

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL**  
**FACULDADE DE INFORMÁTICA**  
**CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**Fernanda Dias**  
**Leonardo Fagundes da Silva**

**UNION**

Um Ambiente Colaborativo de Gestão de Projetos *Scrum* Visando Adequação nas  
Áreas de Processos Planejamento de Projeto e Monitoramento e Controle de  
Projetos do CMMI

Porto Alegre  
2009

**Fernanda Dias**  
**Leonardo Fagundes da Silva**

## **UNION**

Trabalho de conclusão de curso de graduação apresentado como requisito para a obtenção do grau de Bacharel em Sistemas de Informação pela Pontifícia Universidade Católica do Rio Grande do Sul.

**Orientador: Prof. Me. Eduardo Meira Peres**

Porto Alegre

2009

## **AGRADECIMENTOS**

Agrademos primeiramente e principalmente aos nossos pais por nos mostrarem o caminho correto a seguir sempre, por nos entenderem por não termos dado toda a atenção que eles mereciam, pelo amor incondicional que sentem por nós e pela confiança depositada em nós. Confiança que não deixaremos de honrar.

Gostaríamos de agradecer a dedicação, trabalho e esforço de nosso orientador, professor Eduardo Meira Peres, que nos apoiou e deu força durante todo o trabalho. Sua ajuda com certeza foi um grande diferencial para a realização deste.

Ao demais professores e todo o corpo docente da PUCRS, que participaram de forma exemplar em nossa formação, principalmente aos professores Flávio Moreira de Oliveira, que, enquanto coordenador do curso de sistemas de informação não mediu esforços para nos ajudar em nossas dificuldades e ao professor Rafael Prikladnicki, pois suas críticas construtivas nos ajudaram a visualizar diferentes horizontes deste trabalho.

A Hewlett-Packard Brasil e principalmente aos nossos gerentes, que nos entenderam nos momentos de dificuldade.

E por fim, a todos aqueles que de alguma forma nos auxiliaram no desenvolvimento deste trabalho. Muito obrigado.

*“Waiting for the one, the day that never comes...”*

*Metallica*

## RESUMO

Este trabalho tem por objetivo avaliar as práticas presentes na metodologia de gerenciamento ágil *Scrum* em relação às áreas de processos Planejamento de Projeto e Monitoramento e Controle de Projeto do CMMI e propor novas práticas ao *Scrum* com o intuito de satisfazer estas áreas de processos. A partir desta avaliação, propõe-se o desenvolvimento de um sistema colaborativo de gestão de projetos que auxilie na implantação e no cumprimento do processo definido com as práticas sugeridas e gere evidências do processo seguido a serem usadas em uma avaliação CMMI.

**Palavras-chave:** *Scrum*, CMMI, métodos ágeis

## ABSTRACT

The purposes of this paper is analyze the practices present in the agile methodology *Scrum* regarding with the CMMI process areas Project Planning and Project Monitoring and Control and propose new practices to be adopted by *Scrum* intending the satisfaction of these process areas. With this analysis, is proposed the development of a collaborative system of project management that assists the implant and the accomplishment of the defined process with the practices suggested and create evidences of the followed process to be used in a CMMI evaluation.

**Key-words:** *Scrum*, CMMI, agile methodology

## LISTA DE ILUSTRAÇÕES

Figura 1: O Processo <i>Scrum</i> .....	18
Figura 2: Exemplo de um <i>Release Burndown chart</i> .....	23
Figura 3: Exemplo de <i>Iteration Burndown chart</i> .....	24
Figura 4: Níveis de Maturidade.....	30
Figura 5: Perfil de Capacidade.....	32
Figura 6: Componentes do CMMI.....	36
Figura 7: Calculando o número de Sprints.....	47
Figura 8: Calculando o custo total estimado.....	48
Figura 9: <i>Burndown chart</i> ilustrando o andamento dos custos do projeto.....	69
Figura 10: Informações utilizadas no <i>Release Plan</i> .....	87
Figura 11: Release Burndown Chart inicial.....	89
Figura 12: Áreas de Funcionalidades.....	91
Figura 13: Arquitetura física do sistema.....	93
Figura 14: Arquitetura Lógica do Sistema.....	94
Figura 15: Sprint 1 Backlog.....	97
Figura 16: <i>Iteration Burndown Chart</i> da <i>Sprint 1</i> .....	99
Figura 17: <i>Sprint 2 Backlog</i> .....	100
Figura 18: <i>Iteration Burndown Chart</i> da <i>Sprint 2</i> .....	101
Figura 19: <i>Sprint 3 Backlog</i> .....	103
Figura 20: <i>Iteration Burndown Chart</i> da <i>Sprint 3</i> .....	104
Figura 21: Sprint 4 Backlog.....	106
Figura 22: Novo Risco Identificado.....	107
Figura 23: <i>Iteration Burndown chart</i> da <i>Sprint 4</i> .....	108
Figura 24: Sprint 5 Backlog.....	110
Figura 25: <i>Iteration Burndown Chart</i> da <i>Sprint 5</i> .....	111
Figura 26: <i>Release Burndown Chart</i> final.....	115
Figura 27: <i>Risk Backlog</i> final.....	116

## LISTA DE TABELAS

Tabela 1 - Princípios do Manifesto Ágil .....	15
Tabela 2: Áreas de processos agrupadas em Categorias e Níveis de Maturidade ...	34
Tabela 3: Exemplo de <i>Product Backlog</i> de uma aplicação que fornece informações de meteorologia em dispositivos móveis .....	44
Tabela 4: <i>Product Backlog</i> .....	87
Tabela 5: <i>Release Plan</i> .....	88
Tabela 6: <i>Risk Backlog inicial</i> .....	91
Tabela 7: Novo risco identificado na Sprint 1 .....	98
Tabela 8: Relação entre <i>User Stories</i> planejadas e <i>User Stories</i> realizadas .....	114
Tabela 9: <i>Product Backlog</i> inicial para trabalhos futuros.....	121

## LISTA DE SIGLAS

CMMI – Capability Maturity Model Integration®

PMI – Project Management Institute

OOPSLA - Object-Oriented Programming Systems, Languages and Applications

PP – Project Planning

PMC – Project Monitoring and Control

GG – Generic Goal

GP – Specific Practice

SG – Specific Goal

SP – Specific Practice

PA – Process Area

SEI – Software Engineering Institute

XP – eXtreme Programming

MOSS – Microsoft Office Sharepoint Server

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>10</b>
1.1. DEFINIÇÃO DO PROBLEMA .....	12
1.2. ESTRUTURA DO TRABALHO .....	13
<b>2. FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>14</b>
2.1. GERENCIAMENTO DE PROJETOS .....	14
2.1.1. Definição de Projeto .....	14
2.1.2. Gerenciamento de projetos.....	14
2.1.3. Métodos Ágeis .....	14
2.1.4. Gerenciamento de Projetos com <i>Scrum</i> .....	16
2.2.1. Histórico.....	28
2.2.2. Visão Geral.....	29
2.2.3. Representações.....	29
2.2.4. Áreas de Processos.....	33
<b>3. PROPOSTA DE SOLUÇÃO .....</b>	<b>37</b>
3.1. DESCRIÇÃO DA SOLUÇÃO .....	37
3.2. OBJETIVOS.....	37
3.2.1. Objetivos Gerais .....	37
3.2.2. Objetivos Específicos.....	38
3.3. PROPOSTA DE INTEGRAÇÃO ENTRE <i>SCRUM</i> E CMMI .....	38
3.3.1. Planejamento de Projeto ( <i>Project Planning - PP</i> ) .....	39
3.3.2. Monitoramento e Controle de Projeto ( <i>Project Monitoring and Control - PMC</i> ) .....	68
3.4. DESCRIÇÃO DO PROJETO .....	81
<b>4. CONCLUSÃO.....</b>	<b>117</b>
4.1. DIFICULDADES ENCONTRADAS .....	118
4.2. TRABALHOS FUTUROS.....	119
<b>REFERÊNCIAS.....</b>	<b>122</b>
<b>ANEXO A – MANUAL DO SISTEMA .....</b>	<b>124</b>

## 1. INTRODUÇÃO

O sucesso do desenvolvimento de software está intimamente ligado com o comprometimento de profissionais habilitados em um processo colaborativo de planejamento contínuo e execução disciplinada (Schuh, 2005). Por ser um processo caro, complexo e repleto de incertezas, há a necessidade de processos bem definidos e controle contínuo para diminuir as chances de fracasso do projeto.

O Standish Group<sup>1</sup> vem, há mais de uma década, realizando estudos sobre os resultados dos projetos de software ao redor do mundo. O resultado destes estudos é publicado através de um relatório chamado *Chaos Report*.

Em 1994, o *Chaos Report* anunciou que, de acordo com as suas pesquisas, 31% dos projetos de TI falhavam e 53% eram desafiados, ou seja, finalizavam, mas com estouro de prazo, orçamento e/ou com redução do escopo acordado (Standish Group, 1994). O Standish Group apenas formalizou o que todos já sentiam: os projetos de desenvolvimento de software estavam em um verdadeiro caos e muito dinheiro estava sendo investido em projetos que tinham chances potenciais em falhar.

Anos antes, o Departamento de Defesa norte-americano já havia percebido que uma quantia excessiva de dinheiro era gasta com softwares que nunca eram entregues ou eram entregues com poucas das funcionalidades esperadas. Assim, foi fundado o Software Engineering Institute (SEI), que reuniu uma série de profissionais acadêmicos e da indústria de software para expor e promover práticas que demonstraram ser bem sucedidas na prevenção das falhas que cercavam os esforços de aquisição de software do Departamento de Defesa. A partir disto, o framework de práticas desenvolvido para diminuir estas falhas se tornou o CMM (SEI, 2008).

O CMM foi expandido em vários outros modelos que se concentravam em partes específicas do negócio das organizações. Porém, percebeu-se a necessidade de uma abordagem de melhoria que avaliasse sistematicamente os problemas que as organizações estavam enfrentando, pois, focando-se apenas em uma área de negócio estes modelos não estavam diminuindo nem removendo as barreiras que existiam nas organizações (Chrissis, et al., 2003).

---

<sup>1</sup> Fonte de pesquisas e análises de desempenho de projetos de TI

Neste contexto nasceu o CMMI, que fornece uma oportunidade de evitar ou eliminar estas barreiras através de modelos integrados que transcendem disciplinas. O CMMI consiste nas melhores práticas para o desenvolvimento e manutenção de produtos.

