

WAGNER PANSANI

DIRETRIZES PARA IMPLANTAÇÃO DO SCRUM

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para conclusão do curso de MBA em Tecnologia de Software.

São Paulo
2013

WAGNER PANSANI

DIRETRIZES PARA IMPLANTAÇÃO DO SCRUM

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de Software

Orientador: Prof. Dr. Kechi Hirama

São Paulo
2013

DEDICATÓRIA

*Dedico este trabalho a minha esposa
Rosele que sempre me apoia a cada
decisão tomada e aos meus
professores do PECE que foram
importantes na minha vida
acadêmica.*

AGRADECIMENTOS

À Universidade de São Paulo – USP por abrir espaço para este curso.

À Escola Politécnica da Universidade de São Paulo – EPUSP que forneceu toda a estrutura necessária ao longo desses 24 meses.

Ao PECE – Programa de Educação Continuada em Engenharia que elaborou um curso que atendia minhas expectativas.

Ao meu orientador Prof. Dr. Kechi Hirama pela paciência, suporte e incentivo durante o período de produção deste trabalho.

Aos professores do curso que me ajudaram durante o curso para aumentar meus conhecimentos.

Ao Allan Fagner Paiva Ferreira por me ajudar no entendimento da adoção do *Scrum* nas empresas enriquecendo o conteúdo deste trabalho.

À todos que me ajudaram a concretizar este trabalho, avaliando, opinando, corrigindo ou qualquer que fosse o auxílio.

RESUMO

O *Scrum* é o método ágil mais utilizado para o gerenciamento no desenvolvimento de produtos complexos e contém um conjunto de práticas com regras pré-definidas, papéis, eventos com duração pré-definida e envolvimento das partes interessadas durante o desenvolvimento do produto.

Porém, existem trabalhos que relatam experiências de implantação do *Scrum* conforme descrito no Guia do *Scrum* e tiveram problemas no gerenciamento do desenvolvimento de produtos de software.

Este trabalho apresenta o método *Scrum* com base no Guia do *Scrum*, mostrando todos os papéis, artefatos e eventos relacionados. Em seguida, são apresentados relatos referente à implantação do *Scrum*, indicando os problemas referentes ao gerenciamento e as soluções para resolvê-los. Para finalizar, são propostas as diretrizes para facilitar a implantação do *Scrum* com base no estudo do Guia do *Scrum* complementado com experiências encontradas na literatura.

ABSTRACT

Scrum is the most used agile method focused in the management of complex product development and it contains a set of practices with predefined rules, roles, events with specified time and involvement of the participants during this product development.

However, there are some experiences of Scrum implementation following the Scrum Guide that had problems in the management of the software product development.

This paper presents the Scrum method based in the Scrum Guide, showing all rules, artifacts, and related events. Then, some testimonials are presented, relating the Scrum implementation, the problems regarding the management and solutions to be taken in order to solve them. Finally, there are some guideline propositions in order to ease the Scrum implementation based in the Scrum Guide, complemented with experiences found in literature.

LISTA DE ILUSTRAÇÕES

Pág.

Figura 1: Visão geral do <i>Scrum</i> adaptado de Sutherland (2010, p.11).....	19
Figura 2: Exemplo de um Hiper-Épico adaptado de Read e Briggs (2012, p.5322). .	39
Figura 3: <i>Dashboard</i> e <i>Task Card</i> adaptado de Hong, Yoo, Cha (2010, p.313).....	44
Figura 4: Exemplo de um gráfico de <i>Burndown</i> adaptado de Cohn (2006, p.219-220).	45

LISTA DE TABELAS

	Pág.
Tabela 1: Relação dos problemas e soluções encontrados	31
Tabela 2: Papéis e responsabilidades em cada evento do <i>Scrum</i> adaptado de Hong, Yoo e Cha (2010, p.312).	35
Tabela 3: Detalhamento dos requisitos para a criação de uma Hiper-Estória adaptado de Read e Briggs (2012, p.5321-5322).	38
Tabela 4: Exemplo de um <i>Backlog</i> do Produto adaptado de Sutherland (2010, p.18).	41
Tabela 5: Exemplo de um <i>Backlog</i> da <i>Sprint</i> adaptado de Sutherland (2010, p.22). 42	
Tabela 6: Diretrizes apresentadas no capítulo 4.	50
Tabela 7: Tabela base para elaboração das diretrizes.....	55

SUMÁRIO

Pág.

1.	INTRODUÇÃO	11
1.1.	Motivações	11
1.2.	Objetivo	12
1.3.	Justificativas	12
1.4.	Estrutura do Trabalho	13
2.	MÉTODO SCRUM.....	14
2.1.	Papéis do <i>Scrum</i>	14
2.1.1.	<i>Product Owner</i>	14
2.1.2.	Equipe de Desenvolvimento	15
2.1.3.	<i>Scrum Master</i>	16
2.2.	Artefatos do <i>Scrum</i>	17
2.2.1.	<i>Backlog</i> do Produto.....	17
2.2.2.	<i>Backlog</i> da <i>Sprint</i>	18
2.2.3.	Incremento	18
2.3.	Eventos da <i>Sprint</i>	18
2.3.1.	Reunião de Planejamento.....	21
2.3.2.	Reunião Diária	22
2.3.3.	Revisão da <i>Sprint</i>	22
2.3.4.	Retrospectiva da <i>Sprint</i>	23
2.4.	Considerações do Capítulo	24
3.	EXPERIÊNCIAS DE IMPLANTAÇÃO DO MÉTODO SCRUM	25
3.1.	Implantação do <i>Scrum</i> em uma empresa que terceiriza projetos de e-commerce	25
3.2.	Implantação do <i>Scrum</i> em Projetos com equipe terceirizada	27
3.3.	Criação de um formato padrão para especificar histórias no <i>Backlog</i> do Produto	28
3.4.	Resumo de problemas e soluções encontrados	30
3.5.	Considerações do Capítulo	31
4.	DIRETRIZES PARA IMPLANTAÇÃO DO MÉTODO SCRUM.....	32
4.1.	Papéis do <i>Scrum</i>	32
4.2.	Tamanho da Equipe de desenvolvimento.....	36
4.3.	Artefatos do <i>Scrum</i>	36

4.4.	Eventos da <i>Sprint</i>	46
4.6.	Considerações do Capítulo	51
5.	CONSIDERAÇÕES FINAIS	52
5.1.	Contribuições do Trabalho.....	52
5.2.	Trabalhos Futuros.....	53
REFERÊNCIAS.....		54
APÊNDICE 1 – TABELA BASE PARA ELABORAÇÃO DAS DIRETRIZES		55

1. INTRODUÇÃO

Este capítulo apresenta as motivações, o objetivo, as justificativas ao tema e a estrutura do trabalho.

1.1. Motivações

No fim da década de 1990, vários métodos começaram a chamar atenção da indústria de *software*. Elas tinham uma combinação de diversos conceitos porém todas enfatizando uma colaboração mais próxima entre a equipe de programadores e os especialistas da área de negócio, como comunicação face a face (que é mais eficiente do que uma documentação escrita), entregas frequentes agregando valor ao negócio, equipes unidas e auto-gerenciáveis e formas de desenvolver código de modo que as mudanças no requisito não deixem o projeto em crise [Agile Alliance, 2011].

O *Scrum* é usado para gerenciar o desenvolvimento de produtos complexos sendo possível empregar processos ou técnicas de forma iterativa e incremental para melhorar a previsibilidade e conseguir um controle dos riscos no desenvolvimento do projeto [SCHWABER, SUTHERLAND, 2011, p.3-4].

De acordo com Krishna e Basu (2011, p.1), o *Scrum* é um dos métodos ágeis mais usados. Este método contém um conjunto de práticas com regras pré-definidas e com as partes interessadas envolvidas no desenvolvimento.

Krishna e Basu (2011, p.1) afirmaram que muitas organizações estão adotando esse método em seus projetos, mas somente alguns atingem o objetivo previamente estipulado pelas organizações. Uma das razões citadas é que as empresas não adotam o *Scrum* completamente porque não possuem conhecimento completo das práticas do *Scrum*.

Para superar esses problemas, foram propostos complementos ao método *Scrum* com novas práticas [KRISHNA, BASU, 2011, p.1] [HONG, YOO, CHA, 2010, p.1], mas sempre preservando as práticas originais descritas no Guia do *Scrum* [SCHWABER, SUTHERLAND, 2011, p.16].

A implantação do Scrum nas organizações de software não é simples, apesar de o Guia do *Scrum* estar bem descrito. Alguns trabalhos, como os de Hong, Yoo e Chaa (2010) e Read e Briggs (2012), relatam problemas e soluções específicas que defrontaram ao tentar implantar o Scrum.

1.2. Objetivo

O objetivo deste trabalho é propor diretrizes para facilitar a implantação do *Scrum* com práticas baseadas no estudo do Guia do *Scrum* e nos relatos de experiências de empresas que implantaram o *Scrum*.

1.3. Justificativas

Foram encontrados trabalhos contando experiências ao implantar o *Scrum* porém com foco específico nos problemas que encontraram durante a implantação.

Este trabalho é baseado fortemente no Guia do *Scrum* e na análise das experiências de implantação do *Scrum* encontradas na literatura. Por exemplo, Hong, Yoo e Cha (2010, p.1) relataram que implantar o *Scrum* utilizando somente as práticas originais não eram suficientes, então propuseram algumas práticas como a *Master Sprint Plan* para conseguir prever os itens de cada *Sprint* e a criação de *Task Cards* para a Equipe de Desenvolvimento definir uma lista com data de término e tempo necessário para cada tarefa e com isso diminuir a ociosidade da Equipe de Desenvolvimento.

Já Read e Briggs (2012, p.1), comentaram que nos métodos ágeis os requisitos são levantados na forma de histórias possuindo descrições curtas da interação do usuário com o sistema. Esta forma de levantar os requisitos não é adequada quando o projeto se torna maior porque a complexidade aumenta na quantidade de requisitos, quantidade e diversidade das necessidades das partes interessadas e quantidade de mudanças que precisam ser gerenciadas.

Foram realizadas diversas buscas na literatura, porém, até o momento não foi encontrado um trabalho que fornecesse diretrizes para facilitar a implantação do

Scrum de forma a evitar os problemas relatados e para que o método seja o mais efetivo possível.

1.4. Estrutura do Trabalho

Capítulo 1 - INTRODUÇÃO – Este capítulo apresenta as motivações, o objetivo, as justificativas e a estrutura do trabalho.

Capítulo 2 - MÉTODO *SCRUM* – Este capítulo apresenta os conceitos originais do *Scrum* destacando suas características, papéis e artefatos.

Capítulo 3 – EXPERIÊNCIAS DE IMPLANTAÇÃO DO MÉTODO *SCRUM* – Este capítulo apresenta relatos baseados na literatura ao implantar o método *Scrum*, suas experiências e problemas encontrados.

Capítulo 4 – DIRETRIZES PARA IMPLANTAÇÃO DO MÉTODO *SCRUM* – Este capítulo apresenta uma proposta de diretrizes para facilitar a implantação do método *Scrum* com novas práticas baseado nos estudos realizados.

Capítulo 5 - CONSIDERAÇÕES FINAIS – Este capítulo descreve as conclusões deste trabalho e possibilidades de trabalhos futuros.

REFERÊNCIAS - Relaciona todos os artigos e livros que foram usados neste trabalho.

APÊNDICE 1 – TABELA BASE PARA ELABORAÇÃO DAS DIRETRIZES - Apresenta uma tabela criada a partir de estudos do Guia do *Scrum* e da literatura apresentando os papéis, eventos e artefatos para servir como base para criar o capítulo 4.

2. MÉTODO SCRUM

O *Scrum* é usado para desenvolver e manter produtos complexos com uma abordagem iterativa e incremental e é fundamentado em teorias empíricas de controle de processos [SCHWABER, SUTHERLAND, 2011, p.4].

Um aspecto importante do *Scrum* é a inspeção e adaptação do produto sendo necessário que seja sempre verificado o produto resultante e a eficácia das práticas que estão sendo seguidas e adaptá-las caso seja necessário [SCHWABER, SUTHERLAND, 2011, p.4].

A implantação do método somente será considerada *Scrum* se forem aplicadas todos os papéis, artefatos e eventos. Caso seja implantada somente uma parte, não é possível afirmar que o resultado será *Scrum*, porém, podem ser agregadas novas técnicas, metodologias ou práticas [SCHWABER, SUTHERLAND, 2011, p.16].

Nos próximos tópicos serão apresentados os papéis do *Scrum* detalhando as responsabilidades, os artefatos do *Scrum* com suas definições e os eventos com suas finalidades.

2.1. Papéis do *Scrum*

O *Scrum* implementa sua estrutura através da Equipe *Scrum* que possui três papéis: *Product Owner*, Equipe de Desenvolvimento e o *Scrum Master* [SCHWABER, SUTHERLAND, 2011, p.5].

2.1.1. *Product Owner*

O *Product Owner* é o responsável por gerenciar o *Backlog* do Produto. Este gerenciamento inclui [SCHWABER, SUTHERLAND, 2011, p.5]:

- Expressar claramente os requisitos;
- Ordenar os requisitos para alcançar as metas e missões;
- Garantir o valor do trabalho realizado pela Equipe de Desenvolvimento;

- Garantir que o *Backlog* do Produto seja visível, transparente, claro para todos, e mostrar o que a Equipe *Scrum* vai trabalhar a seguir;
- Garantir que a Equipe de Desenvolvimento entenda os requisitos.

O *Product Owner* pode delegar o trabalho de gerenciamento do *Backlog* do Produto para a Equipe de Desenvolvimento, porém continua sendo o responsável pelos trabalhos de gerenciamento do *Backlog* do Produto [SCHWABER, SUTHERLAND, 2011, p.5].

