente em português. Isto fica claro quando os autores declaram que: “Como era de se esperar, tendo em vista a novidade do tema, não foi encontrado nenhum artigo de análise retrospectiva do *Scrum*.” (CARVALHO e MELLO, 2009, p. 8).

2.3 A IMPORTÂNCIA DO TEMA

Num mundo em que sistemas computacionais têm um papel cada vez mais central em praticamente todas as áreas, a falta de uma metodologia na produção de *softwares* pelas pequenas empresas vem causando prejuízos econômicos e, sobretudo, a perda de oportunidades, as quais não podem ser mensuráveis. Segundo Carvalho e Mello (2010), além do problema quanto ao custo dos projetos de *software*, existe também a necessidade de se melhorar a eficácia dos produtos. O autor também afirma que sistemas de *software* em si estão tornando-se a ferramenta básica e fundamental dos processos econômicos e sociais. Atualmente, é improvável que exista alguma atividade econômica que não sofra nenhum efeito de um sistema de informação, por isso, é de suma importância que eles sejam eficazes. Esse conjunto de fatores que marcam a economia realçam também o interesse sobre o papel

que as micro e pequenas empresas de base tecnológica podem ter no desenvolvimento de regiões e países.

2.4 A POSSIBILIDADE DE SUGERIR MUDANÇAS

Scrum é uma metodologia e também um *framework*, uma de suas premissas é que cada grupo que se propõe a utilizá-lo deve adaptar seu uso ao problema ao qual tentam resolver. Como disse (KNIBERG, 2007, p. 16):

Scrum é uma metodologia e um *framework*. O que significa que *Scrum* não vai te dizer exatamente o que fazer. A boa notícia é que vou te dizer exatamente como faço *Scrum*, em um nível de detalhes excruciantemente doloroso. A má notícia é, bem, esta é a maneira como EU faço *Scrum*. E isto não significa que Você deva fazê-lo exatamente da mesma maneira. De fato, eu também faria de maneira diferente, se encontrasse situações diferentes. O melhor e o pior do *Scrum* é que você é forçado a adaptar o processo para sua situação específica.

Como se pode perceber pela fala trazida pelo autor, o *Scrum* exige adaptações, fazendo com que seja necessário o conhecimento deste tutorial para implementar a melhor utilização possível desta metodologia.

2.5 PROBLEMA DA PESQUISA

Segundo estudo do *Standish Group*, que é uma organização americana que realiza pesquisas nos Estados Unidos um grande número dos projetos de software irá falhar. Projetos de desenvolvimento de software estão em caos. (*STANDISH GROUP*, 2009). O mesmo estudo também mostra um escalonamento em que 24% dos projetos serão cancelados antes de serem concluídos.

Esta pesquisa tenta responder sobre a utilidade de *Scrum* como ferramenta na melhoria da produção de *softwares* em aspectos como organização e economia.

2.6 HIPÓTESE BÁSICA

Preliminarmente, sugere-se que a utilização de *Scrum* ofereça como benefícios imediatos à sua aplicação num projeto o conhecimento sobre a velocidade da equipe, maior organização, mais objetividade nos resultados, menos tempo de implantação do projeto, além de uma maior comunicação entre os integrantes da equipe de desenvolvimento do sistema, como sugere Enrik Kniberg (2007).

2.7 REFERENCIAL TEÓRICO SOBRE SCRUM

A análise de bibliografias tem como objetivo embasar com trabalhos a importância do tema, ainda mais sobre a metodologia *Scrum*, que é um tema ainda pouco abordado como demonstra (Carvalho e Mello, 2009), ao considerarem que o *Scrum* ainda é um tema predominantemente empresarial e pouco acadêmico. Isto sugere que exista uma grande lacuna científica a ser preenchida por pesquisadores da área com a produção deste conhecimento. Neste sentido, trabalhos mostrando, através da pesquisa/ação, a implantação do *Scrum* em pequenas empresas de base tecnológica, sejam elas de *software* ou não, podem ser boas propostas para futuros trabalhos.

Scrum é uma metodologia ágil que auxilia na produção de *softwares* de maneira objetiva, através de entregas contínuas de executáveis ao cliente, ciclos, reuniões coesas, e outras técnicas que são agora apresentadas.

2.7.1 *Sprint*

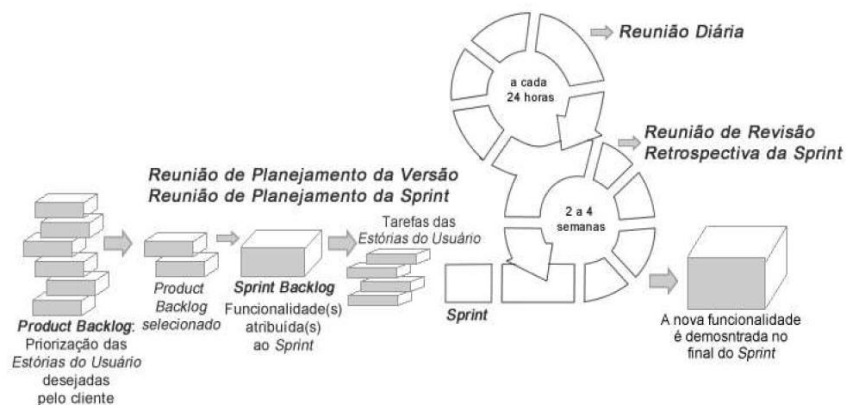
Uma das principais práticas que *Scrum* oferece é separar o projeto em ciclos, conhecidos como *Sprints*. Segundo Carvalho e Mello, a *Sprint* é considerada a principal prática do *Scrum*. É o período de tempo no qual são implementados os itens de trabalho definidos no *Backlog* do produto pela equipe *Scrum* (CARVALHO; MELLO, 2010).

É a principal prática de *Scrum*, é durante a *Sprint* que um utilizável do produto é criado, elas são compostas de reuniões de planejamento da *Sprint*, reuniões diárias, o trabalho de desenvolvimento, a revisão e a retrospectiva da *Sprint* (SCHWABER; SUTHERLAND, 2011).

É possível considerar cada *Sprint* como um projeto, que não deve durar mais do que quatro semanas (um mês) (SCHWABER; SUTHERLAND, 2011), como vemos representado na figura 1.

Para realizar um *Sprint* adequado e objetivo, é necessário planejá-lo, e isto é feito no *Sprint Planning Meeting*, tarefa na qual toda a equipe contribui.

Figura1 – Ciclo de Scrum



Fonte: (PRESSMAN, 2007).

2.7.2 Artefatos

Os artefatos do *Scrum* são representações de ações organizadas que, além da organização, fornecem as especificações detalhadas do sistema a ser produzido, com projeções que maximizam a transparência das informações chave, facilitando o trabalho da equipe para entregar o produto pronto (SCHWABER; SUTHERLAND, 2011). Os artefatos são o *Product Backlog* e o *Sprint Backlog* e são abordados a seguir.

O *Product Backlog* é o principal artefato do *Scrum*. O *Product Backlog* é basicamente uma lista de requisitos que são estórias, coisas que o cliente deseja, descritas utilizando a terminologia do cliente (KNIBERG, 2007).

Conforme (CARVALHO e MELLO, 2010), o *Product Backlog* é a parte do processo *Scrum* em que os dados preliminares (requisitos) são coletados junto aos clientes, colaboradores, parceiros e investidores do projeto, são apontados os itens com todas as necessidades do negócio e os requisitos técnicos a serem desenvolvidos.

Nesta primeira etapa, o importante é enfatizar o que o cliente necessita, esta coleta tem de ser resultado de um consenso entre o cliente, líder do projeto e a equipe de desenvolvimento no processo de produção do sistema.

Esta lista ordenada de tudo o que deve ser necessário no produto, é uma origem única dos itens para qualquer alteração a ser feita no produto. O responsável pelo *Product Backlog* é o *Product Owner*, cabendo a ele o conteúdo, a disponibilidade e a ordenação da lista (SCHWABER; SUTHERLAND, 2011).

O *Product Backlog* evolui na medida em que o produto e a equipe evoluem, é errôneo considerar o *Product Backlog* pronto, antes de todo o produto estar pronto, trata-se de um processo complexo e dinâmico, que muda constantemente. Vale lembrar que ele lista todas as

características, funções, requisitos, melhorias e correções que integram as mudanças a serem feitas no produto nas futuras versões (SCHWABER; SUTHERLAND, 2011).