Porém, estamos vivendo um ritmo acelerado de mudanças e inovações tecnológicas no mercado de software e isto tem causado frustrações crescentes com planos, especificações e documentações pesadas, muitas vezes agregadas por critérios de conformidade dos modelos de maturidade. Foi neste contexto em que uma grande variedade de métodos ágeis surgiu (Boehm, 2006).

Segundo entrevista realizada pela InfoQ com Jim Johnson (InfoQ, 2006), fundador do Standish Group, os resultados do *Chaos Report* de 2004 mostraram que a quantidade de projetos de TI que falhavam baixou de 31% (dados da pesquisa de 1994) para 18%. Segundo o Standish Group, a adoção de processos ágeis foi uma das principais razões para esta diminuição nas falhas.

Alguns princípios dos métodos ágeis existem, de uma forma ou outra, há décadas, porém, o termo “métodos ágeis” conforme conhecemos hoje foi criado em fevereiro de 2001 como resultado de uma reunião de 17 líderes do que era conhecido na época como metodologias “*light*”. Além disto, teve-se como resultado desta reunião, a criação do Manifesto Ágil, que formalizou os métodos ágeis e seus valores (Koch, 2005).

Os métodos ágeis visam o desenvolvimento de software onde as pessoas envolvidas possuem poder de decisão e confiança, aceitando mudanças como uma norma e promovendo *feedback* constante (Schuh, 2005).

Apesar da existência de características distintas entre os métodos ágeis e o modelo CMMI, ambos possuem planos específicos para o desenvolvimento de software e buscam o melhor para que a organização crie software de qualidade (Turner, et al., 2002).

Segundo nota técnica recentemente publicada pelo SEI (SEI, 2008), CMMI e métodos ágeis podem se complementar criando sinergias que beneficiem as organizações que os utilizem. Métodos ágeis fornecem guias para o desenvolvimento de software que não estão contempladas no CMMI. Por outro lado, o CMMI fornece o gerenciamento de processos e práticas de suporte que auxiliam a utilização de métodos ágeis em qualquer organização.

A partir destes problemas no desenvolvimento de software e destas duas abordagens, que hoje são muito utilizadas separadamente, mas pouco utilizadas integradas, este trabalho propõe a integração entre a metodologia ágil *Scrum* e as áreas de processos de gerenciamento de projetos do nível 2 do CMMI (Planejamento de Projeto e Monitoramento e Controle de Projeto). Esta integração poderia ser aplicada tanto em organizações que utilizam *Scrum* e almejam realizar avaliações formais CMMI quanto em organizações que desejam integrar de forma equilibrada a agilidade do *Scrum* com a disciplina do CMMI no que se refere ao gerenciamento de projetos

Além disto, este trabalho visa o desenvolvimento de um sistema colaborativo que auxilie na implantação e no cumprimento do processo definido com as áreas de processo sugeridas e gere evidências do processo seguido a serem usadas em uma avaliação CMMI.

## 1.1. DEFINIÇÃO DO PROBLEMA

O desenvolvimento de software é um processo caro e complexo, onde fatores como escopo, custos, tempo, qualidade e expectativas devem ser gerenciados. Neste cenário, a gerência de projeto torna-se uma tarefa difícil, onde a tomada de decisões adequadas é essencial para que os objetivos do projeto sejam atingidos.

Para aumentar as chances de sucesso no desenvolvimento de software, as organizações estão recorrendo a diferentes estratégias. Dentre estas estratégias, podem-se destacar os modelos de maturidade, como o CMMI, para a definição de processos organizacionais ou então a implantação de métodos ágeis, para ganhar produtividade e flexibilidade.

Porém, métodos de desenvolvimento ágil e as práticas do CMMI são frequentemente vistas como incompatíveis. Se este conceito e suas causas não forem esclarecidos, é provável que haja maior confusão e conflito neste assunto conforme a adoção de ambas as abordagens aumenta (SEI, 2008).

Além disto, cada abordagem possui bons princípios a serem aplicados no desenvolvimento de software que usualmente passam despercebidos, mas são necessários e válidos para aplicação na outra abordagem. Portanto, o conhecimento

de ambas as abordagens é importante para o sucesso do projeto e, em longo prazo, esta discordância não é saudável para a engenharia de software (SEI, 2008).

Desta forma, é importante destacar que a integração de métodos ágeis com o CMMI deve ser feita de maneira cautelosa, visando-se uma melhoria efetiva do processo de software potencializada por esta integração. Deve-se tomar muito cuidado para não reduzir a flexibilidade e adaptabilidade dos métodos ágeis com a adoção de formalismos que o modelo CMMI prescreve. Por outro lado, deve-se compreender que as metodologias ágeis necessitam da definição de processos adicionais de forma a incorporar algumas boas práticas em seu processo essencialmente empírico de desenvolvimento.

## 1.2. ESTRUTURA DO TRABALHO

Este trabalho é organizado da seguinte forma:

- O capítulo 2 aborda os conceitos de gerenciamento de projetos, métodos ágeis, modelos de maturidade, *Scrum* e CMMI. O objetivo do capítulo é explicar cada um destes conceitos, mostrando suas propostas e práticas.
- O capítulo 3 apresenta a proposta de solução para o problema definido na seção 1.1, apresentando a descrição da solução, os objetivos do trabalho e uma integração entre a metodologia *Scrum* e as áreas de processo de Planejamento de Projeto e Monitoramento e Controle de Projeto do modelo de maturidade CMMI. Além disto, este capítulo mostra o escopo do trabalho necessário na criação de um sistema colaborativo que integre *Scrum* com CMMI e auxilie no cumprimento do processo definido, assim como os recursos necessários para o desenvolvimento do sistema. Este capítulo também mostra as fases de planejamento, preparação, desenvolvimento e entrega do produto.
- O capítulo 4 apresenta as conclusões deste trabalho.

## **2. FUNDAMENTAÇÃO TEÓRICA**

### **2.1. GERENCIAMENTO DE PROJETOS**

#### **2.1.1. Definição de Projeto**

Um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo (PMBOK, 2004), ou seja, deve ter um início e fim definidos e seu resultado deve possuir elementos que o diferencie dos demais.

Em uma organização, os projetos servem para auxiliar a mesma a atingir seus objetivos em todos os níveis, sejam eles operacional, tático ou estratégico.

#### **2.1.2. Gerenciamento de projetos**

O gerenciamento de projetos é a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos (PMBOK, 2004).

O gerenciamento de um projeto deve identificar os objetivos a serem alcançados, controlar o andamento do mesmo para assegurar que o resultado gerado atenda o escopo, o tempo, o custo e a qualidade prevista e identificar e mitigar os riscos que podem levar o projeto ao fracasso.

A gerência de um projeto, na maioria das vezes, fica a cargo de uma pessoa, denominada gerente de projeto. Esta pessoa está ligada às tarefas do projeto, mas dificilmente realiza as atividades pertinentes a estas tarefas. No lugar disto, seu trabalho é garantir que os envolvidos no projeto possam progredir para que este tenha êxito.

#### **2.1.3. Métodos Ágeis**

Há alguns anos, um grupo de profissionais da área de desenvolvimento de software se reuniu, com o intuito de melhorar o desempenho dos projetos em que eles trabalhavam. Deste encontro, foi criado o Manifesto para o Desenvolvimento

Ágil de Software (2001), que descreve princípios que visam aumentar a eficácia do desenvolvimento de software. Os princípios são:

<b>Valorizamos</b>	<b>Mais que</b>
Indivíduos e interações	Processos e ferramentas
Software em funcionamento	Documentação abrangente
Colaboração com o cliente	Negociação de contratos
Responder a mudanças	Seguir um plano à risca

**Tabela 1 - Princípios do Manifesto Ágil**

Projetos que utilizam métodos ágeis dividem o desenvolvimento do produto em curtos intervalos de tempo, chamados de iterações, que costumam variar de uma a quatro semanas. Cada iteração é vista como um mini projeto, que possui todas as tarefas necessárias para implementar uma nova funcionalidade e garantir sua qualidade.

Cada iteração possui as fases de planejamento, análise, projeto, codificação e testes. No final de cada iteração, é criado um entregável, que pode ser visto como uma versão limitada do produto final. Todas as funcionalidades acordadas para a iteração em questão devem estar funcionando corretamente no final do mesmo.

Os métodos ágeis se diferem dos tradicionais por serem mais dinâmicos e menos resistentes às mudanças de escopo. O grande segredo dos métodos ágeis, é que eles partem do pressuposto que não é possível ter uma visão completa do escopo do trabalho no início do projeto e, portanto, é aceitável e natural que existam mudanças no decorrer de seu ciclo de vida.

O gerenciamento de projetos que utilizam métodos ágeis possui vantagens ao longo das etapas de planejamento, execução e controle do projeto. Enquanto as metodologias convencionais utilizam como base para estas etapas todo o escopo do projeto, as metodologias ágeis se focam no escopo de uma iteração, que é consideravelmente menor que o do projeto, conseguindo assim planejar, executar e controlar de forma mais precisa.

Algumas das características dos métodos ágeis são:

- Entrega de código funcional por etapas: Ao final de cada iteração, deve ser entregue um código funcional correspondente a tudo o que foi desenvolvido naquela iteração.
- Forte colaboração na equipe: Através da reorganização das tarefas quando um elemento do time está com problemas em cumprir o que lhe foi passado, recebendo ajuda dos demais e da disseminação do conhecimento dos integrantes nas diferentes áreas do projeto, fazendo com que os membros passem a ter conhecimento total do escopo.
- Priorização de tarefas: As tarefas mais importantes, complexas e com maiores riscos são feitas nas primeiras iterações, deixando as últimas para melhorias e tarefas mais simples. Assim, os problemas que geralmente são vistos apenas no final dos projetos que utilizam um processo tradicional são encontrados cedo em um projeto ágil, podendo ser resolvidos antecipadamente, causando menor impacto no programa.

Como exemplo de métodos ágeis, podemos citar *eXtreme Programming (XP)*, *Scrum* e *Feature-Driven Development (FDD)* (Leffingwell, 2008).