2.1.2. Equipe de Desenvolvimento

A Equipe de Desenvolvimento é composta por profissionais que realizam o trabalho de entregar uma versão usável ao final de cada *Sprint*. A Equipe de Desenvolvimento é autorizada pela organização para gerenciar e organizar o próprio trabalho [SCHWABER, SUTHERLAND, 2011, p.6].

Segundo Schwaber e Sutherland (2011, p.6), a Equipe de Desenvolvimento possui as seguintes características:

- Eles são auto-organizados. Ninguém diz a Equipe de Desenvolvimento como transformar o *Backlog* do Produto em incrementos potencialmente usáveis;
- Equipes de Desenvolvimento são multifuncionais, possuindo todas as habilidades necessárias para criar o incremento do produto;
- O *Scrum* não reconhece papéis para os integrantes da Equipe de Desenvolvimento que não seja o Desenvolvedor, independentemente do trabalho que está sendo realizado pela pessoa;
- Individualmente os integrantes da Equipe de Desenvolvimento podem ter habilidades especializadas e área de especialização, mas a responsabilidade pertence à Equipe de Desenvolvimento como um todo;
- Equipes de Desenvolvimento não contem sub-equipes dedicadas a domínios específicos de conhecimento, tais como teste ou análise de negócios.

2.1.3. *Scrum Master*

O *Scrum Master* é o responsável em garantir que as práticas e as técnicas do *Scrum* sejam aplicadas de forma correta pela Equipe *Scrum* [SCHWABER, SUTHERLAND, 2011, p.7].

O *Scrum Master* serve o *Product Owner* de várias maneiras, incluindo [SCHWABER, SUTHERLAND, 2011, p.7]:

- Encontrar técnicas para o gerenciamento efetivo do *Backlog* do Produto;
- Comunicar claramente a visão, objetivo e requisitos do *Backlog* do Produto para a Equipe de Desenvolvimento;
- Ensinar a Equipe *Scrum* a criar os requisitos do *Backlog* do Produto;
- Facilitar os eventos *Scrum* conforme exigidos ou necessários.

Ainda de acordo com Schwaber e Sutherland (2011, p.7), o *Scrum Master* serve a Equipe de Desenvolvimento de várias maneiras, incluindo:

- Treinar a Equipe de Desenvolvimento em auto-gerenciamento;
- Remover impedimentos para o progresso da Equipe de Desenvolvimento;
- Facilitar os eventos *Scrum* conforme exigidos ou necessários;
- Treinar a Equipe de Desenvolvimento em ambientes organizacionais nos quais o *Scrum* não é totalmente adotado e compreendido.

Schwaber e Sutherland (2011, p.7) afirmam que o *Scrum Master* serve à organização de várias maneiras, incluindo:

- Liderar e treinar a organização na adoção do *Scrum*;
- Planejar implementações do *Scrum* dentro da organização;
- Ajudar partes interessadas a compreender e tornar aplicável o *Scrum*;
- Causar mudanças que aumentam a produtividade da Equipe *Scrum*;
- Trabalhar com outro *Scrum Master* para aumentar a eficácia da aplicação do *Scrum* nas organizações.

2.2. Artefatos do *Scrum*

Os artefatos do *Scrum* têm como foco maximizar a transparência das informações principais para que a Equipe *Scrum* tenha êxito na entrega do incremento Pronto [SCHWABER, SUTHERLAND, 2011, p.12].

A definição de Pronto é quando o desenvolvimento de um requisito está completo e usável, ou seja, testado e potencialmente entregável no incremento do produto e isto deve estar claro para toda a Equipe *Scrum*. Esta definição orienta a Equipe de Desenvolvimento de quantos requisitos do *Backlog* do Produto podem ser selecionados durante a reunião de Planejamento da *Sprint* [SCHWABER, SUTHERLAND, 2011, p.15].

2.2.1. *Backlog* do Produto

O *Backlog* do Produto é uma lista ordenada dos requisitos do produto e serve como origem única para qualquer mudança necessária no produto. Esta lista contém as características, funções, requisitos, melhorias e correções que devem ser feitas no produto em futuras versões [SCHWABER, SUTHERLAND, 2011, p.13].

O *Product Owner* é o responsável pelo *Backlog* do Produto, incluindo seu conteúdo, disponibilidade e ordenação [SCHWABER, SUTHERLAND, 2011, p.13].

No início do desenvolvimento, o *Backlog* do Produto possui apenas os requisitos inicialmente conhecidos e conforme o produto evolui, o *Backlog* do Produto será atualizado pelo *Product Owner* para refletir essas mudanças. [SCHWABER, SUTHERLAND, 2011, p.13].

O *Backlog* do Produto normalmente é ordenado por valor, risco, prioridade e necessidade, sendo que os requisitos que estão no topo da lista estão mais detalhados e devem ser desenvolvidos o mais rápido possível. Estes requisitos devem ser decompostos de forma que eles possam ser entregues dentro da *Sprint* [SCHWABER, SUTHERLAND, 2011, p.13].

O *Backlog* do Produto está em constante atualização para adicionar detalhes, estimativas e ordenação aos requisitos da lista. Esta atualização é feita pelo *Product Owner* com a ajuda da Equipe de Desenvolvimento sendo que as estimativas são de responsabilidade da Equipe de Desenvolvimento já que eles vão realizar o trabalho [SCHWABER, SUTHERLAND, 2011, p.14].

2.2.2. Backlog da Sprint

O *Backlog* da *Sprint* contém os requisitos que foram selecionados do *Backlog* do Produto para serem desenvolvidos e entregues na *Sprint*. Esta lista possui detalhes suficientes para que sejam entendidas durante a Reunião Diária [SCHWABER, SUTHERLAND, 2011, p.14].

Durante a *Sprint* a Equipe de Desenvolvimento atualiza constantemente o *Backlog* da *Sprint* adicionando novas tarefas caso sejam necessárias. Sempre que uma tarefa é completada a estimativa da tarefa restante é atualizada e se alguma tarefa for considerada desnecessária ela é removida da lista [SCHWABER, SUTHERLAND, 2011, p.14].

2.2.3. Incremento

Segundo Schwaber e Sutherland (2011, p.15), o incremento são todos os requisitos que foram completados nessa *Sprint* e nas *Sprints* anteriores e que a Equipe *Scrum* definiu que estão prontos para o uso, independente do *Product Owner* decidir por liberá-lo.

2.3. Eventos da Sprint

O *Scrum* possui eventos pré-programados com duração pré-determinada, e todo evento possui uma duração máxima para criar uma rotina entre os participantes e diminuir a necessidade de reuniões que não estavam previstas além de permitir a inspeção e adaptação, seja do produto ou do processo no gerenciamento do desenvolvimento do produto de software. Com isso é possível garantir uma quantidade adequada de tempo sem perdas no processo do planejamento [SCHWABER, SUTHERLAND, 2011, p.8].

A *Sprint* é a parte principal do *Scrum*, onde um incremento possivelmente usável do produto é desenvolvido e, enquanto uma *Sprint* estiver em andamento, os requisitos selecionados do *Backlog* do Produto não podem ser alterados [SCHWABER, SUTHERLAND, 2011, p.8].

Uma *Sprint* somente poderá ser cancelada se a meta da *Sprint* se tornar obsoleta e, nesse caso, todos os requisitos que já estavam desenvolvidos são revisados e os que estavam incompletos são estimados novamente e devolvidos para o *Backlog* do Produto [SCHWABER, SUTHERLAND, 2011, p.8].

Conforme a Figura 1 pode-se ver de forma resumida o funcionamento do *Scrum*:

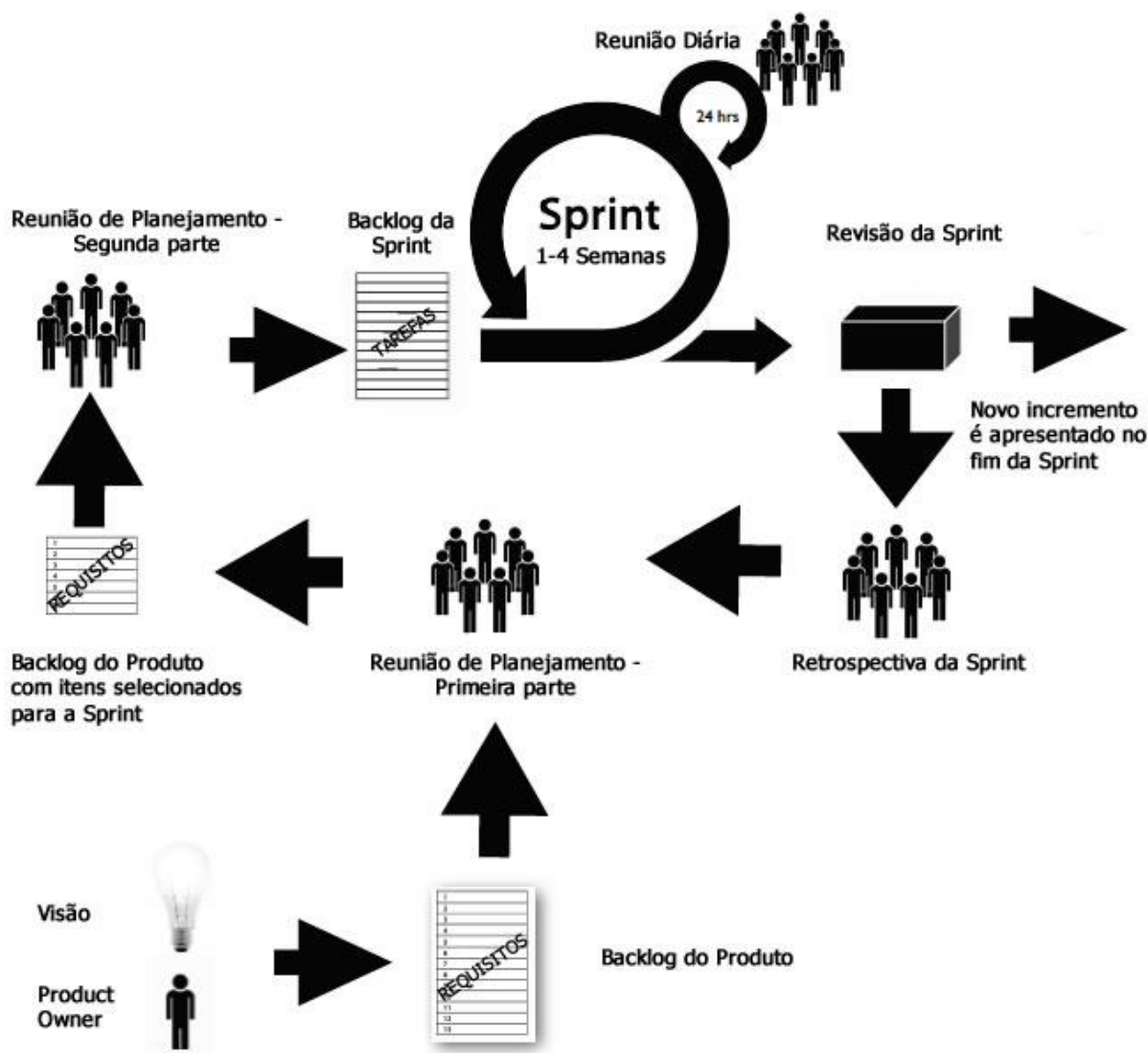


Figura 1: Visão geral do *Scrum* adaptado de Sutherland (2010, p.11).

O *Product Owner* realiza o levantamento dos requisitos, cria e ordena o *Backlog* do Produto [SCHWABER, SUTHERLAND, 2011, p.5].

Após a criação do *Backlog* do Produto é feita a primeira parte da Reunião de Planejamento. O *Product Owner* apresenta o *Backlog* do Produto ordenado e a Equipe de Desenvolvimento seleciona os requisitos que podem ser completados na *Sprint* [SCHWABER, SUTHERLAND, 2011, p.9].

Na segunda parte da Reunião de Planejamento a Equipe de Desenvolvimento irá dividir os requisitos previamente selecionados em tarefas menores com um dia de duração ou menos, irão criar o *Backlog* da *Sprint* com estas tarefas e decidirão como vão desenvolvê-las para criar um incremento de produto Pronto [SCHWABER, SUTHERLAND, 2011, p.10].

Diariamente a Equipe de Desenvolvimento se reúne para inspecionar o trabalho que está sendo feito desde a última reunião, prever o que será feito até a próxima, quais impedimentos estão tendo para o desenvolvimento e avaliar se com o progresso atual, verificando se conseguirão atingir a meta da *Sprint* [SCHWABER, SUTHERLAND, 2011, p.11].

No fim da *Sprint* é feita a reunião de Revisão da *Sprint* para inspecionar o incremento e atualizar o *Backlog* do Produto caso seja necessário. Nesta reunião, a Equipe de Desenvolvimento apresenta o que está Pronto e o *Product Owner* verifica o que está Pronto ou não. A Equipe de Desenvolvimento discute o que foi bem durante a *Sprint*, quais problemas tiveram e como resolveram [SCHWABER, SUTHERLAND, 2011, p.11-12].

Em seguida é feita a reunião de Retrospectiva da *Sprint* para a Equipe *Scrum* inspecionar a si próprio criando um plano de melhorias que será aplicada na próxima *Sprint*. Esta reunião tem como objetivo inspecionar como foi a última *Sprint* em relação às pessoas, processos e ferramentas, identificar e ordenar o que foi bem com suas potenciais melhorias e criar um plano para implementar essas melhorias [SCHWABER, SUTHERLAND, 2011, p.12].

2.3.1. Reunião de Planejamento

No início de cada *Sprint* é feita uma reunião de planejamento para decidir o que será feito e como o trabalho será realizado para entregar o incremento do produto [SCHWABER, SUTHERLAND, 2011, p.9].

Esta reunião possui duração de 8 horas para uma *Sprint* de um mês, caso a *Sprint* possua um tempo menor, a duração da reunião será proporcional ao tempo da *Sprint*. Por exemplo, para uma *Sprint* de duas semanas, o tempo da Reunião de Planejamento deverá ser de quatro horas [SCHWABER, SUTHERLAND, 2011, p.9].