Os requisitos inclusos no *Product Backlog* devem possuir os atributos da descrição, ordem e estimativa. Os itens no topo da lista demonstram as atividades mais imediatas, e devem se mais detalhados do que os itens mais abaixo na lista (SCHWABER; SUTHERLAND, 2011).

Os itens do *Product Backlog* devem conter ID, nome, importância do requisito para o *Product Owner*, campo como demonstrar, estimativa inicial e notas. Estes atributos são abordados a seguir.

O ID é o número de identificação único do requisito, o nome especifica basicamente o que se deseja daquele item, na linguagem do cliente, a importância do requisito para o *Product Owner* é uma pontuação, a estimativa inicial é o tempo necessário para implementar aquele requisito e é obtido multiplicando o número de integrantes da equipe que produziram o item pelos dias necessários para tal produção, o campo como demonstrar é uma especificação em alto nível de como o requisito será apresentado na demonstração da *Sprint* e as notas são informações extras que devem ser breves e objetivas (KNIBERG, 2007). A figura 2 representa esta lista.

Figura2 – Product Backlog(lista de requisitos do produto)

PRODUCT BACKLOG (exemplo)					
ID	Nome	Imp	Est	Como demonstrar	Notas
1	Depósito	30	5	Logar-se, abrir a página de depósito, depositar R\$ 10,00, ir para a página do meu saldo e verificar que este aumentou em R\$ 10,00.	Precisa de uma diagrama UML de sequência. Não é necessário se preocupar com criptografia por enquanto.
2	Verificar seu próprio histórico de transações	10	8	Logar-se, clicar em "transações". Fazer um depósito. Voltar para transações, verificar se o novo depósito é listado.	Usar paginação para evitar consultas muito grandes ao banco de dados. Projetar de forma similar à página de visualização de usuários.

Fonte: (KNIBERG, 2007).

O *Sprint Backlog* é um conjunto de itens retirados do *Product Backlog* para a *Sprint*. O *Sprint Backlog* reúne os itens que serão alvo da próxima *Sprint* (SCHWABER; SUTHERLAND, 2011).

A grande diferença entre o *Product Backlog* e o *Sprint Backlog* é que o primeiro aborda todos os itens que serão produzidos ao longo de todas as *Sprints*, e o segundo aborda os itens existentes no *Product Backlog* que serão trabalhados na *Sprint* imediata.

Segundo Schuaber e Sutherland (2011) a equipe de desenvolvimento altera o *Sprint Backlog* durante toda a *Sprint*, sempre que surge um novo trabalho a ser feito, a equipe adiciona o novo item ao *Sprint Backlog*. Conforme o trabalho vai evoluindo, é recomendável fazer-se uma estimativa do trabalho restante. É importante lembrar que somente a equipe do projeto pode alterar o *Sprint Backlog* durante a *Sprint*.

2.7.3 Time-Boxes

Time-Boxes são os eventos que compõe o *Scrum*, Segundo (ROSA et al, 2011) são compreendidos entre *Release Planning Meeting*, *Sprint Planning Meeting*, *Sprint*, *Daily Scrum*, *Sprint Review Meeting* e *Sprint Retrospective*.

A *Daily Scrum* é a rápida reunião em pé, seu objetivo é definir o que deve ser feito no dia em questão, além de uma breve análise sobre o que foi feito no dia anterior, isso abrange o que funcionou corretamente e os obstáculos encontrados. Segundo (RISING;JANOFF, 2000, p. 26), “Nesta etapa três perguntas devem ser respondidas por cada membro sobre suas responsabilidades: O que foi feito ontem? O que será feito hoje? Há algum obstáculo à realização das atividades?”.

É importante salientar que esta última pergunta reflete um dos principais princípios do *Scrum*, os obstáculos à realização das atividades tais como *hardware* ou incompatibilidades de programas devem ser superadas pelo *Scrum Master*, que será melhor estudado no decorrer deste trabalho. Em linhas gerais, este membro deve resolver problemas diversos para possibilitar aos desenvolvedores que se preocupem exclusivamente em programar (SCHWABER; SUTHERLAND, 2011).

De acordo com Medeiros (2009, p.72):

Apesar desta reunião ser breve, é de grande importância para o time, pois através das três perguntas muito simples, o time irá inspecionar e adaptar de maneira contínua seu ritmo e principalmente os problemas que precisam ser removidos para que a meta da *Sprint* seja alcançada.

As reuniões são objeto do capítulo Reuniões, que será visto mais adiante neste trabalho.

A *Release Planning Meeting* é a reunião de planejamento de uma parte do produto, que será desenvolvido durante a *Sprint*, ela tem o objetivo de planejar o trabalho e definir as metas para o desenvolvimento do incremento objeto da *Sprint* (ROSA et al, 2011).

Segundo (ROSA et al, 2011) a *Sprint Planning Meeting* é a reunião de planejamento da *Sprint* e tem como foco definir e estimar a dificuldade das atividades durante a *Sprint*.

A mesma autora define a *Sprint Review Meeting* como sendo a reunião que ocorre ao fim da execução da *Sprint*, tendo como objetivo a avaliação das metas da *Sprint* pelo *Product Owner* e uma possível obtenção de *feedback* para as *Sprints* posteriores.

A *Sprint Retrospective* é a reunião de retrospectiva da *Sprint* e tem como objetivo fazer a equipe perceber o que funcionou e o que não funcionou durante a *Sprint* (ROSA et al, 2011).

2.7.4 Reuniões

As reuniões, também conhecidas como *Time Boxes*, são parte fundamental da metodologia ágil *Scrum*, por esse motivo as reuniões vistas de forma superficial anteriormente no capítulo referencial teórico, agora recebem uma abordagem mais específica e minuciosa. Conforme pesquisas realizadas em dissertações e livros que abordam o assunto, serão apresentados a seguir os tipos de reuniões compreendidos em *Release Planning Meeting*, *Sprint Planning Meeting*, *Daily Scrum Meeting*, *Sprint Review Meeting*, *Sprint Retrospective*.

2.7.4.1 Release Planning Meeting

Trata-se da Reunião de Planejamento da Versão para Entrega. É uma reunião que tem como objetivo planejar o trabalho e definir metas para a produção de um protótipo funcional, com os requisitos de maior importância no *Product Backlog* e com data prevista de entrega. Este planejamento de protótipo pode receber um *update* ao final de cada *Sprint*. Com essa reunião é viável obter uma visão futura do projeto e dos *Sprints* que farão parte do próximo lançamento (SCHWABER; SUTHERLAND, 2011).

2.7.4.2 Sprint Planning Meeting

Trata-se da Reunião de Planejamento da *Sprint*, que reúne a equipe, o *Product Owner* e o *Scrum Master* para definir e estimar a dificuldade e duração das atividades que serão entregues ao final do ciclo da *Sprint* (ROSA et al, 2011).

Além da equipe, outras pessoas podem participar, desde que venham a somar valores à reunião e tenham permissão aprovada pelo *Product Owner*.

Esta reunião é dividida em duas partes, ambas com a mesma duração, na primeira é decidido o que será feito na *Sprint*. O *Product Owner* definirá a meta do *Sprint* e comunicará à equipe os requisitos mais importantes no *Product Backlog*, além disso, a equipe deverá estimar os itens em tamanhos e selecionar os que demonstram maior viabilidade de serem

feitos durante a *Sprint*, tendo como produto uma lista chamada de *Selected Product Backlog*. Na segunda parte da reunião, a equipe deverá colher mais informações sobre os itens da lista, para transformá-la em um incremento pronto e obtendo assim, o *Sprint Backlog*.

Vale ressaltar que só deve estar na *Selected Product Backlog* itens que a equipe acredita poder terminar dentro da *Sprint*. Caso haja itens que sejam tão grandes a ponto de ser impossível terminá-los na *Sprint*, a equipe e o *Product Owner* podem quebrá-los em partes menores e enquadrá-los nos próximos *Sprints* (KNIBERG, 2007).

Segundo (KNIBERG, 2007) a *Sprint Planning Meeting* é uma das reuniões mais importantes do *Scrum* pois oferece a equipe informações básicas para o trabalho durante a *Sprint* e proporciona confiança ao *Product Owner* que com isso não incomoda a equipe durante a *Sprint*, porém, por se tratar de uma reunião crítica, se o planejamento não for adequado, o *Sprint* poderá ser comprometido.