#### **2.1.4. Gerenciamento de Projetos com *Scrum***

A dificuldade no gerenciamento e controle de um sistema está ligada a complexidade do mesmo. Projetos complexos, que envolvem diversos níveis de conhecimento em diferentes áreas, necessitam que a tomada de decisão seja descentralizada, ou seja, as pessoas envolvidas nas tarefas devem ter poder de influenciar e/ou tomar decisões referentes ao projeto.

Esta é a principal característica e diferencial da metodologia *Scrum*, que visa fornecer aos times que participam do desenvolvimento do software a autoridade necessária para tomarem decisões para que o produto final possua maior qualidade e possa ser entregue no tempo estimado, mesmo que ocorram mudanças durante o desenvolvimento do mesmo.

O *Scrum* foi feito para ser utilizado em projetos com um nível alto de complexidade, onde é impossível levantar todo o escopo e prever todos os problemas e particularidades do projeto em suas primeiras fases.

Ao invés de tentar prever no início de um projeto todos os problemas possíveis que ocorrerão no desenvolvimento, o *Scrum* oferece um conjunto de práticas que auxiliam na visibilidade de problemas que estão acontecendo ou estão para acontecer, ajudando o time a identificar os riscos mais críticos para que estes sejam mitigados o mais rápido possível.

O *Scrum* não pretende assegurar que o projeto seguirá os mesmos passos esperados e planejados inicialmente, mas pretende auxiliar no controle do processo para que o produto final seja o mais próximo possível da expectativa do cliente.

O *Scrum* foi idealizado com o intuito de ser utilizado em projetos de desenvolvimento de softwares, que geralmente possuem um alto grau de complexidade, mas pode ser usado no gerenciamento de projetos em geral, desde linhas de montagens à manutenção de sistemas.

#### 2.1.4.1. Histórico

Em 1986, o *Scrum* foi desenvolvido por Hirotaka Takeuchi e Ikujiro Nonaka para auxiliar o gerenciamento no desenvolvimento de produtos. Seus criadores notaram que companhias líderes continham 6 características utilizadas no processo de desenvolvimento de produtos. Estas características eram: Instabilidade interna, times de projeto auto-organizáveis, fases de desenvolvimento extensíveis, equipes multi-especialistas, controle rígido e transferência de conhecimento dentro da organização.

Times com apenas uma ou algumas destas características não ganhavam velocidade ou flexibilidade. Por outro lado, times que apresentavam a totalidade delas poderiam produzir uma dinâmica poderosa que fazia a diferença (Takeuchi, et al., 1986). Eles associaram este tipo de time à formação do *Scrum* no Rugby.

No início dos anos 90, Ken Schwaber e Jeff Sutherland fizeram as primeiras implantações do *Scrum* em projetos de software nas empresas Advanced Development Methods e Easel Corporation.

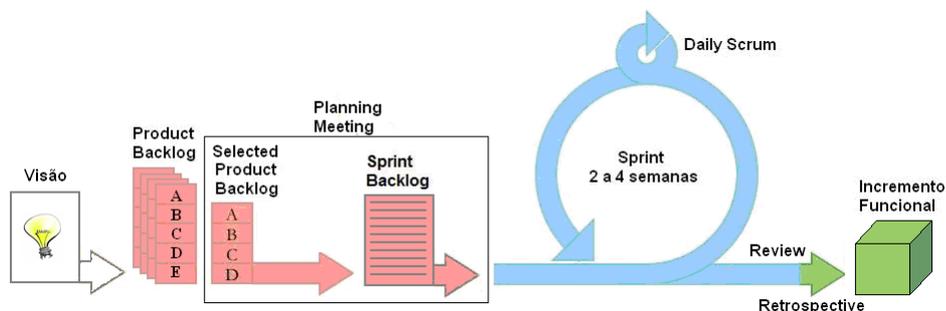
Em 1996, Schwaber e Sutherland formalizaram a definição de *Scrum* na OOPSLA, uma conferência anual sobre programação, sistemas, linguagens e aplicações (Leffingwell, 2008).

Com a proposta de tornar os projetos mais flexíveis e ágeis, a metodologia se encaixou com as necessidades dos projetos de software, se tornando, pouco a pouco, uma das metodologias ágeis mais utilizadas da atualidade.

#### 2.1.4.2. Processo

O *Scrum* é um processo iterativo e incremental. Projetos que utilizam esta metodologia possuirão uma série de iterações, conhecidas como *Sprints*, onde cada iteração inicia com uma lista de especificações a serem desenvolvidas durante aquela iteração e termina com o incremento do produto sendo desenvolvido.

A figura 1 mostra o funcionamento do processo *Scrum*:



**Figura 1: O Processo Scrum**

#### 2.1.4.3. Visão

A visão de um produto representa a sua necessidade. O que está na visão do produto deve ser satisfeito no final do projeto.

Para que esta visão seja definida, o *Product Owner*<sup>2</sup> colhe informações junto a clientes, usuários finais, time, gerentes, executivos, etc.

---

<sup>2</sup> Um dos papéis do *Scrum* abordado detalhadamente na seção 2.1.5.1.

Esta visão pode ser feita, por exemplo, através de uma sessão de *brainstorming*<sup>3</sup> ou derivada de um documento de requisitos do marketing.

#### 2.1.4.4. *Product Backlog*

O *Product Backlog* consiste de todo o trabalho que pode ser previsto para um produto. Ele consiste de funcionalidades, infra-estrutura, arquitetura e tecnologia (Schwaber, 2004).

O responsável por manter o *Product Backlog* é o *Product Owner*. Ele deverá manter o *Product Backlog* atualizado e priorizado de acordo com a necessidade de cada item no produto.

O *Product Backlog* não é um documento que é criado no início do projeto e fica estático até o fim do mesmo. Ele é modificado constantemente, mudando de acordo com as mudanças no escopo do produto, com as mudanças nas prioridades e com outras mudanças que podem aparecer durante o desenrolar do projeto. Ele também é atualizado a cada iteração, para que os itens de uma *Sprint* que foram afetados (positivamente e negativamente) pelas mudanças ocorridas na *Sprint* anterior possam ser atualizados, tanto em escopo dos requisitos quanto em critérios de aceitação e priorização.

#### 2.1.4.5. *User Stories*

Uma *User Story* descreve uma funcionalidade que terá valor para um usuário ou cliente de um sistema ou software (Cohn, 2004). Elas são compostas de três aspectos:

- Uma descrição escrita da *User Story* para ser usada nas sessões de planejamento e mesmo como uma lembrança do desejo do cliente;
- Conversas sobre a *User Story* para destacar os detalhes da mesma;
- Testes de aceitação que podem ser utilizados para determinar quando a *User Story* está completa;

---

<sup>3</sup> Atividade em grupo cujo objetivo é gerar um grande número de idéias para a solução de um problema.

*User Stories* foram originadas como parte de *eXtreme Programming*, e, apesar do *Scrum* não prever sua utilização, elas se encaixam naturalmente em processos iterativos e incrementais (Cohn, 2004).

Com esta abordagem, os itens do *Product Backlog* podem ser expressos como *User Stories*. Isto facilita a priorização do *Product Backlog* já que cada item possui algum valor de negócio para o *Product Owner* (Cohn, 2004).

O objetivo de utilizar *User Stories* é elaborar requisitos pequenos, enxutos e com critérios de aceitação no lugar de elaborar documentos extensos e formais sobre todos os requisitos do projeto.

Quando uma *User Story* é muito grande, ela é chamada de *Epic*. *Epics* podem ser divididas em duas ou mais *User Stories* de tamanho menor (Cohn, 2004).

Quando existem *User Stories* que ainda não foram bem definidas no momento da criação do *Product Backlog*, uma *Epic* é criada para que esta *User Story* seja dividida no futuro.

Quando o time não consegue estimar uma *User Story* por não entender a tecnologia envolvida ou a especificação da mesma, um *Spike* pode ser criado. Um *Spike* é um breve experimento para que se aprenda sobre a área da aplicação. Durante o *Spike*, membros do time aprendem somente o necessário para poder estimar a *User Story*. Desta maneira, uma *User Story* não estimável se torna duas *User Stories*: Um breve *Spike* para se obter informação e a *User Story* para que o trabalho real seja feito (Cohn, 2004).

#### 2.1.4.6. *Release Plan*

O *Release Plan* é um artefato criado após a definição do *Product Backlog*, que apresenta as *User Stories* separadas por *Sprint*, ou seja, a partir da previsão do número de *Sprints* necessárias para conclusão do projeto e da estimativa de tamanho e/ou esforço de uma *User Story*, é possível dividir estas *User Stories* nas *Sprints*, sempre respeitando suas prioridades.

#### 2.1.4.7. *Planning Meeting*

Clientes, usuários, gerentes, o *Product Owner*, o *Scrum Master*<sup>4</sup> e o time determinam os objetivos e funcionalidades da próxima *Sprint* na *Planning Meeting*. O time então planeja as tarefas individuais que precisam ser feitas para criar o incremento do produto (Schwaber, 2004).

No início de cada *Sprint*, o time revisa tudo o que ainda é necessário ser feito para que o produto esteja pronto. Tudo aquilo que possa ser feito dentro desta nova iteração e que possa transformar o resultado da mesma em uma possível funcionalidade entregável é selecionado para ser desenvolvido dentro desta iteração. Estas tarefas são colocadas em um artefato chamado *Sprint Backlog*, que é abordado no tópico 2.1.4.9.

#### 2.1.4.8. *Selected Product Backlog*

O *Selected Product Backlog* é a lista de *User Stories* do *Product Backlog* que foram selecionadas para serem desenvolvidas na *Sprint*. Elas são selecionadas de acordo com a prioridade do *Product Backlog*.

#### 2.1.4.9. *Sprint Backlog*

O *Sprint Backlog* contém as *User Stories* que foram selecionadas no *Product Backlog* para fazerem parte da próxima iteração. As *User Stories* são quebradas em tarefas e estas são estimadas em horas, geralmente tendo entre 4 e 16 horas (Schwaber, 2004). Tarefas menores que 4 horas devem ser agrupadas em uma tarefa maior, enquanto que tarefas maiores que 16 horas devem ser quebradas em tarefas menores.

É uma prática do Scrum incentivar os integrantes do time a escolher as *User Stories* que estes pretendem trabalhar. Isto aumenta a satisfação durante o desenvolvimento da mesma, pois é bem provável que o integrante irá trabalhar no que realmente deseja.