A Reunião de Planejamento é dividida em duas partes sendo cada uma, metade do tempo total da reunião [SCHWABER, SUTHERLAND, 2011, p.9].

A primeira parte da reunião serve para entender o que o *Product Owner* precisa que seja desenvolvido na *Sprint* em questão e toda a Equipe *Scrum* participa. O *Product Owner* apresenta os requisitos do *Backlog* do Produto ordenados para a Equipe de Desenvolvimento que seleciona o número de requisitos que podem ser completados na *Sprint* [SCHWABER, SUTHERLAND, 2011, p.9].

Após a Equipe de Desenvolvimento selecionar os requisitos que poderão ser completados na *Sprint*, a Equipe *Scrum* determina a meta da *Sprint*. A Equipe de Desenvolvimento trabalha com a meta da *Sprint* em mente com o objetivo de atingi-lo. Caso o trabalho não seja o esperado na meta da *Sprint*, é negociado com o *Product Owner* o escopo do *Backlog* da *Sprint* [SCHWABER, SUTHERLAND, 2011, p.9-10].

A segunda parte da reunião serve para a Equipe de Desenvolvimento planejar como irá construir os requisitos selecionados e transformá-las em um incremento de produto Pronto. [SCHWABER, SUTHERLAND, 2011, p.10].

Com o trabalho planejado para os próximos dias, a Equipe de Desenvolvimento detalha os requisitos em unidades de um dia ou menos até o fim desta reunião. O *Product Owner* pode estar presente na segunda parte da reunião para esclarecer os

requisitos do *Backlog* do Produto selecionados. Caso a Equipe de Desenvolvimento determine que exista um excesso ou falta de trabalho, os requisitos podem ser renegociados com o *Product Owner* [SCHWABER, SUTHERLAND, 2011, p.10].

2.3.2. Reunião Diária

A Reunião Diária possui duração de 15 minutos e o propósito é de a Equipe de Desenvolvimento criar um plano para as próximas 24 horas e discutir o que foi feito desde a última reunião, o que será feito até a próxima reunião e quais impedimentos estão no caminho. O *Scrum Master* participa desta reunião apenas para garantir que a reunião aconteça, mas é responsabilidade da Equipe de Desenvolvimento conduzi-la [SCHWABER, SUTHERLAND, 2011, p.10-11].

Essa reunião não serve para reportar o andamento do desenvolvimento e sim para uma auto-organização da Equipe de Desenvolvimento, pois eles devem estar aptos a esclarecer para o *Product Owner* ou o *Scrum Master* como pretendem trabalhar em conjunto para completar o objetivo e criar um incremento do produto no restante da *Sprint* [SCHWABER, SUTHERLAND, 2011, p.11].

Schwaber e Sutherland (2011, p.11) destacam a importância da Reunião Diária, pois melhoram a comunicação, eliminam outras reuniões, destacam e promovem rápidas tomadas de decisão e melhoram o conhecimento da Equipe de Desenvolvimento.

Nesta reunião é verificado o progresso atual da *Sprint* em relação à meta da *Sprint* e se será possível completar o trabalho que está no *Backlog* da *Sprint*. Fazendo esta reunião, a probabilidade de a Equipe de Desenvolvimento atingir a meta da *Sprint* aumenta [SCHWABER, SUTHERLAND, 2011, p.11].

2.3.3. Revisão da *Sprint*

Esta reunião possui duração de 4 horas para uma *Sprint* de um mês, proporcionalmente menor caso a *Sprint* tenha duração menor [SCHWABER, SUTHERLAND, 2011, p.11].

A reunião de Revisão da *Sprint* acontece quando a *Sprint* acaba e participam a Equipe *Scrum* e as partes interessadas e serve para inspecionar o incremento e atualizar o *Backlog* do Produto. [SCHWABER, SUTHERLAND, 2011, p.11].

Schwaber e Sutherland (2011, p.11-12) destacam os seguintes pontos durante a reunião de Revisão da *Sprint*:

- A Equipe de Desenvolvimento apresenta o trabalho que está Pronto e responde as questões sobre o incremento;
- O *Product Owner* identifica o que está Pronto e o que não está Pronto;
- A Equipe de Desenvolvimento discute o que foi bem realizado durante a *Sprint*, quais problemas ocorreram, e como estes problemas foram resolvidos;
- Toda a equipe colabora sobre o que fazer a seguir, e deste modo que a reunião de Revisão da *Sprint* fornece valiosas entradas para a Reunião de Planejamento da próxima *Sprint*.

2.3.4. Retrospectiva da *Sprint*

Esta reunião é uma oportunidade para a Equipe *Scrum* inspecionar a si próprio criando um plano de melhorias que serão implementadas na próxima *Sprint* [SCHWABER, SUTHERLAND, 2011, p.12].

Esta reunião acontece logo após a reunião de Revisão da *Sprint* e antes da Reunião de Planejamento da próxima *Sprint* e possui duração de três horas para uma *Sprint* de um mês, proporcionalmente menor caso a *Sprint* tenha duração menor [SCHWABER, SUTHERLAND, 2011, p.12].

Segundo Schwaber e Sutherland (2011, p.12), o objetivo desta reunião é:

- Inspeccionar como a última *Sprint* foi em relação às pessoas, relações, processos e ferramentas;
- Identificar e ordenar o que foi bem e as potenciais melhorias;
- Criar um plano para implementar melhorias no modo que a Equipe *Scrum* faz seu trabalho.

O *Scrum Master* incentiva a Equipe *Scrum* a melhorar respeitando os processos do *Scrum* para torná-lo mais efetivo para a próxima *Sprint*. Durante a reunião, a Equipe *Scrum* discute as formas para aumentar a qualidade do produto e adaptando a definição de Pronto quando é apropriado [SCHWABER, SUTHERLAND, 2011, p.12].

2.4. Considerações do Capítulo

A forma como está estruturado o Guia do *Scrum* pode confundir quem está implantando o *Scrum* pela primeira vez porque as definições e responsabilidades do papéis, por exemplo, estão especificadas ao longo do Guia do *Scrum* e não em um capítulo específico.

Em alguns momentos o Guia do *Scrum* não é claro o suficiente deixando dúvidas como por exemplo: definições dos papéis nem sempre são claros, não existe um exemplo de como deve ser criado o *Backlog* do Produto e *Backlog* da *Sprint*, não é especificado o nível de detalhamento dos requisitos e nem como o progresso do desenvolvimento pode ser monitorado.

Conforme encontrado na literatura, ao implantar o *Scrum* da maneira como está descrito no Guia do *Scrum*, percebe-se que algumas práticas não estão descritas como por exemplo: formas para prever a quantidade de *Sprints* que um projeto terá, um formato especificar as histórias do *Backlog* do Produto e até mesmo uma tabela com um resumo das responsabilidades de cada papel em cada evento do *Scrum*.

No próximo capítulo são apresentados relatos de problemas encontrados na implantação do *Scrum*.

3. EXPERIÊNCIAS DE IMPLANTAÇÃO DO MÉTODO *SCRUM*

Neste capítulo são apresentados os relatos mais relevantes de experiências de implantações do método *Scrum*, quais foram os desafios enfrentados, os problemas e qual foi o resultado final obtido. As soluções para os problemas encontrados são apresentadas resumidamente neste capítulo e detalhados no capítulo 4.

Estes relatos foram selecionados após a leitura de diversos trabalhos, porém foram selecionados apenas os que traziam informações relevantes dos problemas enfrentados durante a implantação do *Scrum*.

3.1. Implantação do *Scrum* em uma empresa que terceiriza projetos de *e-commerce*

Hong, Yoo e Cha (2010, p.310) relataram que uma empresa de *e-commerce* na Coreia do Sul desenvolvia todos os projetos usando o modelo cascata. Dados estatísticos de 2009 mostraram que os projetos desenvolvidos internamente possuíam uma taxa de sucesso de 89%, porém em projetos que eram terceirizados as taxas de sucesso eram de 60%, atraso 20% e falhas 20%. Essas taxas mostravam que a qualidade do produto terceirizado era inferior comparado aos desenvolvidos internamente.

A empresa decidiu implantar o *Scrum*, porém perceberam que existiam alguns problemas [HONG, YOO, CHA, 2010, p.310]:

1. Papéis e responsabilidades não eram claros o suficiente para a execução do processo durante a fase inicial. Resolveram criar uma tabela relacionando as responsabilidades dos papéis em cada evento do *Scrum*;
2. Durante a fase de planejamento, o *Scrum* possui foco somente nos *Sprints* incrementais não permitindo uma visão do planejamento geral do projeto tornando difícil a projeção de prazos que podem afetar outros processos ou unidades no ciclo de desenvolvimento. Resolveram criar uma fase antes da Reunião de Planejamento chamada de *Master Sprint Plan* onde eram feitas todas as estimativas das histórias e distribuídas em *Sprints*;

3. Relatórios de progresso usando gráficos *Burndown* baseados em *Story Points* eram inconvenientes durante a Revisão da *Sprint* devido ao recálculo dos *Story Points* no fim de cada *Sprint*. Resolveram trocar os *Story Points* por quantidade de páginas porque as páginas que seriam desenvolvidas possuíam complexidade parecida.

Além dos problemas citados acima, foram implementados um *Dashboard* e *Task Cards*.

Os *Task Cards* são uma lista de tarefas que ficam fixadas no *Dashboard* e indicam as tarefas que precisam ser completadas no *Backlog* da *Sprint* e possuem duração máxima de 24 horas. O membro da Equipe de Desenvolvimento responsável por atualizar o *Backlog* da *Sprint* cria os *Task Cards* [HONG, YOO, CHA, 2010, p.313].

Eles foram criados para a Equipe de Desenvolvimento poder planejar durante a *Sprint* a ordem em que as tarefas seriam desenvolvidas, com data de início e término da tarefa permitindo que todos se planejassem diminuindo a ociosidade entre a Equipe de Desenvolvimento [HONG, YOO, CHA, 2010, p.313].

Foram terceirizados dois projetos na Coréia do Sul utilizando o *Scrum* com estas mudanças para verificar sua efetividade e foi verificado que este método pode melhorar a qualidade dos produtos, reduzir a ociosidade em cada fase e realizar a entrega pontual dos projetos terceirizados [HONG, YOO, CHA, 2010, p.310].

Como resultado, foi verificado que as taxas de defeitos dos projetos terceirizados caíram e ficaram muito próximas das taxas de defeitos dos projetos desenvolvidos internamente. Foi percebido também que o *Scrum* com as novas mudanças utilizou os recursos disponíveis com mais eficiência reduzindo o tempo de espera para o desenvolvimento das tarefas [HONG, YOO, CHA, 2010, p.315].

Pesquisas feitas com a Equipe *Scrum* depois de usar o *Scrum* com as novas implementações em novos projetos, mostraram mais de 80% estavam satisfeitos no uso do método [HONG, YOO, CHA, 2010, p.315].

3.2. Implantação do *Scrum* em Projetos com equipe terceirizada

Silva et al. (2012, p.328) relataram que a empresa onde trabalhavam, a Andrade Gutierrez, possuía uma área de Tecnologia da Informação e contava com um escritório de projetos que era responsável pela comunicação com os clientes que eram as outras áreas da empresa.

A empresa possuía também uma fábrica de *software* que gerenciava todo o desenvolvimento de *software* da Andrade Gutierrez. O desenvolvimento era terceirizado e existiam dois tipos de terceirização: a primeira é quando os desenvolvedores terceirizados estão alocados no próprio prédio da empresa e na segunda os desenvolvedores terceirizados estão alocados em outra cidade [SILVA et al., 2012, p.328].

A empresa decidiu migrar para o *Scrum* para agilizar o desenvolvimento, além da proximidade dos papéis do *Scrum* com a formação da área de TI da empresa. O analista de sistemas assumiu o papel de *Scrum Master* e o analista de negócios que antes era o cliente, assumiu o papel de *Product Owner* [SILVA et al., 2012, p.328-332].

A implantação do *Scrum* foi feita em dois sistemas em desenvolvimento no setor de TI da empresa. O *Scrum Master* e o *Product Owner* permaneceram na sede da empresa [SILVA et al., 2012, p.331].

Durante a implantação, a distância entre o *Product Owner* e *Scrum Master* da Equipe de Desenvolvimento que estava alocada em outra cidade foi um problema, causando uma redução no número de Reuniões de Planejamento, Reunião de Retrospectiva e Reunião Diária [SILVA et al., 2012, p.332-334].

Outro problema encontrado foi a necessidade de conscientização do *Product Owner* que deveria possuir uma visão clara da importância do seu papel e que suas escolhas e prioridades eram o que davam valor a todo trabalho já que ele deveria saber o que é prioritário para o negócio e é responsável pelo retorno sobre o investimento [SILVA et al., 2012, p.332-334].

A implantação do *Scrum* na Equipe de Desenvolvimento que estava alocada na empresa não teve os problemas que afetaram a outra equipe, ou seja, todas as reuniões foram feitas e a proximidade diária do *Scrum Master* e o *Product Owner* contribuíram para o sucesso da implantação [SILVA et al., 2012, p.332].

Apesar dos problemas citados, o *Scrum* foi fundamental para a entrega pontual e funcional para o cliente que mostrou estar bastante satisfeito com a flexibilidade do *Scrum*. Antes do *Scrum*, o cliente fazia solicitações urgentes que impactavam o trabalho atual da Equipe de Desenvolvimento fazendo com que o trabalho fosse perdido [SILVA et al., 2012, p.334].

Em seguida, com o cliente se tornando o *Product Owner* ele se sentiu parte da equipe conseguindo acompanhar o que seria entregue e sanando rapidamente as dúvidas da equipe [SILVA et al., 2012, p.334].