2.7.4.3 Daily Scrum Meeting

Segundo (ROSA et al, 2011) é uma reunião diária em que cada membro da equipe informa o que realizou desde a última reunião diária, o que pretende fazer antes da próxima reunião diária e se ele está tendo algum impedimento em suas atividades.

A *Daily Scrum Meeting* também é conhecida como *Stand Up Meeting* (Reunião em pé), pois para que seja objetiva e eficaz, os membros da equipe devem permanecer em pé. Vale ressaltar que esta reunião deve sempre se realizar na mesma hora e no mesmo lugar (KNIBERG, 2007).

É fundamental que o *Scrum Master* garanta que esta reunião ocorra diariamente, pois ela proporciona a equipe que o conhecimento acerca do projeto e a comunicação melhorem a cada dia, evitando assim a necessidade de reuniões de ajustamento posteriores e propiciando a tomada rápida de decisões.

2.7.4.4 Sprint Review Meeting

É a reunião de revisão, que ocorre sempre quando acaba a execução da *Sprint*. O foco dessa reunião é a equipe apresentar as funcionalidades produzidas para o *Product Owner* e para convidados. O *Product Owner* avalia se o objetivo da *Sprint* foi alcançado e produz anotações que podem transformar-se em novos itens para o *Product Backlog* (SCHWABER; SUTHERLAND, 2011).

Têm-se como benefícios da realização adequada desta reunião, a obtenção de crédito à equipe, fazendo os integrantes se sentirem confiantes pelo trabalho feito. Vale lembrar que,

mesmo que o *Sprint* não obtenha sucesso, é recomendável realizar esta reunião, pois ela servirá como base de aprendizado para o próximo *Sprint*, onde a equipe terá mais informações para realizar a entrega das funcionalidades selecionadas.

2.7.4.5 *Sprint Retrospective*

É a retrospectiva da *Sprint*, última reunião do fluxo do *Scrum*, onde somente a equipe deverá participar, além do *Product Owner*, se todos os integrantes da equipe concordarem (SCHWABER; SUTHERLAND, 2011).

Nesta reunião o *Scrum Master* instiga a equipe a identificar os itens que foram produzidos com êxito e os que não obtiveram sucesso no último *Sprint*. O objetivo é, ao final da reunião, a equipe ter identificado maneiras para melhorar os itens que tiveram insucesso no último *Sprint* e que serão utilizadas no próximo *Sprint*.

A duração das reuniões pode variar de acordo com a duração da *Sprint*, que por padrão dura entre 2 e 4 semanas . A figura 3 demonstra uma tabela que considera uma *Sprint* de 4 semanas, e com base nesta *Sprint* define-se o tamanho das *Time Boxes*.

Figura 3 – Duração dos *Time-Boxes*

Evento	Duração	
	<i>Sprint</i> de X tempo	<i>Sprint</i> de 4 semanas
<i>Release Planning Meeting</i>	15 à 20% de um plano tradicional	
<i>Sprint Planning Meeting</i>	5% de X	8 horas
<i>Sprint</i>	2 a 4 semanas	
<i>Daily Scrum Meeting</i>	15 minutos	
<i>Sprint Review Meeting</i>	Até 5% de X	4 horas
<i>Sprint Retrospective</i>	3 horas	

Fonte: (ROSA et al, 2011).

2.7.5 Papéis

Os papéis definem os atores do processo *Scrum*, sendo eles: *Product Owner*, *Scrum Master* e a equipe de desenvolvimento. Estes papéis são abordados a seguir.

O *Product Owner* é segundo (MEDEIROS, 2009, p. 68):

O responsável por garantir o retorno sobre o investimento do projeto. Também conhece as necessidades dos clientes e usuários. É uma parte fundamental durante todo o projeto, pois além de fornecer a visão priorizada de desejos, é extremamente vital para validar as entregas feitas pela equipe e fornecer os principais insumos para a melhoria contínua da forma de trabalho da equipe. Ele representa o cliente e geralmente define o grau de importância e prioridade que os requisitos possuem.

Vale ressaltar que ele é a única pessoa responsável pela gerência do *Product Backlog*, tal gerenciamento inclui: declaração clara dos itens do *Product Backlog*; ordenação dos itens do *Product Backlog*; garantia de visibilidade e entendimento do *Product Backlog* para todos os membros do time *Scrum* (SCHWABER; SUTHERLAND, 2011).

O *Product Owner* pode delegar as tarefas citadas para a equipe, porém é ele o responsável por elas. O *Product Owner* deve ter suas decisões respeitadas pela equipe, visto que ele representa o cliente

O *Scrum Master* é comumente chamado de líder ou facilitador, ele garante a adequada utilização de *Scrum* pelo time. Entre suas funções, estão facilitar o trabalho dos programadores, eliminando problemas adicionais, para que eles somente se preocupem em desenvolver a aplicação em si. (SCHWABER; SUTHERLAND, 2011).

O *Scrum Master* é conhecido como “servo-líder” do time *Scrum*, pois embora lidere a equipe, ele se submete as decisões do *Product Owner*, ele é responsável por ajudar os que estão fora do time *Scrum* a interagirem da forma correta com a equipe *Scrum*.

Segundo (SCHWABER; SUTHERLAND, 2011), a equipe de desenvolvimento é formada por colaboradores que produzem versões do produto. Somente os colaboradores da equipe podem criar incrementos do produto.

As equipes de desenvolvimento são auto-gerenciáveis, ou seja, possuem permissão da organização para exercer sua própria gerência interna. Elas são auto-organizadas e multifuncionais, visto que organizam a forma como produziram os incrementos e, nem o *Scrum Master* deve interferir, elas possuem todas as habilidades necessárias para criar incrementos do produto (SCHWABER; SUTHERLAND, 2011).

O único título a ser dado para os integrantes da equipe de desenvolvimento é o de desenvolvedor, o *Scrum* não reconhece nenhum outro título para integrantes da equipe. Obviamente a equipe possui diferentes integrantes com habilidades diversas, mas as responsabilidades das tarefas pertencem a toda a equipe. Não deve haver subdivisões da equipe dedicadas a tarefas específicas.

O tamanho da equipe de desenvolvimento deve ser pequeno o bastante para manter-se ágil, e grande o bastante para ser capaz de realizar as tarefas das *Sprints*. Recomenda-se que a equipe de desenvolvimento possua no mínimo três e o máximo nove integrantes, vale lembrar que os papéis *Scrum Master* e *Product owner* não devem ser considerados integrantes da equipe, a não ser que eles, além de seus papéis, também desempenhem atividades de incremento dentro das *Sprints* (SCHWABER; SUTHERLAND, 2011).

2.7.6 Método

Um método chamado *Planning Poker* auxilia na otimização das tarefas do *Scrum*, ele auxilia na estimativa de tempo dos requisitos que serão feitos durante o projeto, principalmente em projetos complexos em que há discrepância entre as estimativas individuais de cada integrante da equipe (KNIBERG, 2007).

Nesse método a equipe se reúne e cada membro recebe treze cartas, com números diversos, sempre que uma história deve ser estimada, cada membro da equipe escolhe uma carta (que faz referência a um número) e a vira para baixo na mesa, quando todos escolherem uma carta, as cartas são viradas e se observa se há similaridade entre os números, caso haja discrepâncias significativas, o requisito é discutido e ocorre uma nova rodada, até que haja similaridade entre os valores (KNIBERG, 2007).

O importante nesse método é que todos os integrantes da equipe são forçados a pensar por si mesmos na estimativa, sem qualquer influência dos outros membros, tornando as estimativas o mais fiel possível com a realidade (KNIBERG, 2007). Este processo é exibido na figura 4.

Figura4 – Planning Poker(método de estimativa de tempo)



Fonte: (KNIBERG, 2007).

2.7.7 Sistemas que auxiliam o uso de Scrum

Existem sistemas que facilitam a utilização de *Scrum*, podem ser citados *Scrum-Box*, *Scrum Me* entre outros, para conhecermos mais a fundo alguns a seguir vamos abordar o sistema *Scrum Me* e o *Scrum-Box*, o primeiro por ser largamente utilizado e conhecido e o segundo por ser fruto de uma produção de (ROSA et al, 2011) que é abordada neste trabalho.