---

<sup>4</sup> Um dos papéis do *Scrum* abordado detalhadamente na seção 2.1.5.1.

O *Sprint Backlog* é uma ferramenta que mostra em tempo real o trabalho que o time planeja finalizar durante o *Sprint* (Schwaber, 2004). O mesmo deve ser atualizado diariamente para conter as tarefas e estimativas atualizadas.

#### 2.1.4.10. *Sprint*

Cada *Sprint* possui em média de 2 a 4 semanas.

Dentro desta iteração, o time analisa os requisitos, projeta como isto será feito no software e desenvolve o que foi projetado, efetuando mudanças quando alguma complexidade ou dificuldade for encontrada no meio da *Sprint*.

O time verifica o que precisa ser feito e seleciona a melhor maneira de fazê-lo. Este processo criativo é o coração da produtividade do *Scrum* (Schwaber, 2004).

Durante a *Sprint*, caso o time sinta necessidade de consultar especialistas nas áreas das funcionalidades sendo desenvolvidas, o *Product Owner* fica a cargo de encontrar estes especialistas e agendar reuniões entre os especialistas e o time.

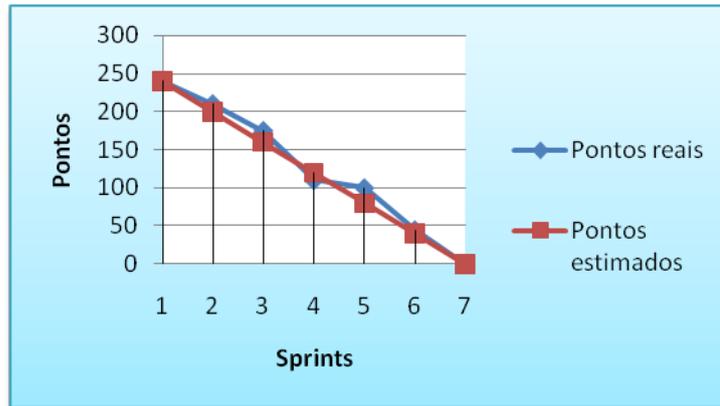
#### 2.1.4.11. *Release Burndown chart*

O *Release Burndown chart* mostra a quantidade de trabalho a ser gasto ao decorrer de um projeto. Feito em forma de gráfico, ele mostra a divisão de trabalho em cada iteração para que o produto final esteja pronto.

A figura 2 mostra um exemplo de *Release Burndown chart*, onde o eixo vertical mostra o número de *Story Points*<sup>5</sup> restantes no projeto e o eixo horizontal mostra o número de iterações.

---

<sup>5</sup> Técnica de estimativa de esforço, onde pontos são atribuídos às *User Stories* do *Product Backlog* e somados para determinar o tamanho do projeto e a divisão das *User Stories* por *Sprints*.



**Figura 2: Exemplo de um *Release Burndown chart***

O *Release Burndown chart* é um poderoso indicador visual que mostra a velocidade em que um time está se aproximando de seus objetivos (Cohn, 2005).

Com estes dados, o *Product Owner*, o *Scrum Master* e o time conseguem rever o andamento do time no final de cada *Sprint* e, caso necessário, fazer um replanejamento para as próximas iterações, como por exemplo: contratar mais recursos ou cortar algumas *User Stories*.

#### 2.1.4.12. Daily Scrum

A cada dia de trabalho, é feita uma reunião, chamada *Daily Scrum*, para verificar como está o progresso e sincronizar o trabalho do time.

As *Daily Scrums* são reuniões de 15 minutos aonde os membros do time respondem 3 questões:

- O que você fez desde a última *Daily Scrum*?
- O que você planeja fazer até a próxima *Daily Scrum*?
- Quais são os impedimentos para o seu trabalho?

Estes impedimentos são registrados em uma lista, conhecida como *Impediments Backlog*, onde o *Scrum Master* é responsável pela priorização e gerência do itens da lista, além de assegurar que os impedimentos sejam solucionados e removidos do *Impediments Backlog* (Sutherland, 2006).

#### 2.1.4.13. Iteration *Burndown chart*

O *Iteration Burndown chart* é uma representação gráfica que mostra o trabalho a ser feito em relação ao tempo restante da iteração.

O gráfico mostra a redução diária da quantidade de horas necessárias para completar a iteração e possui o esforço estimado e o realizado na *Sprint*.

A figura 3 mostra um exemplo de *Iteration Burndown chart*.

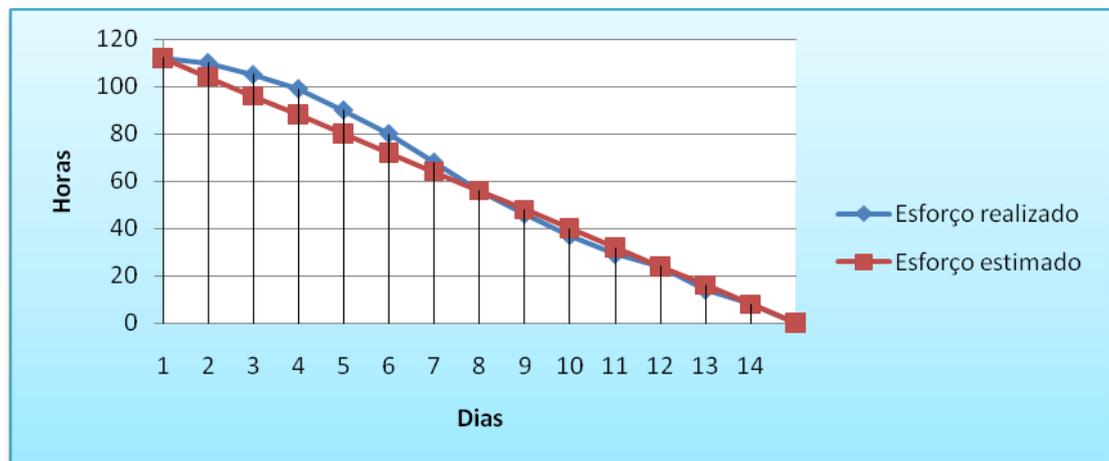


Figura 3: Exemplo de *Iteration Burndown chart*

O processo de criação de um *Iteration Burndown chart* é muito simples. No início da iteração, o time estima todo o trabalho que será feito dentro da iteração. Esta estimativa, em horas, é colocada no eixo vertical do gráfico. Este total de trabalho então é dividido em dias e separado pelo gráfico (Schuh, 2005).

Após a criação da estimativa de esforço, ao final de cada dia, o esforço é atualizado para refletir a realidade do projeto.

#### 2.1.4.14. *Review Meeting*

No final da iteração, é realizada uma reunião, chamada *Review Meeting*. Esta reunião tem duração estimada de 4 horas, onde o time faz uma demonstração aos *Stakeholders* dos incrementos feitos durante a *Sprint* (Schwaber, 2004). Os *Stakeholders* discutem sobre as mudanças no produto, e, caso alguma adaptação seja necessária, o *Product Backlog* é atualizado adicionando estas mudanças.

#### 2.1.4.15. Retrospective

A reunião de retrospectiva de uma *Sprint* é chamada de *Retrospective* e possui o intuito de verificar as lições aprendidas pelo time que devem ser avaliadas. Nesta reunião, o time discute os problemas ocorridos na *Sprint* atual e busca melhorias para as próximas *Sprints*.

#### 2.1.4.16. Incremento Funcional

O *Scrum* requer que um incremento do produto seja produzido a cada iteração. Este incremento deve possuir todas as funcionalidades necessárias para que aquela parte do produto pronta possa ser utilizada. Isto significa que o incremento precisa estar bem estruturado, testado e sem *bugs* para que um executável do mesmo seja criado. Esta é a definição de uma iteração finalizada.

Incrementos são importantes porque muitas vezes, ao pensar em uma funcionalidade, ela faça sentido e funcione, mas apenas quando examinamos esta funcionalidade desenvolvida, testada e entregue, possamos descobrir se ela realmente trás valor agregado ao negócio. O *Scrum* força os *Stakeholders* a verificar as partes do produto mais críticas enquanto o produto ainda não está completamente desenvolvido, auxiliando em alguma mudança necessária no produto antes que esta mudança se torne mais cara ou até inviável.

#### 2.1.4.17. Papéis

Existem 3 papéis dentro do *Scrum* e todas as responsabilidades do projeto serão divididas entre estes 3 papéis. Quem não estiver em nenhum destes papéis, não faz parte do processo de *Scrum*, podendo participar de reuniões diárias e de planejamento, mas sem poder opinar.

Dentro do *Scrum*, há uma clara diferença entre as pessoas que preenchem um destes 3 papéis e outros interessados no projeto, mas não envolvidos com o mesmo. Aqueles que fazem parte do projeto são responsáveis pelo mesmo e tem a autoridade para fazer o que for necessário para o seu sucesso, enquanto aqueles

que não fazem parte do projeto não são responsáveis pelo mesmo e não podem interferir de maneira desnecessária.

A seguir, são apresentados os 3 papéis que fazem parte do *Scrum*:

#### 2.1.4.17.1. *Product Owner*

O *Product Owner* é o responsável por representar os interesses de todos os envolvidos no projeto e também de apresentar o seu resultado (Schwaber, 2004).

Ele deve assegurar, sob uma perspectiva de negócio, para que o time trabalhe da maneira correta, priorizando as tarefas com maiores riscos e com maior valor agregado para que as mesmas sejam feitas nas primeiras *Sprints* do projeto. Ele também é responsável por criar os requisitos iniciais do projeto, os objetivos e os planos de entrega.

#### 2.1.4.17.2. *Scrum Master*

O *Scrum Master* é responsável por garantir que o processo *Scrum* está sendo seguido, por ensinar e explicar o processo para os demais envolvidos, por implantar o *Scrum* para que ele se torne padrão no projeto e na cultura organizacional, para assegurar que o *Scrum* está trazendo retorno positivo ao projeto e para remover os impedimentos que os integrantes do time possam estar tendo.

Na maioria das vezes, o *Scrum Master* é o gerente ou o líder técnico do projeto.