3.3. Criação de um formato padrão para especificar histórias no *Backlog* do Produto

Read e Briggs (2012, p.5319) relataram que fizeram pesquisas por dois anos e meio de como detalhar as histórias dos requisitos para software complexos com base no *Scrum* capturando as complexidades dos requisitos do software sem sobrecarregar a equipe com documentações desnecessárias.

Nos métodos ágeis, os requisitos são levantados na forma de histórias possuindo uma descrição curta da interação do usuário com o sistema. Esta forma de levantar os requisitos não funciona bem quando um projeto se torna maior, pois a complexidade aumenta na quantidade de requisitos, a quantidade e a diversidade das necessidades das partes interessadas que devem ser satisfeitas e a quantidade de mudanças que precisam ser gerenciadas. [READ, BRIGGS, 2012, p.5319].

Os métodos ágeis não fornecem um formato para levantar os requisitos, o que torna difícil analisar uma grande quantidade de requisitos que precisam ser desenvolvidos, aumenta a dificuldade de analisar o impacto das mudanças dos requisitos além de o conhecimento do projeto ficar na cabeça das pessoas que

participaram das reuniões tornando difícil transferir o conhecimento do projeto para uma equipe grande [READ, BRIGGS, 2012, p.5319].

No mês anterior de a Equipe de Desenvolvimento começar o desenvolvimento, o *Product Owner* e o *Scrum Master* se reuniram diversas vezes para escrever as histórias. Com pouca experiência, o *Product Owner* e o *Scrum Master* escreveram histórias curtas, vagas e ambíguas como por exemplo: “Efetuar *login* no sistema”. No exemplo, não era possível identificar quem poderia efetuar o *login*, sob quais condições e quais deveriam ser os controles de acesso [READ, BRIGGS, 2012, p.5321].

Na primeira Reunião de Planejamento da *Sprint*, a Equipe de Desenvolvimento fez diversos questionamentos referente as histórias, fizeram as estimativas e se comprometeram a desenvolvê-las ao longo da *Sprint*. Durante a *Sprint*, com pouca experiência nos métodos ágeis, a Equipe de Desenvolvimento e o *Product Owner* não discutiram o significado das histórias e nem como seriam desenvolvidas [READ, BRIGGS, 2012, p.5321].

No fim da *Sprint*, o *Product Owner* não ficou satisfeito porque muitas funções não ficaram da forma que ele esperava e outras, apesar de estarem bem desenvolvidas não serviam para o propósito do projeto. Então um programador relatou que a história não dizia que ele deveria ter feito isso, então não foi feito [READ, BRIGGS, 2012, p.5321].

Na *Sprint* seguinte, o *Product Owner* tentou criar especificações documentadas mas o trabalho requerido era muito alto e excedia o tempo disponível. A equipe concluiu que o projeto iria falhar se eles não encontrassem uma forma rápida de se comunicar sobre os requisitos. Eles passaram a estudar uma forma de escrever as histórias que fornecessem detalhes suficientes sobre os requisitos para que a Equipe de Desenvolvimento conseguisse desenvolver de forma satisfatória e fosse necessário menos esforço para criar e manter as histórias [READ, BRIGGS, 2012, p.5321].

Nas *Sprints* seguintes, o *Product Owner* e o *Scrum Master*, com ajuda dos outros membros da equipe, criaram, testaram e rejeitaram diversos formatos de histórias até que eles conseguiram chegar no conceito de Hiper-Épico e Hiper-Estórias [READ, BRIGGS, 2012, p.5321-5322].

Uma Hiper-Estória é semelhante a uma história convencional porém possui um formato padrão com regras e podem possuir também parâmetros (caso algum aspecto se repita em várias Hiper-Estórias) e regras de negócio. Um Hiper-Épico contém uma coleção de Hiper-Estórias relacionadas. [READ, BRIGGS, 2012, p.5321-5322].

O Hiper-Épico reduz o tempo para fazer a documentação enquanto fornece mais detalhes que as histórias convencionais. É mais simples de manter atualizado do que as especificações formais porque uma simples revisão de um parâmetro pode alterar todas as histórias que ele faz parte [READ, BRIGGS, 2012, p.5322].

O Hiper-Épico facilita o entendimento do projeto do processo e do produto referente aos métodos ágeis. As partes interessadas também foram capazes de compreender todo o processo mesmo com vários requisitos bem detalhados [READ, BRIGGS, 2012, p.5328].

3.4. Resumo de problemas e soluções encontrados

Conforme foi relatado, o *Scrum* possui alguns problemas no gerenciamento do desenvolvimento do produto que podem ser melhorados, porém, alguns destes problemas estão relacionados com o projeto ou com a cultura organizacional e não podem ser generalizados para qualquer empresa.

A Tabela 1 relaciona os problemas e soluções encontrados.

Tabela 1: Relação dos problemas e soluções encontrados

Problema encontrado	Solução encontrada	Referência
Papéis e responsabilidades não são claros o suficientes em cada evento do <i>Scrum</i> .	Criar uma tabela exibindo a responsabilidade de cada papel em cada evento do <i>Scrum</i> .	HONG, YOO, CHA, 2010
O <i>Scrum</i> possui foco somente nas <i>Sprints</i> incrementais não permitindo uma visão geral do cronograma.	Criar um evento chamado <i>Master Sprint Plan</i> para poder estimar as estórias e distribuir nas <i>Sprints</i> .	HONG, YOO, CHA, 2010
Ociosidade da Equipe de Desenvolvimento durante a <i>Sprint</i> porque os integrantes não sabem quando uma tarefa estará pronta.	Criar um <i>Dashboard</i> e os <i>Tasks Cards</i> para poder criar um planejamento das tarefas existentes com data provável de início e término.	HONG, YOO, CHA, 2010
Definição do papel do <i>Product Owner</i> .	Certificá-lo para que ele saiba suas responsabilidades e a sua importância.	SILVA et al., 2012
Problema nas especificações das estórias porque o Guia do <i>Scrum</i> não descreve uma forma de criá-las fazendo com que as especificações não fiquem detalhadas o suficiente para a Equipe de Desenvolvimento.	Criar um padrão de Hiper-Épicos e Hiper-Estórias para poder especificar as estórias no nível de detalhamento necessário.	READ, BRIGGS, 2012

3.5. Considerações do Capítulo

A implantação do *Scrum* conforme o Guia do *Scrum* pode fazer com que seja necessário a criação de novas práticas para tornar o gerenciamento do *Scrum* mais completo devido a problemas não gerenciados pelo *Scrum*, como por exemplo a falta de uma reunião para definir todas as *Sprints*, um padrão para detalhar os requisitos ou mesmo uma forma de diminuir a ociosidade da Equipe de Desenvolvimento enquanto estão desenvolvendo as tarefas.

As dificuldades na implantação do *Scrum* descritas neste capítulo levaram à necessidade de complementar o método original com algumas novas práticas tornando-o mais completo e eficaz.

No capítulo 4, são apresentadas as diretrizes para a implantação do *Scrum* com base no Guia do *Scrum* descrito no capítulo 2 e nos relatos e soluções apresentados neste capítulo.

4. DIRETRIZES PARA IMPLANTAÇÃO DO MÉTODO *SCRUM*

Este capítulo contém diretrizes para a implantação do método *Scrum* seguindo as práticas descritas no Guia do *Scrum* e complementadas com os relatos de implantação do capítulo 3.

As diretrizes foram criadas com base na Tabela 7 presente no Apêndice 1. A Tabela 7 foi criada com base em estudos do Guia do *Scrum* e na literatura contendo as práticas, definições de papéis e eventos recomendados para facilitar a implantação do *Scrum*.

Algumas diretrizes referentes ao conceito do *Scrum* estarão descritas de forma sucinta, como por exemplo, a definição do *Product Owner*, *Scrum Master* e a Equipe de Desenvolvimento com um detalhamento maior nas diretrizes que não são conceituais, porém todas as etapas para o gerenciamento do desenvolvimento do produto vão estar presentes neste capítulo.

Os implantadores do *Scrum* devem ter um conhecimento prévio do Guia do *Scrum* para que não tenham dúvidas na leitura destas diretrizes.

As diretrizes deste capítulo são apresentadas com um título em negrito informando qual é o propósito da diretriz e logo em seguida é detalhado como a diretriz deve ser aplicada.

4.1. Papéis do *Scrum*

Entender as responsabilidades de cada papel.

Além dos artefatos e eventos, o *Scrum* é composto pela Equipe *Scrum* e são divididos em três papéis que são [SCHWABER, SUTHERLAND, 2011, p.5-7] [SUTHERLAND, 2010, p.14]:

- ***Product Owner***: Responsável por definir os requisitos do produto e criar o *Backlog* do Produto, tornando-os claros o suficiente para que a Equipe

de Desenvolvimento consiga entender e garantir o valor do trabalho realizado pela Equipe de Desenvolvimento;

- **Equipe de Desenvolvimento:** São pessoas autorizadas pela organização para gerenciar e organizar o próprio trabalho, possuem as habilidades necessárias para criar o incremento do produto e são responsáveis por criar o produto que será entregue para o cliente;
- **Scrum Master:** Responsável por garantir que as práticas e técnicas do *Scrum* sejam aplicadas corretamente pela Equipe *Scrum*. O *Scrum Master* serve a todos da seguinte maneira:
 - **Equipe de Desenvolvimento:** Fazer com que os problemas externos não atrapalhem o trabalho da equipe, treiná-los no auto-gerenciamento e interdisciplinaridade e facilitar os eventos necessários;
 - **Product Owner:** Encontrar maneiras para criar e gerenciar de forma efetiva o *Backlog* do Produto e facilitar os eventos necessários;
 - **Organização:** Liderar e treinar a adoção do *Scrum*, fazer as mudanças necessárias que aumentam a produtividade da Equipe *Scrum* e trabalhar com outros *Scrum Masters* para aumentar a eficácia da aplicação do *Scrum*.

O *Scrum Master* e o *Product Owner* não podem ser a mesma pessoa porque as vezes o *Product Owner* tenta inserir novos requisitos em uma *Sprint* que está em andamento e o *Scrum Master* não deve permitir que isso aconteça [SUTHERLAND, 2010, p.16].

Semanalmente deve ser feita uma reunião com todos os *Scrum Masters* e *Product Owners* para conversarem sobre os impedimentos e principais funções que estão sendo desenvolvidas ou planejadas que afetam outras equipes. Devem ser discutidas também questões que estavam pendentes, políticas corporativas e decisões. Esta reunião é importante porque contribui para uma sincronização entre as equipes e a maioria das ações para melhorias contínuas são discutidas nesta reunião [MARANZATO, NEUBERT, HERCULADO, 2012, p.83].

Para garantir que toda a Equipe *Scrum* entenda o funcionamento do *Scrum* e suas responsabilidades, pode-se também certificar toda a equipe como *Professional Scrum Master I* para poder explicar todos os artefatos do *Scrum* para que seja implantada da melhor forma possível [SILVA et al., 2012, p.332].

Com a equipe certificada, o ideal é também certificar o *Product Owner* como *Professional Scrum Product Owner* para que ele entenda claramente a importância do seu papel no gerenciamento do desenvolvimento do produto de software [SILVA et al., 2012, p.332].

Para garantir que todas as reuniões aconteçam é necessário que o *Product Owner* e o *Scrum Master* estejam fisicamente no mesmo local que a Equipe de Desenvolvimento. Caso não estejam, é possível que algumas reuniões não aconteçam com a frequência necessária fazendo com que definições não sejam feitas e dúvidas não sejam sanadas impactando negativamente no andamento da *Sprint* [SILVA et al., 2012, p.333].

Com base nessas informações, Hong, Yoo e Cha (2010, p.312) propuseram organizar as responsabilidades da Equipe *Scrum* em cada evento do *Scrum* conforme a Tabela 2.

A Tabela 2 informa para cada evento do *Scrum* quais são as metas do evento, qual é a responsabilidade do *Product Owner*, *Scrum Master* e Equipe de Desenvolvimento no evento do *Scrum*.

Tabela 2: Papéis e responsabilidades em cada evento do *Scrum* adaptado de Hong, Yoo e Cha (2010, p.312).

Evento	Meta do Evento	Product Owner	Scrum Master	Equipe de Desenvolvimento
Reunião de Planejamento	<ul style="list-style-type: none"> • Selecionar os requisitos do <i>Backlog</i> do Produto para desenvolver na <i>Sprint</i> • Criar a meta da <i>Sprint</i> • Dividir os requisitos do <i>Backlog</i> do Produto em tarefas no <i>Backlog</i> da <i>Sprint</i> 	<ul style="list-style-type: none"> • Apresentar os requisitos do <i>Backlog</i> do Produto ordenados • Esclarecer os requisitos selecionados • Criar a meta da <i>Sprint</i> 	<ul style="list-style-type: none"> • Acompanhar a reunião • Criar a meta para a <i>Sprint</i> 	<ul style="list-style-type: none"> • Ratifica os requisitos do <i>Backlog</i> que podem ser completados na <i>Sprint</i> • Criar a meta para a <i>Sprint</i> • Dividir os requisitos do <i>Backlog</i> do Produto em tarefas no <i>Backlog</i> da <i>Sprint</i>
Reunião Diária	<ul style="list-style-type: none"> • Compartilhar o progresso individual da Equipe de Desenvolvimento • Verificar os impedimentos • Verificar progresso em relação a meta da <i>Sprint</i> 	<ul style="list-style-type: none"> • Não participa 	<ul style="list-style-type: none"> • Garantir que a reunião aconteça • Gerenciar os impedimentos 	<ul style="list-style-type: none"> • Compartilhar o progresso • Compartilhar os impedimentos • Verificar progresso em relação a meta da <i>Sprint</i>
Revisão da Sprint	<ul style="list-style-type: none"> • Verificar os requisitos que estão “Prontos” • Discutir como foi a <i>Sprint</i> • Atualizar o <i>Backlog</i> do Produto 	<ul style="list-style-type: none"> • Verificar os requisitos que estão “Prontos” • Atualizar o <i>Backlog</i> do Produto 	<ul style="list-style-type: none"> • Acompanhar a reunião 	<ul style="list-style-type: none"> • Mostrar requisitos que estão “Prontos” • Discutir como foi a <i>Sprint</i>
Retrospectiva da Sprint	<ul style="list-style-type: none"> • Discutir como foi a última <i>Sprint</i> em relação as pessoas, processos e ferramentas • Identificar e ordenar itens com melhorias 	<ul style="list-style-type: none"> • Discutir como foi a última <i>Sprint</i> em relação as pessoas, processos e ferramentas • Identificar e ordenar itens com melhorias 	<ul style="list-style-type: none"> • Discutir como foi a última <i>Sprint</i> em relação as pessoas, processos e ferramentas • Identificar e ordenar itens com melhorias 	<ul style="list-style-type: none"> • Discutir como foi a última <i>Sprint</i> em relação as pessoas, processos e ferramentas • Identificar e ordenar itens com melhorias

4.2. Tamanho da Equipe de desenvolvimento

Dimensionar a Equipe de Desenvolvimento de forma a conseguir completar o trabalho com qualidade e que não fique difícil o gerenciamento.