O *Scrum Me* é um sistema *web* gratuito que auxilia na organização de projetos baseados em *Scrum*. Com base na criação de uma conta no *Scrum Me*, a seguir é demonstrado como funciona o sistema.

A primeira tela que o usuário tem contato é a tela de cadastro, que é muito simples, consistindo em informar um *e-mail* válido, criação e confirmação de uma senha, como é demonstrado nas figuras 5 e 6.

Figura 5 – Tela de registro do Scrum Me



Fonte: (SCRUM ME, 2012).

Figura 6 – Campos de inscrição do Scrum Me

Fonte: (SCRUM ME, 2012).

Após a criação da conta o usuário é redirecionado para a *HomePage* do Scrum Me e pode criar um novo projeto, como é apresentado nas figuras 7 e 8.

Figura 7 – HomePage do Scrum Me

Fonte: (SCRUM ME, 2012).

Figura 8 – Tela de criação de novo projeto do *Scrum Me*

Fonte: (SCRUM ME, 2012).

Pode se perceber que é possível definir o tempo de duração previsto para o projeto, podendo ser por horas ou pontos, dias ou horas/pontos.

Com o projeto criado, o próximo passo a ser dado é criar a primeira *Sprint* do projeto, como é exibido na figura 9.

Figura 9 – Tela de criação de nova *Sprint*.

Fonte: (SCRUM ME, 2012).

Após a criação da *Sprint*, é possível gerenciar, criar e reajustar requisitos, verificar o progresso da *Sprint*, como é exposto na figura 10.

Figura 10 – Tela de controle das *Sprints* criadas

Fonte: (SCRUM ME, 2012).

O *Scrum Me* pode ser uma ferramenta auxiliadora no controle de projetos baseados em *Scrum*, por ser fácil de ser utilizada, acaba fornecendo uma percepção mais ampla do andamento do projeto, além de facilitar alterações necessárias. Vale ressaltar que o *Scrum Me* é uma ferramenta auxiliar, de organização do projeto e não substitui nenhum método de utilização da metodologia *Scrum*, já que (KNIBERG, 2007) lembra que a simplicidade deve fazer parte de todas as etapas do projeto.

O *Scrum-Box* é um sistema de gestão de projetos baseados em *Scrum* elaborado por alunos do curso de Sistemas da Informação da faculdade Anhembí Morumbi como projeto de conclusão do curso.

A primeira tela que o usuário tem contato no sistema *Scrum-Box* é a tela de *login* como demonstrado pela figura 11.

Figura 11 – Tela de *login* do sistema *Scrum-Box*.

Fonte: ROSA (et al, 2011).

Após efetuar o *login* uma das opções do usuário é gerenciar os colaboradores, compreendendo a lista de funcionários e a possibilidade de inserção de novos membros, como é apresentado na figura 12.

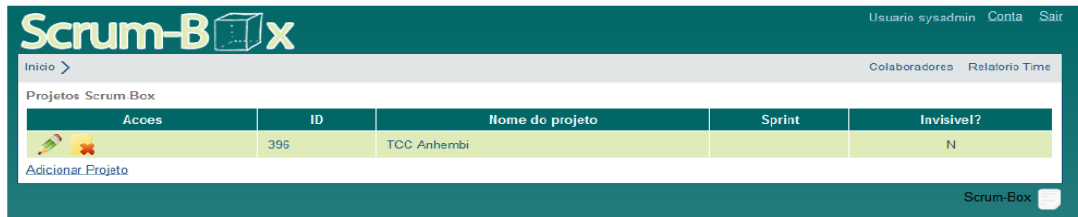
Figura 12 – Tela de gerenciamento de colaboradores do sistema *Scrum-Box*.

Acoes	Usuario	Nome	Iniciais	Telefone	Email
	meg	Magda	ms		ms@gmail.com
	naty	Naty	nm	119999-0000	nathy@gmail.com
	will	Willian	Wf		will@gmail.com
	sysadmin	sysadmin	SYS		no@reply.com

Fonte: ROSA (et al, 2011).

Com o sistema *Scrum-Box* também é possível gerenciar os projetos nele cadastrados que são dispostos em uma lista, como na figura de exemplo 13.

Figura 13 – Tela de gerenciamento de projetos do sistema *Scrum-Box*.



Fonte: ROSA (et al, 2011).

A criação de um novo projeto é possível através da figura 14.

Figura 14 – Tela de criação de um novo projeto do sistema *Scrum-Box*.

Fonte: ROSA (et al, 2011).

Com projetos existentes é possível criar *Sprints*, de forma objetiva como é exposto na figura 15.

Figura 15 – Tela de criação de *Sprints* do sistema *Scrum-Box*.

Fonte: ROSA (et al, 2011).

As *Sprints* existentes podem ser gerenciadas, sendo possível editar, excluir e adicionar novas *Sprints*, como é apresentado na figura 16.

Figura 16 – Tela de criação de *Sprints* do sistema *Scrum-Box*.



Fonte: ROSA (et al, 2011).

Após a observação e utilização dos sistemas citados, algumas considerações podem ser apontadas.

Ambos os sistemas possibilitam uma organização prática em projetos baseados em *Scrum*, facilitando ações como organizar projetos, *Sprints*, gerenciar o andamento e evolução das produções de *software*.

Porém pôde-se observar através da utilização, que o *Scrum Me* oferece mais praticidade e objetividade nas tarefas de gerenciamento de projetos *Scrum*.

2.8 METODOLOGIA

A pesquisa em questão se enquadra no método comparativo, pois compreende as práticas operacionais que consagram este método (UNISC, 2012), tais práticas são: indicação de uma hipótese; coleta de dados; análise da resposta.

2.8.1 Método de abordagem

Enquadra-se como método hipotético-dedutivo, pois parte da hipótese de que o *Scrum* venha a solucionar o problema da falta de metodologia na produção de *softwares* por pequenas empresas, e após uma profunda análise e aplicação dessa metodologia, será feita uma dedução sobre a real eficácia, ou não, do processo em questão.

O método hipotético-dedutivo consiste na eleição de hipóteses (proposições hipotéticas), as quais possuem certa viabilidade para responder a um determinado problema de natureza científica (UNISC, 2012).

2.8.2 Método de procedimento

A presente pesquisa se enquadra no método comparativo, uma vez que, segundo a UNISC (2012): consiste no confronto entre elementos, levando em consideração seus atributos e promove o exame dos dados a fim de obter diferenças ou semelhanças que possam ser constatadas e as devidas relações entre estes elementos.

2.8.3 Aplicação e resultados do questionário

Foi realizado um questionário com uma profissional de uma empresa de Passo Fundo no ramo de desenvolvimento de software, que faz parte de uma equipe *Scrum*. Neste questionário, alguns resultados foram constatados, os quais serão abordados a seguir.

A principal justificativa para utilizar o *Scrum* no desenvolvimento de seus produtos se dá porque o processo prioriza a comunicação e a visibilidade do que é importante para o bom andamento do projeto, bem como também dá mais agilidade no desenvolvimento do *software*, tal fato tem base segundo (KNIBERG, 2007) :”... as práticas *Scrum* ajudam a identificar (e algumas vezes evitar) erros comuns ...”.

Segundo dados do questionário, a utilização de *Scrum* deve-se iniciar com uma orientação para a equipe de como funciona a metodologia, para se definir o líder da equipe, é levado em consideração a experiência e os casos de sucesso que essa pessoa já teve liderando uma equipe. Segundo a profissional, nem todos os passos de *Scrum* são seguidos a risca e alguns são agrupados, sempre de acordo com a necessidade do projeto, são exemplo disso a definição do *Sprint Backlog* (definição do quadro de tarefas) no decorrer do *Daily*, também é um exemplo disso a opção da equipe em não utilizar a programação conjunta, tal estratégia é prevista por Kniberg (2007):”... *Scrum* não vai te dizer exatamente o que fazer...”, “... não significa que você deva fazê-lo exatamente da mesma maneira. De fato eu também faria de maneira diferente, se encontrasse situações diferentes.”

Foi citado como característica a heterogeneidade da equipe, tanto que alguns colaboradores trabalham remotamente, de outras cidades, inclusive o gerente do projeto. Sendo assim, as reuniões são realizadas à tarde, com duração de até uma hora, sempre via sala de conferência. Uma vez por semana, ocorre uma reunião de uma hora com o cliente, para alinhar o projeto e suas pendências, tanto da equipe quanto do cliente, nesta participam, além do cliente, os gerentes, os líderes técnicos e os analistas (de testes e funcional).