#### 2.1.4.17.3. Time

O time é responsável por analisar, projetar, desenvolver e testar funcionalidades. Eles são auto-gerenciáveis e multi-especialistas. Serem auto-gerenciáveis não faz com que o projeto deixe de necessitar de um gerente, mas auxilia o gerente quando o próprio time estima o tempo das tarefas e sabe a responsabilidade que possui em entregá-las no prazo e com qualidade.

Os membros do time são responsáveis pelo sucesso de cada *Sprint* e pelo sucesso do projeto como um todo.

#### 2.1.4.18. Fases

Segundo Larman (2004), o ciclo de vida do *Scrum* possui quatro fases:

- Planejamento: nesta fase a visão do projeto deve ser estabelecida e ficar clara para todos os envolvidos. As versões iniciais do *Product Backlog*, do plano de entregas e a arquitetura de negócio também são criadas.
- Preparação: nesta fase, o escopo e as dimensões do projeto ficam mais claros e, portanto, são adicionados novos itens ao *Product Backlog*. Os times são formados e mecanismos de comunicação e coordenação entre eles são definidos. Esta fase também é conhecida como *Iteration Zero* ou *Sprint Zero*, que é tratada como uma *Sprint*, porém possui um comportamento diferente das demais, pois seu objetivo não é entregar algo funcional no final, mas sim construir a base necessária no time para que a entrega de funcionalidades seja possível (Schuh, 2005).
- Desenvolvimento: nesta fase, ocorre o desenvolvimento iterativo e incremental do produto através das *Sprints*.
- Entrega: nesta fase, a entrega final do produto é realizada.

Outros autores apresentam uma proposição diferente para as fases do *Scrum*. Por exemplo, Ken Schwaber (Schwaber) propõe a divisão do *Scrum* em três fases: *Pré-Game*, que contemplaria um conteúdo similar ao das fases de planejamento e preparação, *Game* e *Post-Game*, que contemplariam, respectivamente, um conteúdo similar as fases de desenvolvimento e entrega.

Neste trabalho será utilizada a abordagem proposta por Larman (Larman, 2004).

## 2.2. CMMI

O CMMI consiste em uma série de boas práticas dirigidas para o desenvolvimento e manutenção de produtos e serviços, abrangendo todo seu ciclo de vida, desde a concepção até a entrega e manutenção (Chrissis, et al., 2003).

### 2.2.1. Histórico

Com a popularização dos computadores modernos, a demanda por desenvolvimento de software cresceu subitamente conforme as organizações sentiam necessidade de automatizar seus processos.

O resultado desta grande demanda, combinado à inexperiência e imaturidade das empresas desenvolvedoras de software, causou a falha de muitos projetos, com estouro dos orçamentos e prazos previstos e principalmente a geração de produtos com baixa qualidade e o abandono de projetos, trazendo muitos prejuízos tanto para as companhias contratantes quanto para as contratadas.

Entidades do governo norte-americano, que possuíam contratos com empresas fornecedoras de software, perceberam estas dificuldades e, em uma tentativa de identificar e corrigir os problemas nos processos de desenvolvimento de software, o Departamento de Defesa norte-americano criou um estudo no Software Engineering Institute (SEI), situado na universidade Carnegie Mellon. O resultado deste estudo foi o *Capability Maturity Model (CMM)*: uma série de diretivas a serem seguidas para a melhoria do processo de desenvolvimento de software, gerando maturidade nas empresas que implantavam o modelo. Após a implantação do modelo, os processos das empresas passavam por avaliações CMM, sinônimo de qualidade nos produtos desenvolvidos.

Assim, sentiu-se necessidade da criação de modelos específicos para determinadas áreas, como o SW-CMM para os processos de desenvolvimento de software, SE-CMM para engenharia de sistemas, SA-CMM para os processos de aquisição de software, entre outros.

Porém, estes modelos possuíam estruturas diferentes um do outro e as formas de medir a maturidade das organizações variavam. Neste cenário, as

empresas ficavam confusas no momento da escolha e implementação dos modelos e, além disso, era muito difícil utilizar mais de um modelo simultaneamente.

Para resolver este problema de incompatibilidade, foi desenvolvido o CMMI (*Capability Maturity Model Integration*<sup>®</sup>) que é a integração dos modelos do CMM existentes em um framework comum de diretivas para a melhoria dos processos das organizações.

Atualmente, o SEI não oferece mais suporte para o CMM e as organizações que possuem CMM são incentivadas a realizar uma atualização em seus processos para se enquadrarem no CMMI.

### **2.2.2. Visão Geral**

Com o intuito de definir e/ou aprimorar os processos das empresas, as diretivas do CMMI se focam em dois pontos: o conteúdo e a melhoria dos processos (Ahern, et al., 2004).

Primeiramente, avaliam-se os processos da empresa e as informações que são necessárias para as atividades da organização. O objetivo é utilizar o modelo como um guia para a definição dos processos através da criação e/ou modificação destes.

Após a definição dos processos, é necessário elaborar um plano para a implantação dos mesmos na organização. Para tanto, o CMMI também possui dicas e diretivas que incluem a padronização dos processos e treinamentos que devem ser ministrados para que as pessoas envolvidas saibam seus papéis e objetivos e como devem proceder para alcançá-los.

Além disso, o modelo trabalha com a melhoria contínua dos processos, que visa a coleta de métricas para ajustá-los conforme a maturidade da organização aumenta.

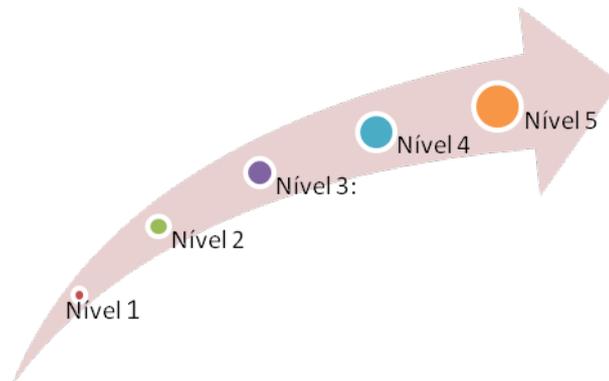
### **2.2.3. Representações**

O CMMI disponibiliza duas representações que podem ser escolhidas pra implantação: representação por estágios e a contínua.

### 2.2.3.1. Representação por estágios

A representação por estágios fornece à organização um guia estruturado para a melhoria dos processos da organização.

Cada estágio do modelo desenvolve uma base sólida para o desenvolvimento e implantação da próxima fase. Estes estágios são chamados de níveis de maturidade e contêm uma série de áreas de processos que estabelecem um caminho a ser seguido pela organização para atingir a maturidade máxima em seus processos. Este caminho pode ser visualizado como na figura 4:



**Figura 4: Níveis de Maturidade**

Os níveis de maturidade são:

- **Nível 1 – Inicial:** neste nível as organizações se encontram em um ambiente instável e em desordem. A qualidade dos produtos desenvolvidos não é previsível e os projetos geralmente excedem o orçamento e ultrapassam o cronograma acordado. Os processos não são institucionalizados e cada indivíduo pode possuir um entendimento diferente dos mesmos.
- **Nível 2 – Gerenciado:** neste nível de maturidade, os projetos da organização gerenciam seus requisitos e os processos são planejados, mensurados e controlados (Chrissis, et al., 2003). Os times de projeto possuem mais disciplina neste nível, onde as práticas a serem aplicadas são seguidas mesmo nos momentos críticos. O ciclo de vida do projeto é controlado, segue os padrões e práticas estipulados e tende a seguir como foi planejado.

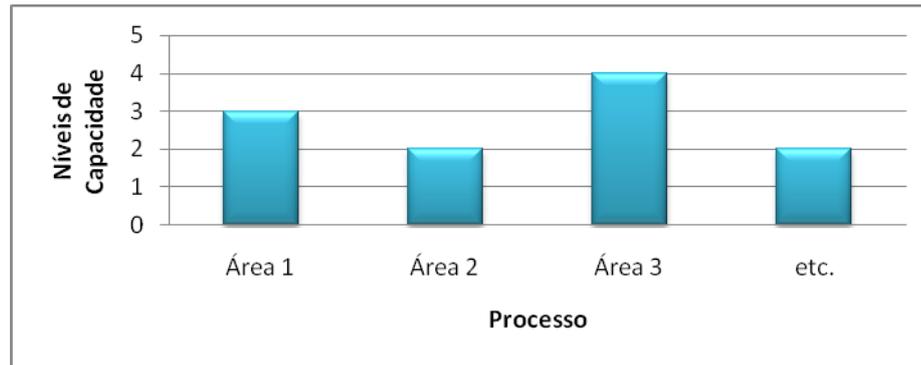
- Nível 3 – Definido: no nível 2, o foco era obter maturidade nos processos a nível de projeto. Já neste nível, o objetivo é obter maturidade nos processos a nível de organização. São criados padrões, práticas, métodos e ferramentas a partir do estudo e entendimento dos processos existentes. Um conjunto de processos padrões da organização é estabelecido, este conjunto é uma série de definições de processos que irão guiar as atividades da organização e devem ser incorporados a todos os projetos.
- Nível 4 – Gerenciado Quantitativamente: neste nível, a organização estabelece objetivos que podem ser medidos quantitativamente e os incorpora no gerenciamento dos processos. Estes objetivos são levantados a partir das necessidades de clientes e do escopo do negócio da organização.
- Nível 5 – Otimizado: este nível se foca na melhoria contínua dos processos, revisando-os e modificando-os para refletir a realidade da empresa, que muda conforme novos objetivos de negócio são inseridos à organização e conforme a empresa aumenta a maturidade em seus processos.

#### 2.2.3.2. Representação Contínua

A representação contínua é uma abordagem flexível de implementação do modelo, pois esta permite que a organização escolha uma área de processos (ou um grupo de áreas de processos) a serem melhorados (Chrissis, et al., 2003).

Nesta representação, a avaliação da empresa se encaixa em níveis de capacidade.