A Equipe de Desenvolvimento deve ser dimensionada de forma a conseguir completar o trabalho com qualidade. Essa quantidade deve ser entre três e nove integrantes [SCHWABER, SUTHERLAND, 2011, p.6].

Equipes menores do que três membros podem enfrentar restrições de habilidades o que pode afetar a capacidade de entregar um incremento potencialmente usável. Equipes maiores exige muita coordenação porque geram muita complexidade para gerenciar [SCHWABER, SUTHERLAND, 2011, p.6].

4.3. Artefatos do *Scrum*

Usar *Story Points* para dimensionar a complexidade para cada requisito do *Backlog* do Produto.

O Guia do *Scrum* não informa a forma como deve ser estimado o *Backlog* do Produto mas é comum estimar os requisitos do *Backlog* do Produto com *Story Points* [SUTHERLAND, 2010, p.19].

Story Points são unidades de medida para expressar o tamanho de uma história, característica ou outra peça de trabalho. Uma história que possui um valor dois, possui o dobro de complexidade de uma história que possui o valor um. Não existe uma formula para definir o valor para uma história porque o valor é uma estimativa da quantidade de esforço envolvido no desenvolvimento da função, a complexidade do desenvolvimento, o risco inerente a história e assim por diante [COHN, 2006, p.36].

Uma forma para definir os pontos para os requisitos do *Backlog* do Produto é selecionar a história mais simples e atribuir o valor um e as próximas histórias receberão os pontos utilizando como base a história mais simples [COHN, 2006, p.36].

Para fazer as estimativas, a ordem dos números podem ser a série de *Fibonacci*: 01, 1, 2, 3, 5, 8... (um número na sequência *Fibonacci* é gerado com a soma dos últimos dois números) porque existe uma diferença maior entre cada número a cada vez que ele aumenta. Quando o integrante está pensando em um número ele aloca o mais próximo possível desta série de números que ele possui [COHN, 2006, p.52].

Uma forma de fazer as estimativas é utilizando o *Planning Poker*, para fazê-lo cada integrante da Equipe de Desenvolvimento possui um conjunto de cartas identificadas com numerações diferentes [COHN, 2006, p.56].

Para cada história que será estimada o moderador lê a descrição da história. O moderador é normalmente o *Product Owner* mas pode ser qualquer um da Equipe *Scrum*. O *Product Owner* responde todas as questões que os integrantes façam referente ao requisito que estão estimando [COHN, 2006, p.57].

Após todas as questões serem resolvidas, cada integrante seleciona uma carta que não é mostrada até que cada integrante faça sua seleção. Após todos escolherem as cartas, todas são mostradas ao mesmo tempo [COHN, 2006, p.57].

Pode acontecer de as estimativas serem muito diferentes, neste caso as pessoas que fizeram as estimativas maiores e menores explicam por que escolheram este número e depois as estimativas são novamente feitas até que os números fiquem próximos. Quando a diferença for pouca (por exemplo, com quatro integrantes e escolheram 5, 5, 5, 3) é perguntado para a pessoa que estimou 3 se ela sente segura com uma estimativa de 5 pontos [COHN, 2006, p.57].

A vantagem de dimensionar o *Backlog* do Produto com *Story Points* é que toda a Equipe de Desenvolvimento concorda que determinada história possui determinados pontos, a diferença está em quantos pontos a Equipe de Desenvolvimento consegue entregar por *Sprint* [COHN, 2006, p.73-74].

Depois de algumas *Sprints* é possível identificar quantas *Story Points* a Equipe de Desenvolvimento consegue entregar a cada *Sprint* e com isso passa a ser possível prever uma data para a entrega de todas os requisitos do *Backlog* do

Produto ou quantos requisitos do *Backlog* do Produto podem ser completados até determinada data [SUTHERLAND, 2010, p.19].

Criar um formato padrão para especificar os requisitos do *Backlog* do Produto.

Os métodos ágeis não utilizam um modelo documentado de requisitos, o que dificulta uma análise de uma quantidade grande de requisitos e sem isso o conhecimento do projeto fica somente com as pessoas que participaram das reuniões o que pode ficar difícil de repassar para uma equipe maior [READ, BRIGGS, 2012, p.5319].

Para superar esse problema, deve-se criar Hiper-Épicos e Hiper-Estórias. Um Hiper-Épico contém um conjunto de estórias relacionadas, com parâmetros compartilhados e uma lista de regras de negócio para cada estória que clarifica o comportamento esperado da função [READ, BRIGGS, 2012, p.5324].

Uma Hiper-Estória é o detalhamento de um requisito e sua estrutura é definida em 5 partes: Em algum lugar, uma pessoa baseado em alguma regra faz alguma ação, para atingir algum resultado que produz algum efeito. Esta estrutura é detalhada na Tabela 3 [READ, BRIGGS, 2012, p.5321].

Tabela 3: Detalhamento dos requisitos para a criação de uma Hiper-Estória adaptado de Read e Briggs (2012, p.5321-5322).

<i>Item</i>	<i>Descrição</i>
Lugar	Em qual parte/tela do sistema a estória está envolvida.
Regra	Tipo do usuário que poderá executar a ação.
Ação	O que o usuário irá fazer no sistema.
Resultado	Motivo pelo qual o usuário quer executar a ação.
Efeito	O que irá acontecer quando o usuário executar a ação

Antes de criar as Hiper-Estórias é realizada uma reunião com as partes interessadas para discutir as características do produto que precisam ser desenvolvidas e para cada requisito é criado um título e um rascunho da solicitação, por exemplo “Efetuar *login* no sistema” [READ, BRIGGS, 2012, p.5323].

Após os requisitos do *Backlog* do Produto estarem definidos, é feita uma reunião com o *Product Owner* e o *Scrum Master* para detalhar e criar as Hiper-Estórias.

Aspectos de uma Hiper-Estória que se repetem em outras estórias podem ser abstraídas em parâmetros. Esses parâmetros devem ser inseridos entre colchetes. A Figura 2 mostra um exemplo de Hiper-Épico [READ, BRIGGS, 2012, p.5322].

ÉPICO: Abrir e fechar elementos do editor

ESTÓRIA: Abrir elementos do editor

No [editor], o usuário [abre] o editor de elementos do [objeto] para configurar as características do [objeto]. O editor de elementos do [objeto] aparece em uma nova aba no painel do [editor]. Campos que possuem valores aparecem preenchidos. Campos sem valores aparecem vazios.

REGRAS DE NEGÓCIO

A primeira vez que o editor de elementos para um [objeto] for aberto, valores padrão serão preenchidos para todos os campos.

FIM DAS REGRAS

FIM DA ESTÓRIA

ESTÓRIA: Fechar um elemento do editor

No [editor], um usuário [fecha] um [objeto] do editor de elementos para esconder as opções configuráveis do [objeto] para reduzir a carga visual. O editor de elementos para o [objeto] é fechado. Os valores de todos os campos no editor de elementos são gravados no servidor.

REGRAS DE NEGÓCIO

O usuário não pode fechar o editor de elementos até que todos os campos estejam visualizados.

FIM DAS REGRAS

FIM DA ESTÓRIA

PARÂMETROS

[editor]: Editor de utilitários, Editor de tela

[fecha]: Clicar no botão "X" na aba do editor de elementos; Clicar com o botão direito do mouse na aba do elemento e selecionar fechar; Clicar no botão fechar no editor de elementos

[objeto]: Utilitário; Controle

[abre]: Duplo clique com o botão esquerdo do mouse; Clicar com o botão direito do mouse e selecionar "Editar"; Clicar no menu "Arquivo" e selecionar "Editar"

FIM DOS PARÂMETROS

FIM DO ÉPICO

Figura 2: Exemplo de um Hiper-Épico adaptado de Read e Briggs (2012, p.5322).

Hiper-Estórias relacionadas são normalmente desenvolvidas em paralelo como um Hiper-Épico como mostrado na Figura 2. Uma lista única de parâmetros de todas as estórias aparecem no fim do Hiper-Épico. Um Hiper-Épico normalmente possui entre cinco e quinze Hiper-Estórias [READ, BRIGGS, 2012, p.5322].

Criar o *Backlog* do Produto com a lista de requisitos do produto.

O *Backlog* do Produto é uma lista priorizada de requisitos que o produto deverá possuir. Somente existirá um *Backlog* do Produto e conterá uma visão definitiva de tudo o que poderá se tornar Pronto pela Equipe de Desenvolvimento e deverá estar priorizado [SUTHERLAND, 2010, p.18].

O *Product Owner* é o responsável por definir todos os requisitos do produto no *Backlog* do Produto. Ele utiliza esta lista para priorizar os itens que agreguem mais valor ao produto. Esta priorização é feita frequentemente para garantir que os itens mais importantes sejam entregues a cada Sprint [SUTHERLAND, 2010, p.14].

Apesar de o *Product Owner* ser o responsável pelo *Backlog* do Produto, a Equipe de Desenvolvimento é a responsável por fornecer para o *Product Owner* os *Story Points* necessários para cada requisito do *Backlog* do Produto [SUTHERLAND, 2010, p.19].

O *Backlog* do Produto inclui uma variedade de requisitos, principalmente novas funções como: “Permitir todos os usuários inserir livros no carrinho” mas também poderá possuir itens de melhorias como: “Refazer o módulo de processamento de transação para torná-lo escalável”, itens para pesquisar como: “Pesquisar soluções para aumentar a velocidade de validação do cartão de crédito” ou ainda itens de desempenho, segurança ou defeitos conhecidos como: “Diagnosticar e corrigir os erros *script* para processamento de pedidos” [SUTHERLAND, 2010, p.18-19].

Os requisitos devem ser escritos de forma clara sobre a função e deve ser atualizado constantemente pelo *Product Owner* para refletir as mudanças requisitadas pelo cliente, novas idéias, dificuldades técnicas que aparecem e assim por diante [SUTHERLAND, 2010, p.19].

A Tabela 4 mostra um exemplo de *Backlog* do Produto. A função de cada coluna é [SUTHERLAND, 2010, p.18]:

- **ID:** Número sequencial para identificar o requisito para ser usado no *Backlog* da *Sprint* ou em outro lugar que precise referenciar o requisito;
- **Requisito:** Descreve de forma sucinta os itens que devem ser desenvolvidos;
- **Detalhes:** Possui um *link* para mostrar a descrição detalhada da estória;
- **Story Points:** São as estimativas de esforço para o desenvolvimento da função fornecidos pela Equipe de Desenvolvimento;
- **Prioridade:** Informa qual é a prioridade de desenvolvimento do requisito.

Tabela 4: Exemplo de um *Backlog* do Produto adaptado de Sutherland (2010, p.18).

ID	Requisito	Detalhes do Requisito	Story Points	Prioridade
1	Como comprador, eu quero inserir um livro no carrinho.	...	5	1
2	Como comprador, eu quero remover um livro do carrinho.	...	2	2
3	Aumentar o desempenho no processamento da transação.	...	13	3
4	Pesquisar soluções para aumentar a velocidade na validação do cartão de crédito.	...	20	4
5	Atualizar todos os servidores para o <i>Apache 2.2.3</i>	13	5
6	Diagnosticar e corrigir os erros de <i>script</i> para processamento do pedido.	...	3	6

Criar o *Backlog* da *Sprint* com base nos requisitos selecionados na Reunião de Planejamento.

Após a seleção dos requisitos que serão desenvolvidos na *Sprint*, eles são detalhados em tarefas menores que são inseridos no *Backlog* da *Sprint* [SUTHERLAND, 2010, p.20-21].

Como já foi mencionado, o *Product Owner* pode estar presente para esclarecer os requisitos do *Backlog* do Produto selecionados. Ao término da criação do *Backlog* da *Sprint*, a Equipe de Desenvolvimento possuirá uma lista de tarefas com estimativas em horas [SUTHERLAND, 2010, p.22].

Esta lista é um ponto de partida porque outras tarefas podem aparecer conforme estão desenvolvendo as tarefas do *Backlog* da *Sprint*. Para desenvolver estas tarefas, a Equipe de Desenvolvimento escolhe a ordem para poder aumentar a velocidade de produção e qualidade do produto Pronto [SUTHERLAND, 2010, p.22].

A Tabela 5 mostra um exemplo de *Backlog* da *Sprint*. A função de cada coluna é [SUTHERLAND, 2010, p.22]:

- **ID Requisito:** Informa qual é o requisito que determinada tarefa faz parte;
- **Tarefa da *Sprint*:** Informa qual é a tarefa que será desenvolvida;
- **Voluntário:** Informa qual é o integrante da Equipe de Desenvolvimento que irá desenvolver a tarefa;
- **Estimativa de esforço:** Estimativa em horas que serão necessários para desenvolver a tarefa;

Tabela 5: Exemplo de um *Backlog* da *Sprint* adaptado de Sutherland (2010, p.22).