O método *Scrum* melhorou a comunicação e a colaboração da equipe e, por consequência, o retrabalho diminuiu consideravelmente. Quanto à motivação da equipe, a colaboradora acredita que a visibilidade que todos têm das atividades em andamento faz crescer o comprometimento com as mesmas.

Um dificultador encontrado foi uma confusão quanto à ligação do *Scrum* com a documentação do produto, por não estar formalmente presente na metodologia. Como consequência desta falta de documentação, o projeto já atingiu o dobro de horas que haviam sido estimadas. Também há o problema de a metodologia não prever os riscos e as

possibilidades de insucesso do projeto. Segundo a colaboradora, o *Scrum* não faz “milagre”, apenas ajuda a conduzir a equipe de maneira ágil.

Atividades e decisões que precisam ser inseridas na *Sprint*, para solucionar um imprevisto, são avaliadas e estimadas pelo líder e pelo responsável da atividade.

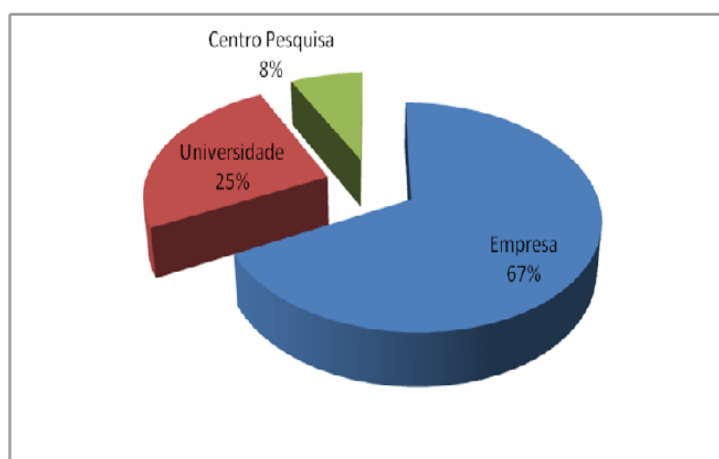
2.9 TRABALHOS RELACIONADOS

Nesta seção serão abordados trabalhos acadêmicos que tem como foco o tema *Scrum*.

Uma Pesquisa sobre a quantidade de material acadêmico abordando *Scrum*¹ foi realizada por (CARVALHO e MELLO, 2009) e neste trabalho teórico-conceitual quantitativo foi elaborada uma coleta de trabalhos que tratam do tema *Scrum* no meio acadêmico para constatar o quanto este tema já foi estudado, tendo por motivação o fato de *Scrum* ser um tema relativamente novo (foi inventado na década de 1990).

O resultado da pesquisa mostrou que o tema *Scrum* é muito abordado na indústria, porém pouco observado no meio acadêmico, tal fato foi concluído devido a filiação dos autores que abordaram o tema, sua grande maioria está no meio industrial e não no meio acadêmico, como pode ser observado na figura 17 (CARVALHO e MELLO, 2009).

Figura 17 – Distribuição das publicações por filiação dos autores



Fonte: Carvalho e Mello, 2009.

A figura 17 apresenta a relação das publicações sobre a metodologia *Scrum* e o meio em que foram fomentadas. Percebe-se que o meio empresarial representa mais da metade dos trabalhos sobre o tema.

Isso pode ser explicado pelo fato de *Scrum* ter sido criado na indústria de *softwares* e não em uma universidade (CARVALHO e MELLO, 2009).

A utilização de *Scrum* em projetos de desenvolvimento distribuído de *software*² é objeto do trabalho de Prado (et al, 2013) tentando solucionar os problemas de implementar *Scrum* em projetos desse tipo.

A comparação entre projetos anteriores que tentaram utilizar *Scrum* norteiam as soluções propostas. As seguintes soluções foram formuladas:

- É recomendado o aumento do número de reuniões presenciais para diminuir os problemas decorrentes da distância física (PRADO et al. 2013);
- A estimativa de tempo das atividades é melhorada quando utilizado o método *Planning Poker* e para sanar a dificuldade da distância física recomenda-se utilizar ferramentas instantâneas de comunicação (PRADO et al. 2013);
- Distribuir tarefas de acordo com o perfil e experiência de cada integrante da equipe sana o problema da falta de domínio da plataforma, para apurar este método recomenda-se aplicar um questionário para verificar habilidades e experiências individuais (PRADO et al. 2013);
- Utilizar largamente a audioconferência e o *e-mail*, pois são as mais eficientes ferramentas de comunicação existentes (PRADO et al. 2013);
- Substituição das reuniões diárias de quinze minutos, por reuniões semanais de quarenta minutos visto que devido a distância física é inviável realizar rigorosamente as reuniões diárias (PRADO et al. 2013);
- Recomenda-se agendar um dia da semana para integrar os vários artefatos gerados pela equipe geograficamente distribuída, isso torna a integração mais efetiva e segura do que o simples envio das atividades prontas por algum meio de comunicação, evitando o conflito de versões do sistema e o conseqüente gasto desnecessário de tempo (PRADO et al. 2013).

A elaboração de um sistema de gestão de projetos baseados em *Scrum*³ foi realizado por (ROSA et al, 2011).

Trata-se de um projeto de desenvolvimento de uma aplicação *web* que gerencia a utilização das práticas *Scrum* na produção de um *software*.

O trabalho foca em facilitar a utilização e organização do *Scrum*, ajudando a diminuir riscos e evitando problemas enfrentados principalmente por empresas que utilizam planilhas para gerenciar os seus projetos. O sistema produzido no trabalho ainda dispõe de uma funcionalidade de geração de relatórios para facilitar a percepção do trabalho já feito e o trabalho que ainda está por ser feito (ROSA et al, 2011).

A aplicação de *Scrum* no desenvolvimento de produtos em uma empresa foi realizada por (CARVALHO e MELLO, 2010).

Este trabalho apresenta o resultado de uma pesquisa-ação feita em uma pequena organização de base tecnológica na qual se aplicou *Scrum* em um projeto de desenvolvimento de um *software*. O principal foco deste trabalho é analisar a aplicação do método *Scrum* no desenvolvimento de um *software* nesta pequena empresa de base tecnológica, além de compreender e mensurar o impacto dessa aplicação na empresa (CARVALHO e MELLO, 2010).

Entre as contribuições obtidas neste trabalho de pesquisa está a amostragem científica de como empresas de base tecnológica são impactadas pela implantação de *Scrum* em seus projetos (CARVALHO e MELLO, 2010).

Alguns benefícios foram percebidos pela equipe que utilizou o método *Scrum*, entre eles a melhoria na comunicação e aumento da colaboração entre envolvidos, aumento da motivação da equipe de desenvolvimento, diminuição perceptível nos prazos estipulados, diminuição da possibilidade de insucesso do projeto (riscos), entre outros (CARVALHO e MELLO, 2010).

Além destas vantagens outros aspectos positivos puderam ser percebidos, como a diminuição dos custos de produção e aumento da produtividade da equipe (CARVALHO e MELLO, 2010).

2.10 SIMULAÇÃO DE SCRUM EM PARTE DE UM PROJETO DE UMA EMPRESA

Em uma empresa de publicidade, produtora de sistemas *web* foi realizado a simulação dos princípios *Scrum* em um ciclo (*Sprint*) de um projeto.

O gerente geral da empresa relatou já ter utilizado *Scrum* em projetos anteriores, mas que depois a metodologia foi abandonada por métodos que a própria equipe criou para organizar suas produções de *software*.

Após uma conversa sobre o método ágil *Scrum*, foi estabelecida uma equipe de três desenvolvedores, além disso foi escolhido por consenso o *Product Owner*, que já demonstrou em linhas gerais, o que o cliente desejava do sistema, nesta mesma conversa também foi definido o *Scrum Master* que acumulou essa função com a de desenvolvedor na equipe de desenvolvimento.