Estes níveis se focam em avaliar as habilidades que uma organização possui em executar, controlar e melhorar os processos de uma área de processos específica (Kulpa, et al., 2003). Os resultados podem ser visualizados em um perfil de capacidade, ilustrado na figura 5:



**Figura 5: Perfil de Capacidade**

Os níveis de capacidade são:

- **Nível 0 – Incompleto:** neste nível, um ou mais objetivos da área de processo não são satisfeitos pelos processos existentes. Este nível é equivalente ao nível 1 da representação por estágios.
- **Nível 1 – Executado:** neste nível, todos os objetivos, genéricos e específicos, definidos no nível 1 do modelo são implementados. Porém, a definição do processo pode não estar ainda de acordo com a realidade e muitas vezes, objetivos específicos como qualidade, custos e prazos podem não ser satisfeitos. Este é um nível inicial e atingi-lo significa que a empresa está caminhando em direção à melhoria dos seus processos, porém sem poder mensurar se houve melhoria significativa nesta institucionalização.
- **Nível 2 – Gerenciado:** um processo gerenciado é planejado, monitorado e controlado para que atinja um determinado propósito (Kulpa, et al., 2003). Isto significa que todos os objetivos deste nível descritos pelo modelo devem ser atingidos e é possível também atingir outros objetivos definidos pela organização. Neste nível, é necessário coletar métricas e aplicá-las constantemente no gerenciamento a fim de desenvolver metas cada vez mais realistas.
- **Nível 3 – Definido:** um processo definido é um processo gerenciado que é reconhecido como processo padrão da organização, isto é, a definição do processo foi documentada, revisada e aprovada. Além disso, o processo possui sentido dentro da empresa, que, ao institucionalizar uma série de processos forma uma identidade, onde o

desenvolvimento de seus produtos é feito de uma maneira única e documentada e os resultados são medidos.

- Nível 4 – Gerenciado Quantitativamente: um processo quantitativamente gerenciável é um processo definido que é controlado e avaliado através de estatísticas e outras técnicas de análise quantitativa (Kulpa, et al., 2003). Objetivos que podem ser medidos quantitativamente como qualidade e desempenho do processo são estabelecidos e são utilizados como metas no gerenciamento dos processos.
- Nível 5 – Em Otimização: neste nível, são avaliados os resultados e o desempenho dos processos definidos e quantitativamente gerenciados da organização. Esta avaliação foca-se na procura de problemas, que possam ter uma causa raiz em comum, e busca melhorias que acompanhem os objetivos e realidade da empresa. As mudanças detectadas nestas avaliações devem ser documentadas, além de serem fornecidos treinamentos às pessoas envolvidas.

#### **2.2.4. Áreas de Processos**

Uma área de processo, ou *Process Area (PA)*, é um grupo de práticas relacionadas a uma área que, quando implementada coletivamente, satisfaz uma série de objetivos considerados importantes para a melhoria daquela área (Chrissis, et al., 2003). As áreas de processos apresentadas pelo modelo são iguais para ambas as representações.

O CMMI possui 22 PAs que podem ser agrupadas em 4 categorias: Gerência de Processos, Gerência de Projeto, Engenharia e Suporte. A tabela 2 mostra as PAs agrupadas e o nível de maturidade que elas correspondem:

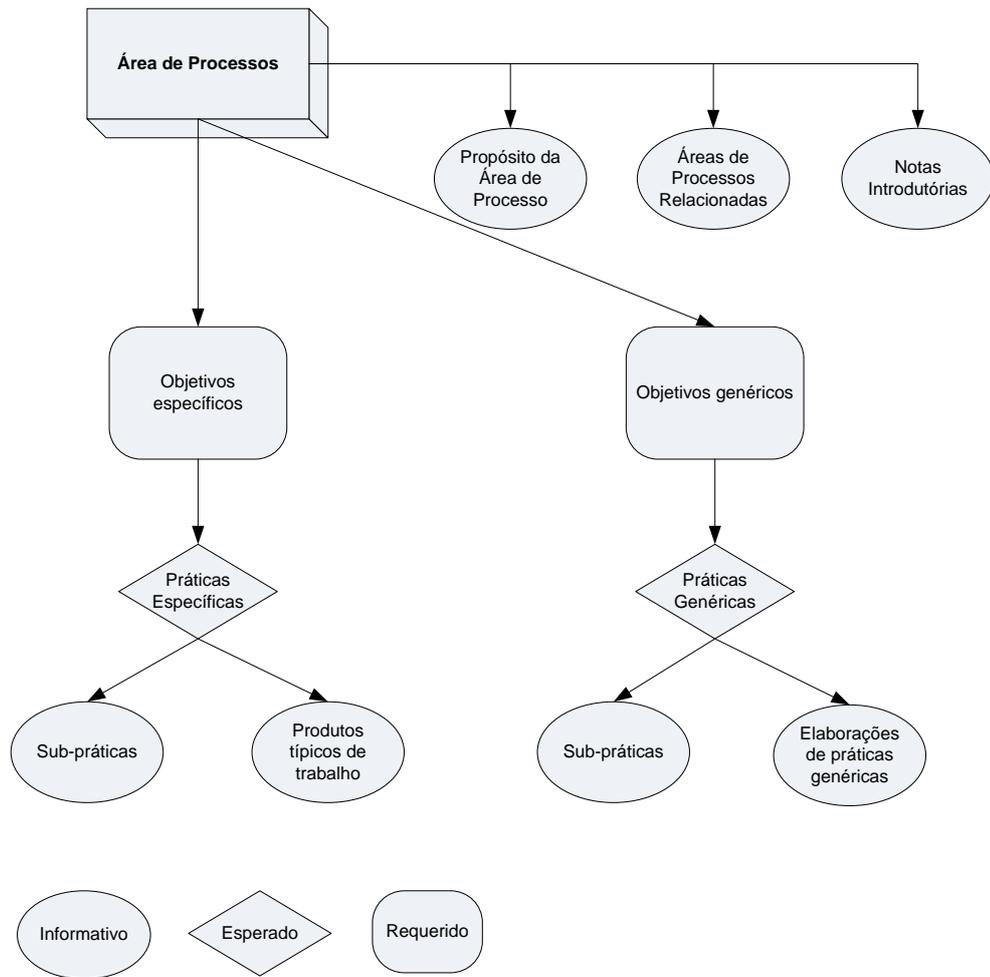
<b>Categoria</b>	<b>Nível de Maturidade</b>	<b>Área de Processo</b>
Gerência de Projetos	2	Monitoramento e Controle de Projeto
		Planejamento de Projeto
		Gerenciamento de Acordo com o Fornecedor
	3	Gerenciamento Integrado de Projeto
		Gerenciamento de Riscos
	4	Gerenciamento Quantitativo de Projeto
Engenharia	2	Gerenciamento de Requisitos
	3	Integração do Produto
		Desenvolvimento de Requisitos
		Solução Técnica
		Validação
		Verificação
Suporte	2	Gerência de Configuração
		Medição e Análise
		Garantia de Qualidade de Processo e Produto
	3	Análise de Decisão e Resolução
	5	Análise de Causa e Resolução
Gerência de Processos	3	Definição de Processo Organizacional
		Foco no Processo Organizacional
		Treinamento Organizacional
	4	Desempenho do Processo Organizacional
	5	Inovação Organizacional e <i>Deployment</i>

**Tabela 2: Áreas de processos agrupadas em Categorias e Níveis de Maturidade**

Os principais componentes de uma área de processos são:

- Componentes requeridos: descrevem o que deve ser alcançado pela organização para satisfazer a área de processo e deve ser visivelmente adotado em seus processos. Os componentes requeridos no CMMI são os objetivos genéricos e os objetivos específicos.
  - Objetivos Genéricos (GG): descrevem as características que precisam estar presentes para institucionalizar os processos que implementam uma área de processo. São chamados de genéricos, pois eles aparecem em diversas áreas de processos;
  - Objetivos Específicos (SG): descrevem as características específicas de uma área de processos que precisam estar presentes para satisfazer os requisitos desta área;
- Componentes esperados: descrevem o que uma organização pode implementar para atingir um componente requerido. Antes de um objetivo ser considerado satisfeito, tanto as práticas descritas no modelo quanto práticas alternativas aceitáveis pelo modelo estão presentes nos processos das organizações. Componentes esperados incluem práticas específicas e práticas genéricas.
  - Práticas genéricas (GP): descrevem uma atividade que é considerada importante para atingir o objetivo genérico associado.
  - Práticas específicas (SP): descrevem uma atividade que é considerada importante para atingir o objetivo específico associado.
- Componentes informativos: fornecem detalhes que ajudam as organizações a pensar na abordagem a ser tomada para a implementação dos componentes requeridos e os esperados. Componentes informativos incluem sub-práticas, produtos de trabalho típicos, elaborações de práticas genéricas, notas introdutórias, etc.

A figura 6 ilustra os componentes de uma área de processo:



**Figura 6: Componentes do CMMI**

Portanto, apesar de o modelo descrever as práticas genéricas e específicas para atingir os objetivos, não é necessário que a organização utilize especificamente essas práticas, ou seja, ela pode propor práticas diferentes que não são descritas no modelo, desde que estas novas práticas produzam evidências que atinjam os objetivos da área de processo.

### 3. PROPOSTA DE SOLUÇÃO

#### 3.1. DESCRIÇÃO DA SOLUÇÃO

A solução para os problemas apresentados na seção 1.1 esta dividida em:

- **Objetivos:** a seção 3.2 mostra os objetivos gerais e específicos desta proposta de solução.
- **Proposta de integração entre o *Scrum* e o CMMI:** a seção 3.3 apresenta uma proposta de integração entre a metodologia ágil *Scrum* e as práticas genéricas das áreas de processos Planejamento de Projeto e Monitoramento e Controle de Projeto do CMMI.
- **Descrição do projeto:** a seção 3.4 apresenta a o desenvolvimento de um sistema que proporcione um ambiente colaborativo para projetos de software, onde os processos definidos estejam inerentes ao sistema e que este auxilie no planejamento, monitoramento e controle de um projeto que utilize a metodologia *Scrum*. Além disto, o sistema gera automaticamente evidências do cumprimento dos processos definidos, diminuindo a sobrecarga do *Scrum* com formalismos solicitados pelo CMMI.

#### 3.2. OBJETIVOS

##### 3.2.1. Objetivos Gerais

Os objetivos gerais deste trabalho são:

- Propor uma forma de integração entre a metodologia ágil *Scrum* e as áreas de processos de Planejamento de Projeto e Monitoramento e Controle de Projeto do nível 2 do CMMI.
- Desenvolver a prototipação de um sistema colaborativo que auxilie na implantação e no cumprimento do processo definido com as áreas de processo sugeridas e gere evidências do processo seguido a serem usadas em uma avaliação CMMI.