ID Requisito	Tarefa da <i>Sprint</i>	Voluntário	Estimativa de esforço
1	Modificar banco de dados		5
	Criar a página (<i>design</i>)		8
	Criar a página (lógica <i>Javascript</i>)		13
	Escrever testes de aceitação automatizados		13
	Atualizar a página de ajuda do comprador		3
3	Mesclar código <i>DCP</i> e completar os testes por camada		5
	Completar a ordem de máquina <i>pRank</i>		8
	Alterar o <i>DCP</i> e o leitor para usar a <i>API pRank http</i>		13

Melhorar o desempenho da Equipe de Desenvolvimento usando *Dashboard* e *Task Cards*.




O processo referente ao desenvolvimento da *Sprint* foi aperfeiçoado para evitar que algum membro da Equipe de Desenvolvimento fique ocioso quando existe uma diferença na conclusão das tarefas entre os membros inserindo um *Dashboard* e os *Tasks Cards* [HONG, YOO, CHA, 2010, p.313].

Quando um *Task Card* é criado, ele é colocado próximo a data esperada de início no *Dashboard*. O *Task Card* possui duas informações importantes: O *ID* do requisito no *Backlog* do Produto e a data estimada para a conclusão da tarefa. O *ID* permite a rastreabilidade do desenvolvimento do produto entre todos os membros da equipe [HONG, YOO, CHA, 2010, p.313].

A data de conclusão permite que os outros membros se planejem para diminuir a ociosidade enquanto outra tarefa está sendo completada. Por exemplo: Um desenvolvedor somente consegue iniciar a codificação de uma página quando o projeto da página estiver completo. [HONG, YOO, CHA, 2010, p.313].

Hong, Yoo e Cha (2010, p.3) mostram na Figura 3 como é feito o *Dashboard* e o *Task Card*. A função de cada coluna é:

- **ID Requisito:** Informa qual é o requisito que determinada tarefa faz parte;
- **A fazer:** Exibe os *Task Cards* com previsão de início a cada dia da *Sprint*;
- **Em progresso:** Exibe os *Task Cards* em desenvolvimento a cada dia da *Sprint*;
- **Feito:** Informa os *Task Cards* estão finalizados em cada dia da *Sprint*.

ID Requisito	A fazer		Em progresso	Feito	
#Nº requisito	01/02/2013 Data esperada de início	08/02/2013 	Segunda	01/02/2013 Data em que foi completado	08/02/2013
#Nº requisito	04/02/2013 	11/02/2013 	Terça	04/02/2013	11/02/2013
#Nº requisito	05/02/2013 	12/02/2013 	Quarta	05/02/2013	12/02/2013
#Nº requisito	06/02/2013 	13/02/2013 	Quinta	06/02/2013	13/02/2013
#Nº requisito	07/02/2013 	14/02/2013 	Sexta	07/02/2013	14/02/2013

Definição do Task Card

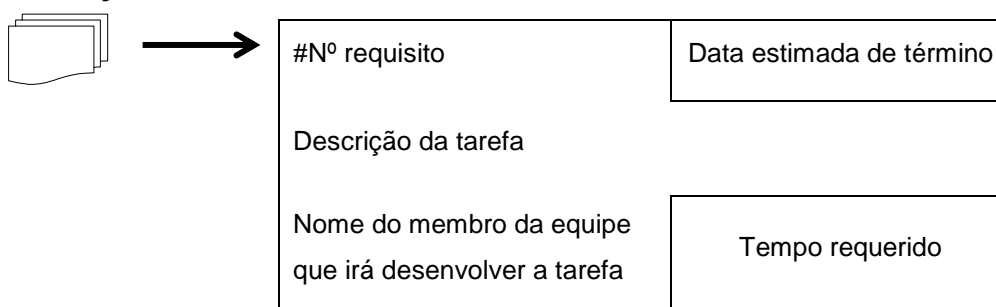


Figura 3: Dashboard e Task Card adaptado de Hong, Yoo, Cha (2010, p.313).

Criar um gráfico *Burndown* para acompanhar o desenvolvimento do Backlog do Produto.

O gráfico *Burndown* é usado para acompanhar o progresso do desenvolvimento do produto e é responsabilidade do *Product Owner* garantir que o gráfico esteja sempre atualizado. Caso o *Product Owner* não saiba como criar este gráfico, é responsabilidade do *Scrum Master* ajudá-lo [SUTHERLAND, 2010, p.30].

A Figura 4 mostra um exemplo de gráfico *Burndown* adaptado de Cohn (2006, p.219-220).

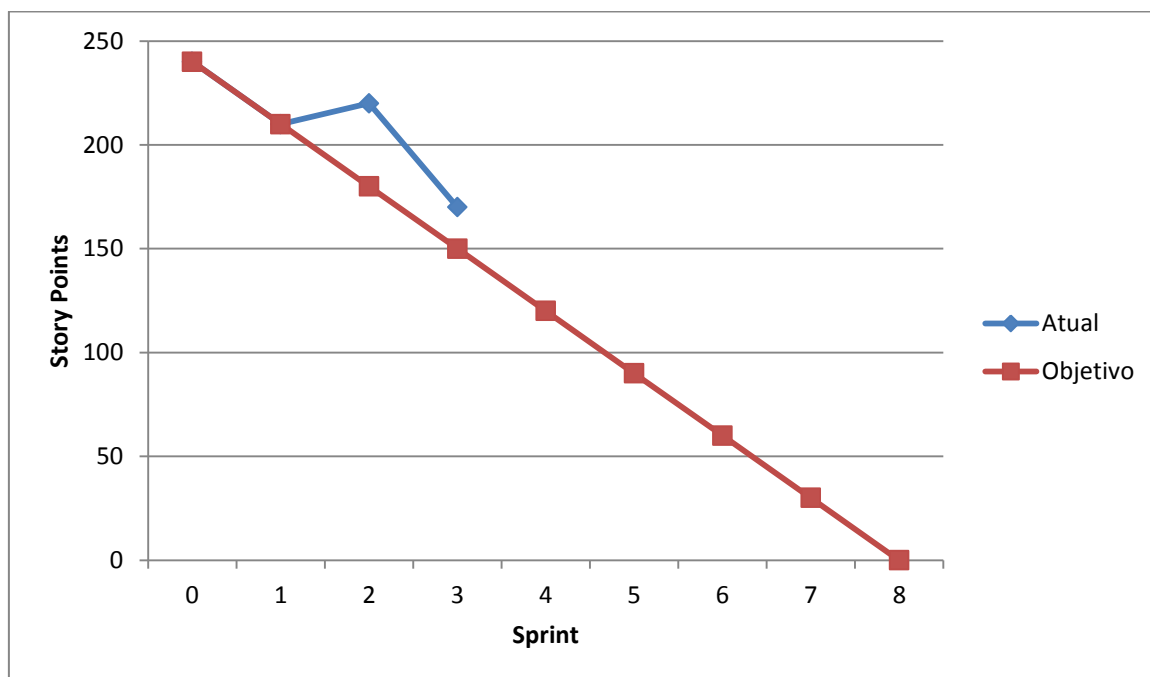


Figura 4: Exemplo de um gráfico de *Burndown* adaptado de Cohn (2006, p.219-220).

A Figura 4 mostra um gráfico *Burndown* para um projeto com 240 *Story Points* e foram distribuídos 30 pontos igualmente em todas as *Sprints*. Não se deve esperar que a Equipe de Desenvolvimento tenha uma velocidade exata de 30 pontos em todas as *Sprints*, então seu progresso não será consistente conforme é mostrado a Figura 4 [COHN, 2006, p. 219].

Na primeira *Sprint*, o esperado de 30 *Story Points* foi completado mas no fim da segunda *Sprint* sobrou mais trabalho do que quando começaram a primeira *Sprint*. Isto pode acontecer, por exemplo, se a Equipe de Desenvolvimento estimou pouco o esforço necessário para a *estória* e os pontos acabaram aumentando [COHN, 2006, p. 219].

Um outro exemplo é se o *Product Owner* identificou uma outra *estória* que precisa ser entregue, então eles tiveram mais trabalho na *Sprint* do que o previsto. Como o gráfico *Burndown* reflete o progresso da Equipe de Desenvolvimento o gráfico mostra essa diferença [COHN, 2006, p. 220].

4.4. Eventos da *Sprint*

Realizar a reunião *Master Sprint Plan* antes da Reunião de Planejamento para ter uma previsão de quantas *Sprints* o projeto terá.

Durante a Reunião de Planejamento, o *Scrum* possui foco somente nas *Sprints* individuais, o que dificulta criar metas e ter uma visão da quantidade de *Sprints* do projeto [HONG, YOO, CHA, 2010, p.312].

Hong, Yoo e Cha (2010, p.312) propõem criar uma reunião antes da Reunião de Planejamento chamada de *Master Sprint Plan* para poder planejar o projeto como um todo. Nesta reunião são estimados o tempo e a quantidade de *Sprints* necessárias para poder completar o projeto. Nesta reunião participam o *Scrum Master*, *Product Owner* a Equipe de Desenvolvimento.

Antes desta reunião, o *Product Owner* cria o *Backlog* do Produto e durante a reunião a Equipe de Desenvolvimento estima os *Story Points* do *Backlog* do Produto de acordo com sua experiência. Assim que a reunião termina é verificado se o escopo é factível com o planejamento [HONG, YOO, CHA, 2010, p.312-313].

Discutir e esclarecer na Reunião de Planejamento os requisitos selecionados do *Backlog* do Produto para a *Sprint*.

Esta reunião deve durar 4 horas para uma *Sprint* de um mês, caso a *Sprint* possua um tempo menor, a duração da reunião deve ser proporcional ao tempo da *Sprint* [SCHWABER, SUTHERLAND, 2011, p.9].

Na primeira parte da Reunião de Planejamento são discutidos os requisitos do *Backlog* do Produto que serão desenvolvidos na *Sprint*. O *Product Owner* apresenta o *Backlog* do Produto ordenado e esclarece os requisitos que precisam ser desenvolvidos para toda a Equipe *Scrum*. A Equipe de Desenvolvimento ratifica os requisitos que podem ser concluídos na *Sprint* [SCHWABER, SUTHERLAND, 2011, p.9] [HONG, YOO, CHA, 2010, p.312-313].

Criar a meta da *Sprint* na Reunião de Planejamento da *Sprint*.

Na primeira parte da reunião são selecionados os requisitos do *Backlog* do Produto que podem ser desenvolvidos na *Sprint* e também é criada a meta da *Sprint*. Caso aconteça de o andamento do trabalho da Equipe de Desenvolvimento ser diferente do previsto, ou seja, aparecer alguma tarefa que não estava programada, a Equipe de Desenvolvimento discute com o *Product Owner* o escopo do *Backlog* da *Sprint* dentro da *Sprint* [SCHWABER, SUTHERLAND, 2011, p.10].

Um exemplo de meta da *Sprint*: Como um usuário quero ser capaz de mapear uma pasta na rede, assim o sistema poderá importar arquivos de diferentes locais.

Definir na Reunião de Planejamento como a Equipe de Desenvolvimento irá desenvolver os requisitos selecionados do *Backlog* do Produto na *Sprint*.

Esta reunião deve durar 4 horas para uma *Sprint* de um mês, caso a *Sprint* possua um tempo menor, a duração da reunião deve ser proporcional ao tempo da *Sprint* [SCHWABER, SUTHERLAND, 2011, p.9].

Na segunda parte da Reunião de Planejamento, a Equipe de Desenvolvimento deve planejar como desenvolver as tarefas do *Backlog* da *Sprint* e transformá-los em um incremento do produto Pronto. O *Product Owner* pode participar desta reunião para esclarecer os requisitos selecionados do *Backlog* do Produto e caso a Equipe de Desenvolvimento verifique que exista um excesso ou falta de trabalho para a *Sprint*, a quantidade de requisitos pode ser renegociado com o *Product Owner* [SCHWABER, SUTHERLAND, 2011, p.10].

Discutir na Reunião Diária o que foi feito, o que será feito e os impedimentos que estão enfrentando.

Esta reunião deve durar 15 minutos e a Equipe de Desenvolvimento tem como foco discutir o que foi feito desde a última reunião, o que será feito até a próxima reunião e os impedimentos que estão tendo para desenvolver suas tarefas [SCHWABER, SUTHERLAND, 2011, p.11].

O progresso da *Sprint* é verificado nesta reunião em relação à meta da *Sprint* e deve-se avaliar se, com o progresso atual, será possível completar o trabalho do *Backlog* da *Sprint* [SCHWABER, SUTHERLAND, 2011, p.11].

O *Scrum Master* participa desta reunião para garantir que ela aconteça e caso seja detectado algum impedimento que a Equipe de Desenvolvimento esteja tendo para desempenhar suas tarefas, o *Scrum Master* irá ajudá-los a resolver [SCHWABER, SUTHERLAND, 2011, p.11].

Evitar o cancelamento de uma *Sprint*.

Uma *Sprint* poderá ser cancelada antes do término somente pelo *Product Owner*, porém ele poderá fazer isto sob influência das partes interessadas, Equipe de Desenvolvimento ou *Scrum Master* [SCHWABER, SUTHERLAND, 2011, p.8-9].

A *Sprint* somente poderá ser cancelada se alguma circunstância externa alterar as prioridades e a Equipe de Desenvolvimento estará pendendo tempo se continuar com a *Sprint* atual [SCHWABER, SUTHERLAND, 2011, p.9] [SUTHERLAND, 2010, p.23-24].

Caso a *Sprint* seja cancelada, todos os requisitos do *Backlog* do Produto que estejam “Prontos” são revisados. Todos os requisitos incompletos devem ser estimados novamente e devolvidos ao *Backlog* do Produto [SCHWABER, SUTHERLAND, 2011, p.9].