A produção do *Product Backlog* transcorreu sem maiores dificuldades e como se tratava da aplicação de *Scrum* em um ciclo do projeto, a quantidade de requisitos era reduzida, após a definição do *Product Backlog* a seguinte lista foi obtida conforme a figura 18:

Figura 18 – Product Backlog

Id	Nome	Imp	Est	Como demonstrar	Notas
1	Cadastro de atendimento	30	10	O cadastro deve ser feito no módulo de prospecção .	Cada atendimento cadastrado deve gerar um email para o gerente.
2	Listar atendimentos	20	8	Cada atendimento deve ser incluso na lista e um email deve ser enviado para o gerente.	Cada atendimento anexado a lista, deve ser enviado por email para o gerente.
3	Relatórios/Filtros	10	2	Os relatórios/filtros devem ser por período/data, corretor ou tipo de imóvel.	

Fonte: Elaborada pelo autor, 2014.

O projeto objeto da aplicação de *Scrum* é um sistema para uma empresa corretora de imóveis, como demonstrado na figura 18, os requisitos definidos no *Product Backlog* são: o cadastro de um atendimento, a listagem destes atendimentos, relatórios e filtros.

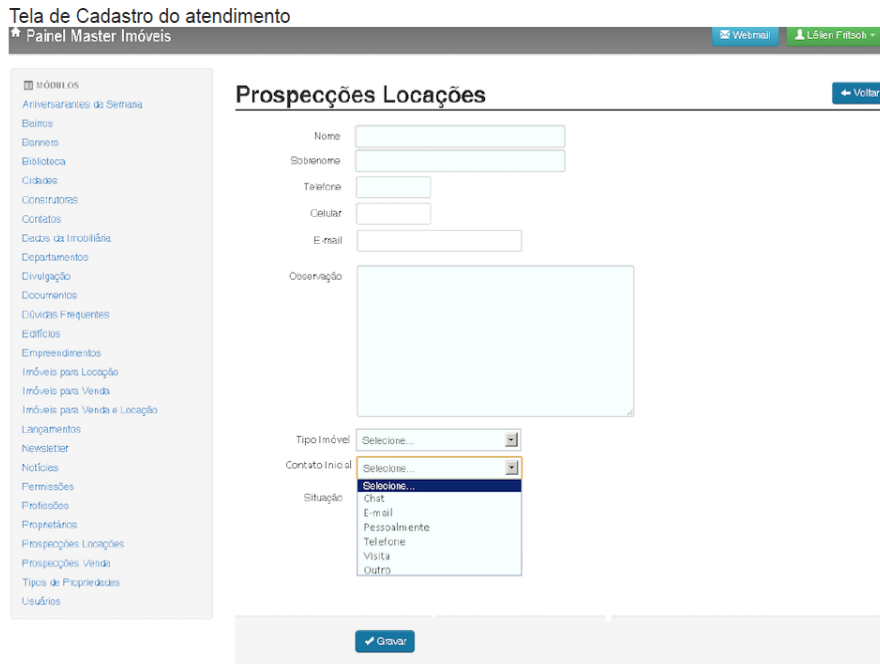
O colaborador deve cadastrar o seu atendimento na área administrativa da aplicação, além do cadastro é necessário armazenar o histórico de cada intervenção no atendimento do colaborador. A cada intervenção no atendimento, o sistema deve disparar uma mensagem sete dias após a data da intervenção, com um lembrete de que seu atendimento ainda está em aberto.

Por se tratar de um ciclo do projeto, ao qual foi aplicado *Scrum* e a quantidade de requisitos ser reduzida, o *Product Backlog* foi também estabelecido como *Sprint Backlog* após a reunião de planejamento da *Sprint*, tal atitude encontra respaldo segundo (KNIBERG, 2007): “O melhor e o pior do *Scrum* é que você é forçado a adaptar o processo para sua situação específica”

A *Sprint* durou uma semana, na qual os três desenvolvedores criaram os incrementos que foram incluídos no sistema após prévia apresentação e aprovação pelo *Product Owner*.

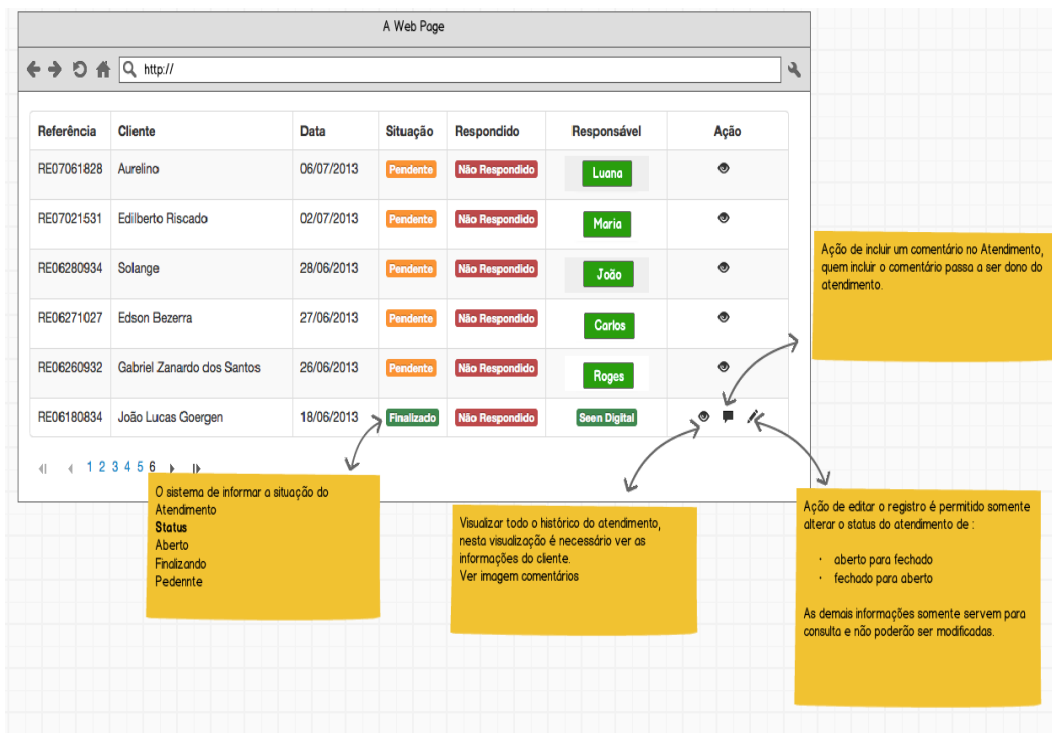
As figuras 19, 20 e 21 apresentam as telas dos incrementos prontos:

Figura 19 – Tela de cadastro do atendimento.

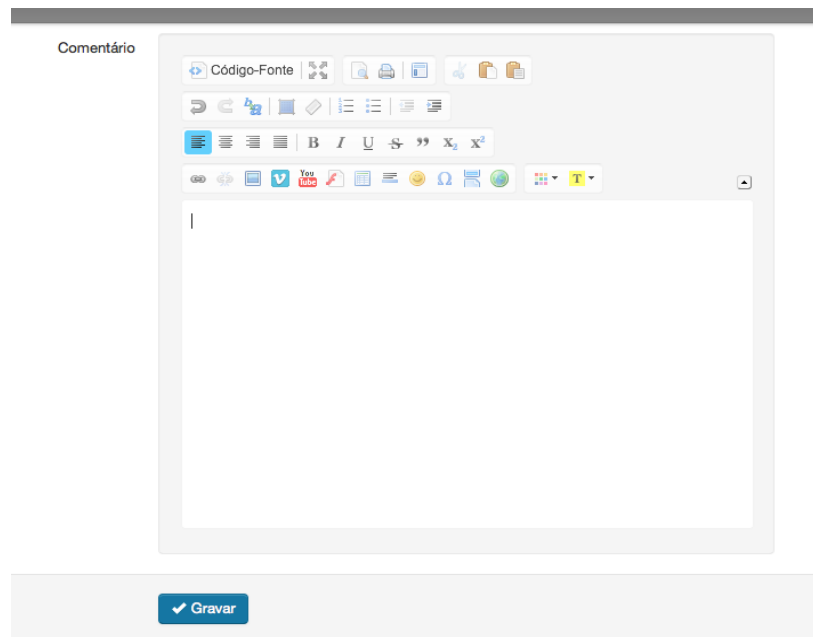


Fonte: Elaborada pelo autor, 2014.

Figura 20 – Tela de listagem dos atendimentos.



Fonte: Elaborada pelo autor, 2014.

Figura 21 – Tela de relatórios/filtros dos atendimentos.