### 3.2.2. Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Desenvolver um sistema que incentive a colaboração entre o time de um projeto através de um ambiente de fácil utilização e que centralize as informações, facilitando o controle do projeto, aumentando a produtividade da equipe e auxiliando o cumprimento do processo definido;
- Aprofundar o conhecimento do grupo em gerenciamento de projetos que utilizem *Scrum* e/ou CMMI;
- Aprofundar o conhecimento do grupo no desenvolvimento de sistemas colaborativos;
- Aplicar a metodologia proposta ao gerenciar o desenvolvimento da ferramenta.

### 3.3. PROPOSTA DE INTEGRAÇÃO ENTRE SCRUM E CMMI

A proposta de integração está focada em analisar o *Scrum* e suas práticas e artefatos em relação às práticas específicas das áreas de processos escolhidas, com o intuito de satisfazê-las.

Nos casos em que o *Scrum* atende o propósito das práticas específicas analisadas, será explicado como os métodos e técnicas já existentes no *Scrum* poderiam ser considerados como práticas alternativas em uma avaliação CMMI. Já, nos casos em que o *Scrum* não atende ou atende parcialmente o propósito das práticas específicas analisadas, será sugerida a adoção de novos conceitos, técnicas e artefatos ao *Scrum* que satisfaçam o propósito da prática específica.

Além disto, as práticas do CMMI são meramente indicativas, pois, para obter uma avaliação CMMI positiva em uma área ou nível de maturidade, a organização deve apenas provar, através de evidências de práticas, que os objetivos daquela área ou nível estão sendo satisfeitos, isto é, a organização é totalmente livre para propor novas práticas e evidências que a ajudem a satisfazer as exigências do CMMI (Anderson, 2005).

Outra abordagem que poderia ser adotada é a proposição de práticas alternativas às práticas específicas prescritas no CMMI. Entretanto a proposição deste trabalho é verificar até que ponto as práticas específicas poderiam ser efetivamente adotadas em um processo *Scrum*, que neste caso deve então ser estendido.

As funcionalidades sugeridas buscarão auxiliar na adoção e cumprimento da prática específica descrita, gerando evidências para os formalismos do CMMI. Com isto, espera-se que a agilidade dos times *Scrum* não seja diminuída com o atendimento às práticas do CMMI.

Neste trabalho, serão analisadas as áreas de processos Planejamento de Projeto (PP) e Monitoramento e Controle de Projeto (PMC) em busca de uma integração com o *Scrum*. Estas áreas foram escolhidas por serem áreas de gerenciamento de projetos do nível 2. A proposta de integração será estruturada da seguinte forma:

- Descrição das práticas e objetivos específicos das PAs;
- Integração da metodologia *Scrum* com as práticas específicas;
- Lista de funcionalidades que o sistema proposto poderia possuir;

Os objetivos e práticas genéricas das áreas de processos não foram contemplados nesta integração, pois se procurou analisar objetivos e práticas que favoreceriam a criação de funcionalidades tangíveis para o sistema colaborativo proposto.

### **3.3.1. Planejamento de Projeto (*Project Planning* - PP)**

#### 3.3.1.1. SG 1 Estabelecer Estimativas

##### ***Descrição***

O propósito deste objetivo específico é estabelecer e manter os parâmetros de planejamento de projeto.

Os parâmetros de planejamento de projeto incluem todas as informações necessárias para executar o planejamento, organização, gestão de recursos, direção, coordenação, divulgação e orçamento (Chrissis, et al., 2003).

Fatores que são tipicamente considerados para realizar as estimativas destes parâmetros incluem:

- Requisitos do projeto;
- Escopo do projeto;
- Produtos de trabalho e tarefas e seus atributos (ex.: tamanho e complexidade);
- Abordagem técnica;
- Ciclo de vida do projeto;
- Cronograma.

#### 3.3.1.1.1. SP 1.1 Estimar o Escopo do Projeto

##### ***Descrição***

Esta prática visa estabelecer uma estrutura analítica de trabalho (WBS) em alto nível para estimar o escopo do projeto (Chrissis, et al., 2003).

A WBS é a decomposição hierárquica, orientada à entrega do trabalho, a ser executada para que os objetivos do projeto sejam atingidos (PMBOK, 2004).

O desenvolvimento da WBS divide o escopo do projeto em uma série de componentes gerenciáveis. Seu objetivo é fornecer um esquema para identificação e organização dos pacotes de trabalho a serem gerenciados. Esta estrutura em alto nível do projeto pode servir para a realização de uma estimativa de trabalho inicial. (CMMI Product Team, 2006).

Os produtos de trabalho típicos desta prática são: descrição das tarefas, descrição dos pacotes de trabalho e a WBS.

## ***Integração com Scrum***

O *Scrum* prevê a criação de um documento, chamado *Product Backlog*, que representa todo o trabalho previsto para o desenvolvimento do produto incluindo todas as funcionalidades, melhorias, correções de defeitos, entre outros itens (Schwaber, et al., 2001).

No início do projeto, o *Product Backlog* deve ser elaborado a partir do documento de visão do projeto, de reuniões de brainstorming ou da derivação das solicitações do marketing (Schwaber, et al., 2001).

Para a criação dos itens do *Product Backlog*, pode-se utilizar *User Stories*, conforme descrito na seção 4.3.4.1.

Usualmente, *Product Backlog* é utilizado apenas para o escopo do produto. Porém, o mesmo também pode ser utilizado para o escopo do projeto, mapeando treinamentos necessários em forma de Spikes, por exemplo.

Portanto, apesar de não prever a criação de uma WBS, o *Scrum* prevê a criação de uma lista que proporciona uma visão da dimensão do projeto, permitindo a visualização dos limites do escopo do trabalho, os pacotes de trabalho a serem gerenciados e uma estimativa de esforço inicial para o projeto. Assim, podemos considerar que a prática do *Scrum* de criação do *Product Backlog* satisfaz esta prática específica.

## ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Proporcionar um *template* para a criação de *User Stories* e uma forma simples e prática de manutenção;
- Proporcionar dicas de como escrever *User Stories*;
- Facilitar a priorização das *User Stories* dentro do *Product Backlog*;
- Lembrar o *Product Owner* que este deve criar o *Product Backlog*;
- Realizar o versionamento do *Product Backlog* para que as informações nele contidas desde o início do projeto estejam disponíveis durante todo seu ciclo de vida;

- Disponibilizar o *Product Backlog* de maneira fácil a todos os envolvidos no projeto para que todos possam acessá-lo, visualizá-lo e saber quais os itens do *Product Backlog* possuem maior prioridade e precisam ser trabalhados primeiro;
- Incentivar a atualização do *Product Backlog* pelo *Product Owner* através de lembretes, para que o documento fique conciso com a realidade do projeto;
- Permitir a inserção de *Epics* e diferenciá-los das demais *User Stories*;
- Permitir a quebra de *Epics* em *User Stories* menores de forma simples;
- Permitir a associação de um *Spike* com uma *User Story* para auxiliar o rastreio deste *Spike*.

#### 3.3.1.1.2. SP 1.2 Estabelecer Estimativas de Atributos de Produtos de Trabalho e Tarefas

##### **Descrição**

Esta prática visa estabelecer e manter estimativas de atributos de trabalho e tarefas (Chrissis, et al., 2003).

Tamanho é uma entrada primária que muitos modelos utilizam para estimar esforço, custo e cronograma (CMMI Product Team, 2006).

Exemplos de produtos de trabalho cujas estimativas de tamanho são realizadas incluem o que será e o que não será entregue, documentos e arquivos, hardware, firmware e software operacionais e de suporte.

Exemplos de medidas de tamanho incluem número de funções, pontos de função, linhas de código, número de classes e objetos, número de requisitos, volume de dados, restrições físicas, etc.

As estimativas devem ser consistentes com os requisitos do projeto para determinar o esforço, custo e cronograma do projeto. Um nível relativo de dificuldade ou complexidade deve ser atribuído para cada atributo de tamanho (CMMI Product Team, 2006).

Produtos de trabalho típicos desta prática específica são: abordagens técnicas, tamanho e complexidade de produtos de trabalho e tarefas, modelos de estimativa, atributos estimados.

### ***Integração com Scrum***

No *Scrum*, depois que o *Product Backlog* é criado, o *Product Owner* estima o tamanho do projeto e, para isso, ele conversa com os desenvolvedores, os envolvidos com controle de qualidade e outras pessoas que entendem o produto e a tecnologia utilizada (Schwaber, et al., 2001).

Porém, o *Scrum* não indica, em suas práticas, técnicas ou modelos a serem utilizados para realizar as estimativas do projeto.

Uma forma de atender a necessidade de estimativa de tamanho seria o uso de *Story Points* (Cohn, 2005).

*Story Points* é uma unidade de medida que expressa o tamanho total de uma *User Story*, funcionalidade ou alguma outra porção de trabalho (Cohn, 2005).

Esta técnica consiste em avaliar as *User Stories* do projeto e atribuir pontos para estas levando em consideração sua complexidade, seu esforço e riscos envolvidos, entre outros atributos (Cohn, 2005). No *Scrum*, esta avaliação poderia ser realizada sobre os itens do *Product Backlog*, conforme proposto a seguir:

Primeiramente, os itens do *Product Backlog* devem ser priorizados por importância ao negócio que este representa para o produto final. Após, deve ser identificado o item de menor valor em relação aos atributos escolhidos para avaliação (complexidade, tamanho, esforço, riscos, etc.). Quando este item for encontrado, deve-se atribuir 1 ponto para ele. A seguir, os outros elementos do *Product Backlog* devem ser avaliados em comparação com este, onde um item que exija 2x mais esforço para ser concluído receberá 2 pontos e assim sucessivamente.

A tabela 3 ilustra um exemplo de pontuação.

ID	Descrição	Pontos
1	O usuário pode visualizar a temperatura em pelo menos uma unidade de temperatura	5
2	O usuário pode visualizar o estado do tempo (nublado, chuvoso, etc.)	5
3	O usuário pode selecionar a cidade a qual deseja visualizar a temperatura	3
4	O usuário pode visualizar a previsão do tempo para no máximo os próximos 3 dias	8
5	O usuário pode visualizar a temperatura em outra unidade de temperatura	5
6	O usuário pode visualizar ícones que representam o estado do tempo	1

**Tabela 3: Exemplo de *Product Backlog* de uma aplicação que fornece informações de meteorologia em dispositivos móveis**

Estes pontos servirão como base para a realização das estimativas das atividades em unidades de tempo posteriormente.