Revisar a *Sprint* desenvolvida pela Equipe de Desenvolvimento para decidir o que está Pronto.

Esta reunião deve durar 4 horas para uma *Sprint* de um mês, proporcionalmente menor caso a *Sprint* tenha duração menor [SCHWABER, SUTHERLAND, 2011, p.11].

No fim da *Sprint*, a Equipe *Scrum* e as partes interessadas devem se reunir para inspecionar o incremento e atualizar o *Backlog* do Produto caso seja necessário. A

Equipe de Desenvolvimento apresenta o trabalho que está Pronto, o *Product Owner* identifica o que está Pronto e o que não está Pronto e discute o *Backlog* do Produto da forma como está projetando as prováveis datas de conclusão baseado no progresso até o momento [SCHWABER, SUTHERLAND, 2011, p.11-12].

Nesta reunião, a Equipe de Desenvolvimento deve informar como foi o andamento da *Sprint*, o que foi bem, os problemas que tiveram e como foram resolvidos referente ao desenvolvimento do produto [SCHWABER, SUTHERLAND, 2011, p.12] [SUTHERLAND, 2010, p.28].

Discutir na reunião de Retrospectiva da *Sprint* o resultado da última *Sprint* em relação ao processo.

A reunião de Revisão da *Sprint* tem como foco a inspeção e adaptação do produto, já a reunião de Retrospectiva da *Sprint* tem como foco a inspeção e adaptação do processo [SUTHERLAND, 2010, p.28].

Esta reunião deve durar 3 horas para uma *Sprint* de um mês, proporcionalmente menor caso a *Sprint* tenha duração menor e serve para a Equipe *Scrum* inspecionar como a última *Sprint* foi em relação as pessoas, processos e ferramentas, identificar e ordenar os principais itens em relação ao processo que foram bem, o que pode ser melhorado e criar um plano para implementar melhorias na forma com que a Equipe *Scrum* faz seu trabalho [SCHWABER, SUTHERLAND, 2011, p.12].

Nesta reunião são discutidas as formas para melhorar a qualidade do produto e adaptar a definição de Pronto caso seja necessário. No fim da reunião, a Equipe *Scrum* terá uma lista de melhorias que serão implementadas na próxima *Sprint* [SCHWABER, SUTHERLAND, 2011, p.12].

4.5. Resumo das diretrizes

A Tabela 6 mostra todas as diretrizes apresentadas neste capítulo em cada categoria.

Tabela 6: Diretrizes apresentadas no capítulo 4.

<i>Categoria</i>	<i>Diretriz</i>
Papéis do Scrum	Entender as responsabilidades de cada papel.
Tamanho da Equipe de desenvolvimento	Dimensionar a Equipe de Desenvolvimento de forma a conseguir completar o trabalho com qualidade e que não fique difícil o gerenciamento.
Artefatos do Scrum	Usar <i>Story Points</i> para dimensionar a complexidade para cada requisito do <i>Backlog</i> do Produto.
	Criar um formato padrão para especificar os requisitos do <i>Backlog</i> do Produto.
	Criar o <i>Backlog</i> do Produto com a lista de requisitos do produto.
	Criar o <i>Backlog</i> da <i>Sprint</i> com base nos requisitos selecionados na Reunião de Planejamento.
	Melhorar o desempenho da Equipe de Desenvolvimento usando <i>Dashboard</i> e <i>Task Cards</i> .
	Criar um gráfico <i>Burndown</i> para acompanhar o desenvolvimento do <i>Backlog</i> do Produto.
Eventos da Sprint	Realizar a reunião <i>Master Sprint Plan</i> antes da Reunião de Planejamento para ter uma previsão de quantas <i>Sprints</i> o projeto terá.
	Discutir e esclarecer na Reunião de Planejamento as tarefas selecionadas do <i>Backlog</i> do Produto para a <i>Sprint</i> .
	Criar a meta da <i>Sprint</i> na Reunião de Planejamento da <i>Sprint</i> .
	Definir na Reunião de Planejamento como a Equipe de Desenvolvimento irá desenvolver os requisitos selecionados do <i>Backlog</i> do Produto na <i>Sprint</i> .
	Discutir na Reunião Diária o que foi feito, o que será feito e os impedimentos que estão enfrentando.
	Evitar o cancelamento de uma <i>Sprint</i> .
	Revisar a <i>Sprint</i> desenvolvida pela Equipe de Desenvolvimento para decidir o que está Pronto.
	Discutir na reunião de Retrospectiva da <i>Sprint</i> o resultado da última <i>Sprint</i> em relação ao processo.

4.6. Considerações do Capítulo

As diretrizes descritas neste capítulo foram criadas para facilitar a implantação do *Scrum* seguindo as análises feitas na Tabela 7 e sua estrutura foi feita pensando no fluxo dos acontecimentos do *Scrum*.

A implantação do *Scrum* com as diretrizes descritas neste capítulo irá fazer com que o gerenciamento do desenvolvimento do produto seja mais completo porque além de seguir as diretrizes do Guia do *Scrum*, também contém informações valiosas de como resolver problemas que foram relatados no capítulo 3.

5. CONSIDERAÇÕES FINAIS

O Scrum é o método ágil mais usado para gerenciar o desenvolvimento de produtos complexos possuindo em sua estrutura artefatos, definições de papéis e eventos com duração pré-determinada.

Apesar de o Scrum estar bem definido no Guia do Scrum, nem sempre é possível realizar sua implantação sem que seja necessário complementá-lo com novas práticas para torná-lo mais completo.

O objetivo deste trabalho foi apresentar o Scrum conforme descrito no Guia do Scrum, apresentar relatos de implantação mostrando os problemas que tiveram, como resolveram e criar diretrizes para a implantação do Scrum complementando com os relatos das referências.

Apesar do esforço para juntar novas práticas para as diretrizes, foi verificado que existe uma escassez de publicações referente a adição de novas práticas no *Scrum*.

5.1. Contribuições do Trabalho

O formato da estrutura do Guia do *Scrum* dificulta o entendimento de quem não possui conhecimentos profundos e está implantando o *Scrum*. Falta clareza e algumas definições em alguns momentos, como por exemplo, as responsabilidades dos papéis não estão especificadas em um só capítulo mas ao longo do Guia do *Scrum*, ausência de um exemplo para a criação de um *Backlog* do Produto e da *Sprint* ou ainda não informando como o progresso do desenvolvimento deve ser monitorado.

Foi verificado na literatura que existem dificuldades na implantação do Scrum conforme descrito no Guia do *Scrum*, pois nem sempre as práticas sugeridas resolve todos os problemas do gerenciamento do desenvolvimento do produto.

Baseado nisto, foi criado a Tabela 7 com estudos no Guia do *Scrum* em todos os papéis, eventos e artefatos disponíveis fazendo uma análise comparativa entre o

que é proposto no Guia do *Scrum* e as novas práticas propostas na literatura para poder sugerir as diretrizes que foram criadas.

Em seguida foi criado o capítulo 4 fornecendo as diretrizes para facilitar a implantação do *Scrum* com base na Tabela 7.

5.2. Trabalhos Futuros

Não foi possível realizar uma implantação real e fazer uma avaliação mais rigorosa das diretrizes propostas para verificar sua viabilidade.

Recomenda-se a implantação do método *Scrum* em dois projetos, no primeiro seguindo as definições do Guia do *Scrum* e no segundo seguindo estas diretrizes e avaliar cada item para verificar se os resultados obtidos são positivos em relação à primeira implantação.

Segundo Krishna e Basu (2011), novas abordagens estão surgindo como por exemplo o *ScrumBut* e o *ScrumAnd*, que significam o *Scrum* com menos práticas e o outro com todas as práticas do *Scrum* complementando com novas.

REFERÊNCIAS

Agile Alliance. **What is Agile?** Disponível em: <http://www.agilealliance.org/thealliance/what-is-agile/>. Acessado em 01/12/2012.

COHN, M.. **Agile Estimating and Planning**. Prentice Hall, 2006, 365p.

HONG, N.; YOO, J.; CHA, S.. **Customization of Scrum Methodology for Outsourced E-commerce Projects**. In Proceedings of: Software Engineering Conference (APSEC), 2010 17th Asia Pacific, Dezembro, 2010, p. 310 – 315.

KRISHNA, V.; BASU, A.. **Scrum+ :: Is it ScrumBut or ScrumAnd**. In Proceedings of: India Conference (INDICON), 2011 Annual IEEE, Dezembro, 2011.

MARANZATO, R. P.; NEUBERT, M.; HERCULANO, P.. **Scaling Scrum Step by Step: “The Mega Framework”**. In Proceedings of: Agile Conference (AGILE), 2012, Agosto, 2012, p. 79 – 85.

READ, A.; BRIGGS, R. O.. **The Many Lives of an Agile Story: Design Processes, Design Products, and Understandings in a Large-Scale Agile Development Project**. In Proceedings of: System Science (HICSS), 2012 45th Hawaii International Conference on, Janeiro, 2012, p. 5319 – 5328.

SCHWABER, K.; SUTHERLAND, J.. **Guia do Scrum**. Disponível em: <http://scrum.org/Scrum-Guides>. Acessado em 05/12/2012.

SILVA, A. L. M.; CAVALHEIRO, L. T. A.; ROMAN, N. T.; CHAIM, M. L.. **Implementação de Metodologia de Desenvolvimento Ágil em Projetos com Time Alocado e Não Alocado**. Anais do: Simpósio Brasileiro de Sistemas de Informação, 2012, p. 327 – 336.

SUTHERLAND, J.. **Scrum Handbook**. Disponível em: <http://jeffsutherland.com/scrumhandbook.pdf>. Acessado em: 20/12/2012.

APÊNDICE 1 – TABELA BASE PARA ELABORAÇÃO DAS DIRETRIZES

A Tabela 7 foi criada com o objetivo de servir como um guia para criar as diretrizes baseada em estudos feitos nos papéis, eventos e artefatos contidos no Guia do *Scrum* fazendo uma análise comparativa com as novas práticas propostas na literatura.

Para cada papel, evento e artefato existente no Guia do *Scrum*, existem comentários referente ao que diz o Guia do *Scrum*, artigos, observações pessoais e prováveis diretrizes.

Tabela 7: Tabela base para elaboração das diretrizes

<i>Papéis do Scrum</i>			
	<i>Product Owner</i>	<i>Equipe de Desenvolvimento</i>	<i>Scrum Master</i>
Guia do Scrum	Definir os requisitos do projeto, criar, ordenar e atualizar o <i>Backlog</i> do Produto, divulgar o <i>Backlog</i> do Produto, garantir que os requisitos do <i>Backlog</i> do Produto estejam claros e garantir o valor do trabalho realizado pela Equipe de Desenvolvimento.	Pessoas auto-organizadas e multifuncionais que irão criar o produto que será entregue para o cliente, compartilhar os detalhes das tarefas que estão desenvolvendo.	Pessoa responsável em garantir que as práticas do <i>Scrum</i> sejam seguidas, resolver problemas que podem estar impedindo a Equipe de Desenvolvimento de concluir o trabalho, garantir que as reuniões aconteçam, trabalhar com outros <i>Scrum Masters</i> para aumentar a eficácia da aplicação do <i>Scrum</i> .
Artigos	<p>É importante realizar uma reunião semanal com todos os <i>Products Owners</i> e <i>Scrum Masters</i> da empresa para poder discutir sobre os impedimentos, características que podem estar em desenvolvimento ou planejadas que afetem outras equipes, problemas pendentes, regras organizacionais e decisões importantes [MARANZATO, NEUBERT, HERCULADO, 2012, p.83].</p> <p>Não é claro o suficiente as responsabilidades de cada papel durante cada evento do Scrum. Isso pode fazer com que as pessoas não trabalhem de forma adequada [HONG, YOO, CHA, 2010, p.310].</p>		

Observações	<p>Não é explicado no Guia do <i>Scrum</i> como o <i>Scrum Master</i> trabalha com outros <i>Scrum Masters</i> para dividir experiências, discutir as regras da organização que possam afetar as equipes <i>Scrum</i> e nem como e quando isso é aplicado na sua equipe.</p> <p>No Guia do <i>Scrum</i> não existe uma tabela exibindo as responsabilidades dos papéis nas etapas do. As responsabilidades estão distribuídas ao longo do Guia do <i>Scrum</i> e quem estiver adotando o <i>Scrum</i> pode não ter uma visão clara e com isso não adotar da melhor forma possível.</p>
Diretrizes	<ul style="list-style-type: none"> - O <i>Scrum Master</i> também trabalha para a organização liderando, treinando e ajudando a todos sobre a adoção do <i>Scrum</i>, planejando implementações, causando mudanças que aumentem a produtividade da Equipe <i>Scrum</i> e trabalhando com outros <i>Scrum Masters</i> para aumentar a eficácia da aplicação do <i>Scrum</i> na organização; - Uma <i>Sprint</i> pode ser cancelada somente pelo <i>Product Owner</i> podendo ter influência das partes interessadas, <i>Scrum Master</i> ou da Equipe de Desenvolvimento. A <i>Sprint</i> somente poderá ser cancelada caso o objetivo da <i>Sprint</i> não seja mais importante para o produto. No cancelamento da <i>Sprint</i> é verificado o que está Pronto e o que não estiver, volta para o <i>Backlog</i> do Produto e são re-estimados. <p>Fazer uma reunião semanal com os <i>Scrum Masters</i> e <i>Product Owners</i> da organização, sendo que pode aumentar de acordo com a quantidade de equipes <i>Scrum</i> para trocar experiências, aumentar a eficácia da adoção do <i>Scrum</i>, discutir assuntos que afetem mais de uma equipe <i>Scrum</i> como: Impedimentos, características do produto que estão em desenvolvimento ou planejadas para serem desenvolvidas, pendências, regras organizacionais e decisões.</p> <p>As responsabilidades de cada papel em cada fase do projeto está distribuído ao longo do Guia do <i>Scrum</i>, o que não traz uma visão clara e imediata dessas responsabilidades em cada evento do <i>Scrum</i>.</p> <p>As empresas que estão adotando o <i>Scrum</i> podem ficar confusas com as responsabilidades então faz-se necessário criar uma tabela com as responsabilidades de cada papel durante cada fase do projeto para garantir que todos entendam as regras e consigam fazer seu trabalho adequadamente.</p>