Fonte: Elaborada pelo autor, 2014.

3 CONSIDERAÇÕES FINAIS

Com base na análise bibliográfica, no questionário aplicado a uma funcionária de uma empresa de Passo Fundo que usa *Scrum* e na simulação dentro de uma empresa, pôde-se perceber diante das declarações que a metodologia ágil *Scrum* agiliza o processo de produção de *softwares*, aprimora a análise e seleção de requisitos e diminui o desperdício de tempo em reuniões mal planejadas.

Outros pontos positivos percebidos são a melhora da comunicação e interação da equipe, a confiança do cliente no projeto, definição das responsabilidades de todos os envolvidos no projeto, facilidade em realizar alterações não previstas no projeto inicial e principalmente a diminuição do desperdício financeiro com projetos que se prolongam indefinidamente.

Um empecilho encontrado durante o presente trabalho foi a dificuldade de se inserir nas empresas. Várias foram contatadas por *e-mail*, telefone e pessoalmente na tentativa de ter espaço para testar a metodologia, porém nenhuma aceitou a proposta, alegando não ter interesse, já utilizar outra metodologia ou já ter utilizado *Scrum* no passado. A empresa em que foi realizada a simulação de ciclos de *Scrum*, aceitou esta simulação por ter interesse em auxiliar e manter contato com o meio acadêmico, mas relatou não ter interesse em utilizar *Scrum* pois já haviam utilizado em projetos anteriores e não se adaptaram a metodologia, preferindo utilizar seus próprios métodos de organização de projetos.

Um problema identificado na literatura, no questionário e na experiência dentro de uma empresa, é a falta de documentação contínua em todas as fases do projeto, tal problema ocorre por que como *Scrum* é uma metodologia ágil, que foca nos resultados, ele não prima por fatores que são importantes em metodologias tradicionais, como a documentação em Cascata por exemplo.

A falta de documentação é uma preocupação que pode ser objeto de um trabalho futuro, mesclando a metodologia com técnicas de documentação que são tradicionais em outras metodologias.

Como parte das considerações finais, criou-se uma lista de recomendações para a implementação de *Scrum* em projetos de desenvolvimento de *softwares*, sendo esta lista uma das contribuições do presente trabalho.

3.1 RECOMENDAÇÕES PARA A IMPLEMENTAÇÃO DE SCRUM

Com base no referencial teórico estudado e no questionário que foi aplicado no presente trabalho, seguirá uma lista de recomendações para a implementação de *Scrum* em uma empresa.

A primeira etapa a ser executada é uma reunião com toda a equipe envolvida no projeto e passar uma espécie de tutorial sobre *Scrum*, instruindo a equipe de como funciona a metodologia, como lembra (KNIBERG, 2007):

“As equipes precisam conhecer o básico do *Scrum*. Como você cria e estima um *product backlog*? Como você o transforma num *sprint backlog*? O livro do Henrik é um *kit* para iniciantes com práticas básicas que ajudam equipes a irem além de apenas tentativas de praticar o *Scrum* para executá-lo bem.”

Em um segundo momento, logo na segunda reunião deve ocorrer a definição da lista de requisitos (*Product Backlog*), com itens, histórias, em resumo os detalhes que o cliente deseja utilizando a linguagem do cliente, estes itens devem ter um ID para que não se perca o controle do item uma vez que ele costuma mudar de nome durante o processo. Também recomenda-se dar uma estimativa inicial sobre o tempo e pessoal envolvidos na atividade. Juntamente com a definição das tarefas deve-se também definir o *Scrum Master*, o líder da equipe, recomenda-se que este líder seja a pessoa com mais experiência e *cases* de sucesso, também nessa reunião é necessário definir o colaborador que será o *Product Owner* que representará o cliente. Recomenda-se simplicidade tanto nas ferramentas como nos métodos utilizados pela equipe (KNIBERG, 2007).

Com os passos iniciais feitos, pode-se realizar a reunião que definirá a lista de itens retirados do *Product Backlog*, que serão produzidos durante a *Sprint*, esta lista é conhecida como *Sprint Backlog* e para a sua elaboração a equipe deve discutir os itens que podem ser realizados na *Sprint*, vale lembrar que itens muito grandes, que demandaram muito tempo para serem feitos e que extrapolem o período de uma *Sprint*, que por padrão é de duas a quatro semanas, podem ser quebrados em partes menores e distribuídos em outras *Sprints*. Nessa reunião é importante definir um *Sprint Backlog* consistente, evitando assim, interferências do *Product Owner* durante a *Sprint* e permitindo a equipe trabalhar com confiança e tranquilidade durante a *Sprint*.

Durante a *Sprint* a equipe deve focar na produção dos itens selecionados que fazem parte da *Sprint Backlog*, qualquer interferência ou dificuldade encontrada deve ser sanada pelo *Scrum Master* para que os membros da equipe não fiquem impedidos de realizarem suas

atividades e com isso, comprometam o prazo estipulado para entregar os itens prontos (KNIBERG, 2007).

Segundo (SCHWABER; SUTHERLAND, 2011) é durante a *Sprint* que deve ocorrer a *Daily Scrum Meeting* ou *Stand Up Meeting*, que são pequenas reuniões diárias que os integrantes da equipe participam, mantendo-se em pé, com duração de quinze minutos, em que todos relatam o que foi realizado desde a última reunião diária, os problemas que encontraram e o que será realizado até a próxima reunião diária.

Ao final da *Sprint* deve ser realizada a reunião de revisão da *Sprint*, com a finalidade de avaliar o incremento e adaptar o *Product Backlog* se for preciso. Durante essa reunião, a equipe e demais interessados conversam sobre os itens desenvolvidos na *Sprint* e com base nisso idealizam o que será realizado a seguir. Vale ressaltar que esta é uma reunião informal e seus objetivos principais são motivar, obter comentários e promover a colaboração.

Tendo como base uma *Sprint* de quatro semanas, a reunião de revisão da *Sprint* deve ter quatro horas de duração. Esta reunião deve reunir alguns elementos: a identificação pelo *Product Owner* do que está e do que não está pronto; realização de um balanço pela equipe, do que foi realizado durante a *Sprint*; *feedback* da equipe sobre os incrementos realizados; o *Product Owner* avalia a evolução do *Product Backlog* e a equipe estabelece o que será feito a seguir. Nota-se que, o que se obtêm nessa reunião, são dados importantes para a próxima reunião de definição da *Sprint Backlog* seguinte (SCHWABER; SUTHERLAND, 2011).

Por fim deve ocorrer a reunião de retrospectiva da *Sprint*, que serve para a equipe traçar metas de melhorias a serem aplicadas na próxima *Sprint*, essa reunião deve sempre ser realizada depois da revisão da *Sprint* e antes da reunião de planejamento da próxima *Sprint*, por padrão ela tem uma duração de três horas, tendo como base uma *Sprint* de quatro semanas, para *Sprints* menores, as retrospectivas devem ser menores.

Esta reunião tem como propósitos: avaliar de que maneira se deu a relação entre as pessoas, processos e ferramentas; identificar e organizar os requisitos que foram bem, e possíveis melhorias e; elaborar um plano de implementação de melhorias, com base nas características de trabalho da equipe (SCHWABER; SUTHERLAND, 2011).

Ainda segundo o mesmo autor é importante salientar que, ao final de cada ciclo (*Sprint*), com as reuniões de revisão e de retrospectiva da *Sprint*, a equipe vai desenvolvendo práticas e percebendo melhorias a serem aplicadas nas *Sprints* seguintes.

REFERÊNCIAS

CARVALHO, Bernardo Vasconcelos de; MELLO, Carlos Henrique Pereira. **Revisão, análise e classificação da literatura sobre o método de desenvolvimento de produtos ágil Scrum**. São Carlos. Universidade Federal de Itajubá, 2009.

CARVALHO, Bernardo Vasconcelos de; MELLO, Carlos Henrique Pereira. **Aplicação do método ágil scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica**. São Carlos. Trabalho de conclusão de curso de graduação (Gestão de produtos) - Instituto de Engenharia de Produção e Gestão, Universidade Federal de Itajubá, 2010 .

KNIBERG, Enrick. SCRUM e XP direto das trincheiras: Como nós fazemos SCRUM, 2007. Disponível em:<www.infoq.com.br> Acesso em: 01/03/2013

MEDEIROS, Manoel Pimental. O diferencial Scrum. **Java Magazine**. Rio de Janeiro, ed. 73, p. 68-74, 2009.