	Para garantir que toda a Equipe <i>Scrum</i> entenda o funcionamento e as responsabilidades, pode-se também certificar toda a equipe como <i>Professional Scrum Master I</i> para poder explicar todos os artefatos do <i>Scrum</i> para que seja implantada da melhor forma possível. Como a equipe foi certificada, o ideal é também certificar o <i>Product Owner</i> como <i>Professional Scrum Product Owner</i> para que ele entenda claramente a importância do seu papel no projeto.
Equipe	
	Tamanho da Equipe de Desenvolvimento
Guia do Scrum	A Equipe de Desenvolvimento deve ser pequena o suficiente para se manter ágil e grande o suficiente para completar uma parcela significativa do trabalho. O ideal é entre 3 e 9 integrantes.
Artigos	Não aplicável
Observações	Não aplicável
Diretrizes	<p>A Equipe de Desenvolvimento deve ser dimensionada de forma a conseguir completar seu trabalho com qualidade. O ideal é que essa quantidade seja entre 3 e 9 integrantes. Equipes menores do que 3 porque podem enfrentar restrições de habilidades e equipes maiores do que 9 exige muita coordenação porque geram muita complexidade para gerenciar.</p> <p>Os papéis do <i>Product Owner</i> e <i>Scrum Master</i> não entram nessa contagem a não ser que eles também executem o trabalho de atualização do <i>Backlog</i> da <i>Sprint</i>.</p>

Eventos da Sprint	
	Reunião de Planejamento
Guia do Scrum	<p>Parte 1: Entender o que o <i>Product Owner</i> precisa para a <i>Sprint</i> atual baseado na lista de requisitos ordenadas previamente e criar uma meta para a <i>Sprint</i> que será desenvolvida.</p> <p>Parte 2: Equipe de Desenvolvimento transforma os requisitos previamente selecionados do <i>Backlog</i> do <i>Produto</i> em tarefas menores e planejam como irá desenvolvê-los.</p>

Artigos	Na primeira parte da Reunião de Planejamento, o método original do <i>Scrum</i> é focado somente nas <i>Sprints</i> incrementais não permitindo uma visão clara de quantas <i>Sprints</i> serão necessárias para a conclusão do projeto [HONG, YOO, CHA, 2010, p.310].
Observações	Os eventos do <i>Scrum</i> não incluem uma forma de ter uma visão de todas as <i>Sprints</i> na fase inicial do projeto e isso é importante porque é possível prever todos os recursos e esforços necessários para desenvolver o projeto.
Diretrizes	<p>Na primeira parte da reunião são selecionados os requisitos do <i>Backlog</i> do Produto que podem ser desenvolvidas na <i>Sprint</i> e também é criado a meta da <i>Sprint</i>. Caso o trabalho seja diferente do esperado pela Equipe de Desenvolvimento, é negociado com o <i>Product Owner</i> o escopo do <i>Backlog</i> da <i>Sprint</i> atual.</p> <p>O <i>Scrum</i> possui foco somente nas <i>Sprints</i> individuais o que dificulta criar metas e ter uma visão da quantidade de <i>Sprints</i> do projeto. É proposto criar uma reunião antes da Reunião de Planejamento da <i>Sprint</i> chamada de <i>Master Sprint Plan</i> para poder planejar todo o projeto. Nesta reunião são estimados o tempo e a quantidade de <i>Sprints</i> necessárias para poder completar o projeto. Nesta reunião participam o Scrum Master, Product Owner e a Equipe de Desenvolvimento.</p> <p>Antes dessa reunião, o <i>Product Owner</i> cria o <i>Backlog</i> do Produto e durante a reunião a Equipe de Desenvolvimento estima o <i>Backlog</i> do Produto de acordo com sua experiência. Assim que a reunião termina é verificado se o escopo é factível com o planejamento.</p> <p>Na segunda parte da Reunião de Planejamento, a Equipe de Desenvolvimento irá planejar como desenvolver as tarefas dos <i>Backlog</i> da <i>Sprint</i> e transformá-los em um incremento do produto Pronto. O <i>Product Owner</i> pode participar desta reunião para esclarecer os requisitos selecionados e caso a Equipe de Desenvolvimento verifique que exista um excesso ou falta de trabalho para a <i>Sprint</i>, os requisitos podem ser renegociados com o <i>Product Owner</i>.</p>
Reunião diária	
Guia do Scrum	Discutir o que foi feito desde a última reunião, o que será feito até a próxima e os impedimentos para desenvolver as tarefas.
Artigos	Não aplicável
Observações	Não aplicável
Diretrizes	Esta reunião tem como foco discutir o que foi feito desde a última reunião, o que será feito até a próxima reunião e os obstáculos que estão tendo para desenvolver suas tarefas.

	É verificado nesta reunião como está o progresso da <i>Sprint</i> em relação a meta da <i>Sprint</i> e avaliar se com o progresso atual será possível completar o trabalho do <i>Backlog</i> da <i>Sprint</i> . O <i>Scrum Master</i> participa desta reunião para garantir que ela aconteça e caso seja detectado algum impedimento que a Equipe de Desenvolvimento esteja tendo para desempenhar suas tarefas, o <i>Scrum Master</i> irá ajuda-los a resolver.
	Revisão da <i>Sprint</i>
Guia do <i>Scrum</i>	Inspecionar o incremento desenvolvido identificando o que está Pronto e o que não está, discutir os problemas que tiveram na <i>Sprint</i> , como foram resolvidos e atualizar o <i>Backlog</i> do Produto.
Artigos	Não aplicável
Observações	Nesta reunião é apresentado tudo o que foi desenvolvido para a Equipe <i>Scrum</i> e as partes interessadas para validar e verificar o que pode ou não ser entregue para o cliente e caso seja necessário, atualizar o <i>Backlog</i> do Produto com os requisitos que não estão "Prontos".
Diretrizes	<p>No fim da <i>Sprint</i>, a Equipe <i>Scrum</i> e as partes interessadas se reúnem para inspecionar o incremento e atualizar o <i>Backlog</i> do Produto caso seja necessário.</p> <p>A Equipe de Desenvolvimento demonstra o trabalho que está Pronto, o <i>Product Owner</i> identifica o que está Pronto e o que não está Pronto e discute o <i>Backlog</i> do Produto da forma como está projetando as prováveis datas de conclusão baseado no progresso até o momento.</p> <p>Nesta reunião a Equipe de Desenvolvimento informa como foi o andamento da <i>Sprint</i>, o que foi bem, os problemas que tiveram e como foram resolvidos referente ao desenvolvimento do produto.</p>
	Retrospectiva da <i>Sprint</i>
Guia do <i>Scrum</i>	Discutir o que está funcionando referente ao processo, planejar formas de aumentar a qualidade do produto e identificar melhorias que serão implementadas na próxima <i>Sprint</i> .
Artigos	Não aplicável
Observações	Caso algum processo não esteja funcionando de forma adequada, é identificado nesta reunião e será discutido as formas para resolver o problema e como poderá ser resolvido na próxima <i>Sprint</i> .

Diretrizes	<p>A Retrospectiva da <i>Sprint</i> é onde a Equipe <i>Scrum</i> pode inspecionar a si próprio e criar um plano com melhorias a serem aplicadas nas próximas <i>Sprints</i>.</p> <p>O propósito da reunião é inspecionar como a última <i>Sprint</i> foi em relação as pessoas, processos e ferramentas, identificar os principais itens em relação ao processo que foram bem, o que pode ser melhorado e criar um plano para implementar melhorias na forma com que a Equipe <i>Scrum</i> faz seu trabalho.</p>
-------------------	--

Artefatos do Scrum	
	Backlog do Produto
Guia do Scrum	Lista ordenada dos requisitos do produto que contém as características, funções, requisitos, melhorias e correções que devem ser feitas no produto em futuras versões.
Artigos	Não existe uma forma prescrita de como deve ser criado o <i>Backlog</i> do Produto, então pode acontecer de as histórias não estarem no nível de detalhamento necessário para poder dar a importância real a cada requisito [READ, BRIGGS, 2012, p.5319].
Observações	Apesar de o <i>Scrum</i> não informar a forma de criar um <i>Backlog</i> do Produto, ter uma recomendação de como fazer ajudará bastante o <i>Product Owner</i> com menos experiência e criará um padrão durante a criação do <i>Backlog</i> do <i>Produto</i> para que todos consigam entender independente de quem faça.
Diretrizes	<p>No Scrum os requisitos são feitos na forma de histórias com descrições curtas da interação do usuário com o sistema.</p> <p>Esta forma de levantar os requisitos não funciona bem quando o projeto se torna maior, pois a complexidade aumenta na quantidade de requisitos, a quantidade e diversidade dos interesses das as partes interessadas aumenta e também a quantidade de alterações que precisam ser gerenciadas.</p> <p>Os métodos ágeis não utilizam um modelo formal de requisitos, o que dificulta uma análise de uma quantidade grande de requisitos e sem isso o conhecimento do projeto fica somente com as pessoas que participaram das reuniões o que pode ficar difícil de repassar para um time maior.</p>

	<p>A falta de experiência do <i>Scrum Master</i> e do <i>Product Owner</i> também pode afetar a qualidade das histórias, tornando-as vagas e ambíguas como por exemplo "Efetuar <i>login</i> no sistema", que não especifica em quais condições o <i>login</i> será feito e quais controles de acesso serão aplicados. Histórias desse tipo podem fazer com que o tempo estimado esteja errado ou ainda a Equipe de Desenvolvimento não faça os requisitos de acordo com que o <i>Product Owner</i> esperava.</p> <p>Para superar esses problemas foi criado o conceito de Hiper-Épico e Hiper-Histórias. O Hiper-Épico é um conjunto de Hiper-Histórias que estão relacionadas e uma Hiper-História é similar a história convencional mas possui uma estrutura em 5 partes: Em <u>algum lugar</u>, uma pessoa baseado em <u>alguma regra</u> faz <u>alguma ação</u>, para atingir <u>algum resultado</u> que produz <u>algum efeito</u>.</p>
	Backlog da Sprint
Guia do Scrum	São os requisitos que foram selecionados do <i>Backlog</i> do Produto para serem desenvolvidos e entregues na <i>Sprint</i> .
Artigos	O <i>Scrum</i> não diz como a Equipe de Desenvolvimento deve se organizar para que os requisitos selecionados sejam feitos de forma otimizada, com isso pode acontecer de alguns membros da Equipe de Desenvolvimento ficarem ociosas enquanto outras estão trabalhando [HONG, YOO, CHA, 2010, p.310].
Observações	Ter uma forma para otimizar o tempo durante o desenvolvimento das tarefas é importante para que não haja uma ociosidade na equipe fazendo com que a equipe desenvolva as tarefas com uma qualidade maior.
Diretrizes	<p>No <i>Scrum</i> tradicional cada membro do time possui foco em apenas uma meta durante a <i>Sprint</i> e esse é o principal motivo pela ociosidade de alguns membros por causa do tempo entre um término e outro das tarefas porque cada membro pode depender da conclusão do trabalho de outro para poder iniciar seu trabalho.</p> <p>Para minimizar a ociosidade entre os membros da equipe é proposto criar um <i>Task Card</i> e um <i>Dashboard</i> sendo que o <i>Task Card</i> é uma lista de tarefas que precisam ser completadas do <i>Backlog</i>.</p> <p>Cada tarefa tem um tempo de conclusão máxima de 24 horas e a pessoa responsável pelo <i>Backlog</i> da <i>Sprint</i> cria o <i>Task Card</i>.</p> <p>Quando o <i>Task Card</i> é criado ele é colocado próximo a data esperado para iniciar no <i>Dashboard</i> e contém duas informações importantes: O <i>ID</i> do requisito no <i>Backlog</i> do Produto e a data esperada para a conclusão da tarefa.</p>

	O <i>ID</i> do requisito permite uma rastreabilidade do progresso entre todos os membros do time e a data esperada para o término permite que os outros membros do time se organizem para diminuir o tempo ocioso porque agora fica claro quando uma tarefa possivelmente será completada, por exemplo, sem o <i>Task Card</i> não conseguimos estimar a data inicial para a codificação sem saber a data de término do <i>design</i> . Com o <i>Task Card</i> isso é possível.
	Incremento
Guia do Scrum	São todos os requisitos que foram completados nessa <i>Sprint</i> e nas <i>Sprints</i> anteriores e que o <i>Time Scrum</i> definiu que estão prontos para o uso independente do <i>Product Owner</i> decidir por liberá-lo.
Artigos	Não aplicável
Observações	Não aplicável
Diretrizes	No fim de cada <i>Sprint</i> é feito uma reunião para a <i>Revisão</i> da <i>Sprint</i> onde é validado o incremento do produto. O incremento são todos os requisitos que foram completados nessa <i>Sprint</i> e nas <i>Sprints</i> anteriores e que a Equipe <i>Scrum</i> definiu que estão prontos para o uso independente do <i>Product Owner</i> decidir por liberá-lo.
	Gráfico <i>Burndown</i>
Guia do Scrum	Várias praticas como <i>Burndown</i> e <i>Burnup</i> tem sido usadas para prever o progresso e apesar de terem se mostrado úteis não substituem a importância do empirismo.
Artigos	O gráfico <i>Burndown</i> é usado para acompanhar o progresso do desenvolvimento do produto [COHN, 2006, p. 219].
Observações	Não aplicável
Diretrizes	O gráfico <i>Burndown</i> é usado para acompanhar o progresso do desenvolvimento do produto e é responsabilidade do <i>Product Owner</i> garantir que o gráfico esteja sempre atualizado.