PRADO, Gustavo Somadossi. **Avaliação do Impacto do Desenvolvimento Distribuído de Software em um Projeto Adotando o Scrum::** Um Estudo Comparativo. 2013. 10 f. TCC (Graduação) - Curso de Ciência da Computação, Departamento de Departamento de Computação, Universidade Federal de São Carlos, São Carlos, 2013. Disponível em: <<http://revistatis.dc.ufscar.br/index.php/revista/article/view/64>>. Acesso em: 05 jun. 2014.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: AMGH, 2010.

RISING, L.; JANOFF, N. S. The Scrum software development process for small teams. **IEEE Software**, v. 17, n. 4, p. 26-32, 2000.

ROSA, Magda Sharon Okumura. **SISTEMA DE GESTÃO DE PROJETOS BASEADOS EM SCRUM**. 2011. 103 f. Monografia (Especialização) - Curso de Sistemas da Informação, Universidade Anhembi Morumbi, São Paulo, 2011. Disponível em: <<http://engenharia.anhembi.br/tcc-11/si-04.pdf>>. Acesso em: 27 maio 2014.

SCHWABER, Ken; SUTHERLAND, Jeff. **Guia do Scrum: Um guia definitivo para o Scrum: As regras do jogo**. Cambridge: Scrum.org, 2011. 18 p. Disponível em: <www.scrum.org>. Acesso em: 10 abr. 2012.

SCRUM me: Ferramenta de Gestão de Projetos Scrum. , 2012. Disponível em: <<http://www.scrumme.com.br/web/login.aspx>>. Acesso em: 20 apr. 2014.

STANDISH GROUP (Usa) (Org.). **O relatório Standish Group de 2009**. Denver: Standish Group, 2009. Disponível em: <[http:// www.standishgroup.com/](http://www.standishgroup.com/)>. Acesso em: 11 jul. 2014.

UNISC (Santa Catarina). **Métodos de abordagem e de procedimento**. Florianópolis, 2012. p. 32 Disponível em: <http://www.unisc.br/portal/upload/com_arquivo/metodos_de_abordagem_e_de_procedimento.pdf>. Acesso em: 28 abr. 2013.

ANEXO A – QUESTIONÁRIO

Por que utilizam *Scrum*?

Como existem vários diferentes projetos na empresa e cada um deles conta com seu próprio gerente, líder e equipe, cada um possui uma metodologia de trabalho diferente. Sendo assim, o *Scrum* foi implementado diretamente pelo gestor do projeto em que faço parte principalmente para deixar a cargo de cada membro da equipe (cerca de 15 profissionais) a decisão de quando e como melhor atuar nas atividades que lhe são apresentadas. Também para priorizar a comunicação e visibilidade do que é importante para o bom andamento do projeto, de forma que todos saibam o que deve ser feito e o que está sendo feito em cada momento. Resumindo, pode-se dizer que utilizamos o *Scrum* para tentar dar agilidade ao processo de desenvolvimento do sistema na medida do possível, afinal, perseguimos sempre resultados.

Por onde começar?

Pode-se iniciar a implementação do *Scrum* orientando a equipe de como a metodologia funciona, apresentando as práticas e setando as prioridades para assim criar as *Sprints*. Além disso, foi importante aperfeiçoar as práticas conforme os processos que a empresa já utiliza e funcionam, de modo a ter resultados eficazes. Não utilizamos o *Scrum* como uma receita de bolo, e sim tirando proveito de seus melhores ingredientes baseando-se na realidade da empresa e do projeto.

Como definem o líder da equipe?

O líder da equipe foi escolhido devido ao seu conhecimento técnico da tecnologia utilizada no projeto e baseado em outros cases de sucesso em que o mesmo esteve liderando.

Como se dá a definição do *Sprint Backlog*?

Dá-se com o envolvimento de toda a equipe. Com o passar do tempo, a equipe adquire um entendimento maior sobre cada item sendo desenvolvido, sobre as regras de negócio e sobre a tecnologia utilizada. Desta forma novas ideias aparecem, ideias antigas são descartadas e o *Sprint Backlog* acompanha estas mudanças. Geralmente estas alterações são definidas no *Daily*.

Utilizam a programação conjunta?

Não.

Qual é o tempo da reunião diária e sua importância para o processo?

Não seguimos à risca o conceito da *Daily* (15 minutos, toda manhã, todos de pé, etc). Alguns dos profissionais trabalham remoto, de outras cidades (inclusive o gerente do projeto), portanto, esta atividade precisou ser adaptada à situação. As reuniões são agendadas para levarem até uma hora e são feitas à tarde, via sala de conferência. O tempo varia conforme os assuntos discutidos e a complexidade dos mesmos. Depois de o projeto atingir as fases finais, alguns profissionais da equipe foram realocados em outros projetos e isso resultou na diminuição do tempo das reuniões. Atualmente com 6 pessoas envolvidas, a *Daily* leva aproximadamente 30 minutos. Eu classifico a importância destas reuniões, em uma escala de 0 a 10, como 10. É a partir delas que revisamos a prioridade das atividades de cada um, identificamos os gargalos do desenvolvimento, dúvidas quanto aos testes, problemas identificados na aplicação, no servidor, na documentação, etc. e que medidas devem ser tomadas para resolvê-los e quem é o responsável por isso. Uma vez por semana temos uma reunião com o Cliente (duração de 1 hora) para discutir e alinhar os principais pontos que surgiram nos últimos dias e as pendências, tanto nossas, quanto deles. Desta reunião participam os gerentes, líderes técnicos e analistas (de testes e funcional).

O método *Scrum* melhorou a comunicação e a colaboração entre os envolvidos?

Sim. Em comparação a outros projetos que não utilizam a metodologia eu diria que nosso nível de comunicação e colaboração é bastante alto. A equipe trabalha mais unida por um objetivo comum, a qualidade aumenta e o retrabalho reduz consideravelmente.

O método *Scrum* aumentou a motivação da equipe?

Como o *Scrum* incentiva e aumenta muito o grau de interação entre os membros da equipe e entre cliente e equipe, acaba aumentando também a motivação dos mesmos. A visibilidade que todos têm das atividades em andamento faz crescer o comprometimento com as mesmas. Eu acredito que isso seja uma forma de motivação.

O método *Scrum* facilitou para que o projeto terminasse mais rápido?

Neste caso não. Eu particularmente acredito que pode ter havido certa confusão em nível de gestão quanto à ligação do *Scrum* com a documentação do projeto/produto, por não estar formalmente presente na metodologia. À medida que as entregas foram realizadas e deu-se início à fase de aceitação do Cliente, este percebeu que o sistema precisava de alguns ajustes

quanto às regras de negócios (falha da fase de análise e levantamento dos requisitos feitos em conjunto com o cliente). Felizmente já tínhamos a confiança deles quanto a entregar tudo aquilo com que tínhamos nos comprometido e os cronogramas puderam ser revistos, as melhorias levantadas e avaliadas. O projeto já atingiu o dobro das horas que haviam sido estimadas.

O método *Scrum* diminuiu os riscos do projeto e as possibilidades de insucesso?

Novamente não. Eu vejo o *Scrum* como um método ágil de gerenciamento de projetos e infelizmente, no caso deste projeto, os riscos e possibilidades de insucesso não foram previstos através deste método.

Existem dados comparando antes e o depois do *Scrum*?

Não.

Como manter o time motivado para não abandonar o processo?

Acredito que mantendo e cobrando a disciplina indiferente da metodologia utilizada. Além dos processos sugeridos pelo *Scrum*, há uma série de outros processos seguidos para que a organização e alinhamento da equipe se dêem do início ao fim do projeto. O *Scrum* não faz milagre, apenas ajuda a conduzir a equipe de maneira ágil.

Como são gerenciadas as atividades que precisam ser inseridas na *Sprint* (apagar o fogo)?

Geralmente as atividades são avaliadas e estimadas pelo líder e pelo responsável pela atividade, para que possam ser executadas causando o menor impacto possível.

Cite alguns pontos negativos do *Scrum*.

Geralmente as atividades são avaliadas e estimadas pelo líder e pelo responsável pela atividade, para que possam ser executadas causando o menor impacto possível.