Desta forma, ao adicionarmos uma técnica de estimativa como *Story Points* ao *Scrum* é possível satisfazer esta prática específica, já que será utilizado um método de estimativa apropriado e que os produtos de trabalho, neste caso os itens do *Backlog*, serão propriamente estimados. Porém, para isso, é necessário realizar a estimativa de tamanho de todos os itens do *Product Backlog* logo na fase de planejamento, o que não é exigido pelas práticas do *Scrum*.

### ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Permitir a pontuação das *User Stories* no *Product Backlog*;
- Adicionar uma tarefa ao *Product Owner* para que este se lembre de marcar reuniões para discutir os pontos dos itens do *Product Backlog* e realizar a pontuação propriamente dita;

- Disponibilizar uma lista de atributos comuns para servir de base para a pontuação dos itens do *Product Backlog*. Esta lista pode permitir entradas de outros atributos que podem ser relevantes para o projeto. Com isso, o *Product Owner* e a equipe de projeto poderão realizar esta tarefa tendo como base os mesmos atributos para todos os itens.

#### 3.3.1.1.3. SP 1.3 Definir o ciclo de vida do projeto

##### **Descrição**

Esta prática visa definir as fases do ciclo de vida do projeto para determinar o escopo do esforço de planejamento (Chrissis, et al., 2003).

A definição do ciclo de vida do projeto e de suas fases fornece momentos planejados para avaliação e tomada de decisão. Estes períodos permitem correções e mudanças no curso do projeto, além da determinação do escopo e custo futuros.

Estas fases são definidas de acordo com o escopo dos requisitos, a natureza e as estimativas dos recursos do projeto.

Projetos maiores podem conter múltiplas fases, tais como concepção, desenvolvimento, produção, operação, descarte e subfases, como análise de requisitos, projeto, fabricação, integração, etc. (CMMI Product Team, 2006).

Para a determinação do escopo do esforço e adequação do tempo no planejamento e replanejamento de um projeto, o entendimento do ciclo de vida do mesmo é fundamental (Chrissis, et al., 2003).

O produto de trabalho típico desta prática é: fases do ciclo de vida do projeto.

##### **Integração com Scrum**

O *Scrum* possui um ciclo de vida iterativo e incremental bem definido composto por 4 fases: Planejamento, Preparação, Desenvolvimento e Entrega (Larman, 2004) descritos na seção 4.3.5. Portanto, o *Scrum* já satisfaz esta prática específica e seu ciclo de vida favorece o planejamento. Além disto, por considerar que o projeto está em constante mudança, o ciclo de vida do *Scrum* favorece o replanejamento do mesmo, através de iterações curtas que conseguem prover a

monitoração constante do projeto e identificar pontos que necessitam de novo planejamento.

### ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivarão o cumprimento desta prática específica:

- Possuir as 4 fases definidas no *Scrum* (Planejamento, Preparação, Desenvolvimento e Entrega), com os artefatos a serem criados, revisados e/ou atualizados e as reuniões de cada fase a serem agendadas;
- Disponibilizar *checklists* com o que deve ser feito em cada fase para seguir o ciclo de vida do *Scrum*;
- Permitir a visualização das fases planejadas, suas tarefas e artefatos de maneira simples e clara.

#### 3.3.1.1.4. SP 1.4 Determinar Estimativas de Esforço e Custo

### ***Descrição***

Esta prática visa estimar o esforço e custo do projeto para os produtos de trabalho e tarefas com base em fundamentos lógicos de estimativas (Chrissis, et al., 2003).

Estimativas de esforço e custo são, em geral, baseadas no resultado de análises integrando modelos de estimativas e dados históricos com as características do projeto.

Existem situações onde os esforços são sem precedentes, ou seja, um componente ou o produto inteiro é inédito e, portanto, os dados históricos disponíveis podem não se aplicar ao projeto. Estas ocasiões possuem maiores riscos, requerem mais pesquisas para desenvolver bases de estimativas razoáveis e demandam maiores reservas estratégicas. Por isso, as características particulares do projeto devem ser documentadas para garantir um entendimento comum de

todas as suposições feitas nos estágios iniciais do planejamento (CMMI Product Team, 2006).

Os produtos típicos de trabalho desta prática são: fundamento lógico das estimativas, estimativas de esforço do projeto e estimativas de custo do projeto.

### ***Integração com Scrum***

Com o uso da técnica de *Story Points* nos itens do *Product Backlog*, ao final do processo de estimativa têm-se o tamanho, em pontos, necessários para a realização do projeto.

Para determinar estimativas de esforço e custo, é necessária a introdução do conceito de velocidade oriundo da técnica de estimativa *Story Points*.

Velocidade é a medida da taxa de progresso do time. Esta taxa é calculada através da soma dos pontos das *User Stories* que o time completou durante uma *Sprint* (Cohn, 2005). Por exemplo, se o time completou 3 *User Stories* que valem 5 pontos cada uma, a velocidade deste time é 15 pontos por *Sprint*.

Assim, se somarmos todos os pontos do *Product Backlog* obtém-se uma estimativa de tamanho total do projeto e, com a velocidade do time, é possível dividir o tamanho pela velocidade e encontrar uma estimativa do número de iterações (Cohn, 2005). A figura 7 ilustra este cálculo.

Pontos Totais	Velocidade	Número de Sprints
90	/ 15	= 6

**Figura 7: Calculando o número de Sprints**

Com uma estimativa do número de *Sprints* necessárias para finalizar o projeto e com a duração em semanas de cada *Sprint*, tem-se uma estimativa de duração do projeto. No exemplo acima, se os *Sprints* fossem de 2 semanas, o projeto iria ser finalizado no final de 12 semanas.

Para determinar as estimativas de custo do projeto, pode-se seguir uma abordagem similar. Primeiramente, deve-se calcular o custo médio de uma iteração do projeto, tendo como parâmetros de entrada os recursos humanos e físicos que

serão utilizados, seu custo por período de tempo (semana, dia, hora) e o período de duração de uma iteração (Schuh, 2005). Com o custo estimado por iterações, pode-se determinar o custo total estimado do projeto, ilustrado na figura 8.

Pontos Totais	Velocidade	Número de Sprints	Custo por <i>Sprint</i>	Custo Total
90	/ 15	= 6	X 13.500,00	= 81.000,00

**Figura 8: Calculando o custo total estimado**

O cálculo do custo de uma iteração pode ser muito complexo, já que alguns recursos podem ser compartilhados e cada projeto pode ter particularidades que devem ser inseridas neste cálculo. Porém, fornecer um modelo para o cálculo deste custo não é a intenção deste trabalho, visto que um cálculo similar deverá existir em projetos que utilizam outras metodologias.

Portanto, com a técnica de *Story Points* aplicada sobre os itens do *Product Backlog* e o conceito de velocidade do time, acredita-se que é possível estimar esforço e custo do projeto, que é o objetivo desta prática específica.

### ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Gerar automaticamente a base histórica, com a velocidade do time em pontos por *Sprint* a partir dos dados das *Sprints* já realizadas, a fim de proporcionar dados consistentes com a realidade para uso nas estimativas da próxima *Sprint*. Por exemplo, se estavam estimados 20 pontos em uma determinada *Sprint* e, ao seu final, o time conseguiu entregar 25 pontos, a ferramenta poderia ser capaz de atualizar a velocidade do time automaticamente, auxiliando no planejamento da próxima *Sprint*.
- Gerar alertas quando o *Sprint Backlog* estiver alocando mais ou menos pontos que a velocidade da equipe. Por exemplo, se a velocidade da equipe for 25 pontos e para a próxima *Sprint* estejam sendo estimados

35 pontos, a ferramenta poderia ser capaz de alertar o *Scrum Master* sobre esta situação.

### 3.3.1.2. SG 2 Desenvolver o Plano de Projeto

Este objetivo específico visa estabelecer e manter o plano de projeto como a base do gerenciamento do projeto (Chrissis, et al., 2003).

O plano de projeto é utilizado para gerenciar e controlar a execução do projeto. Ele é baseado nos requisitos do projeto e das estimativas estabelecidas.

#### 3.3.1.2.1. SP 2.1 Estabelecer o Orçamento e o Cronograma

##### ***Descrição***

Esta prática visa estabelecer e manter o orçamento e cronograma do projeto (Chrissis, et al., 2003).

O orçamento e o cronograma do projeto são baseados em estimativas e asseguram que a alocação de recursos, complexidade de tarefas e dependências entre elas serão tratadas apropriadamente (CMMI Product Team, 2006).

Cronogramas com marcos definidos e recursos limitados trazem maiores benefícios no tratamento de riscos do projeto, já que neste planejamento é possível identificar situações problemáticas e, assim, proporcionar certa flexibilidade para negociações e melhor entendimento destas situações.

Os produtos típicos de trabalho desta prática são: cronogramas de projeto, dependência entre cronogramas e orçamento de projeto.

##### ***Integração com Scrum***

Por ser um processo iterativo, o *Scrum* prevê dois níveis de planejamento: um macro, que é realizado na fase de planejamento do projeto como um todo e um no nível das iterações.

No nível macro, são estabelecidos o *Product Backlog*, a visão, o *Release Plan*, as estimativas gerais de esforço e custo e outros artefatos de planejamento. No nível de iterações, são estabelecidos o *Sprint Backlog* e os outros artefatos de planejamento são revisados e atualizados de acordo com a realidade do projeto no momento.

Assim, no *Scrum*, o cronograma em nível macro está especificado no *Release Plan*, que separa as *User Stories* em *Sprints*, sempre respeitando as prioridades de cada uma. Este plano fornece uma visualização das *Sprints* planejadas e é aprovado, juntamente com o *Product Backlog*, ao final da fase de planejamento.

Já o *Sprint Backlog* é um cronograma no nível de iterações, aprovado durante as *Planning Meetings*, onde as *User Stories* selecionadas para uma *Sprint* são quebradas em tarefas que tornam possível o cumprimento dos objetivos daquela *Sprint*. Estas tarefas possuem estimativas em horas e recursos alocados para sua execução, porém elas não são delegadas ao recurso e sim, este escolhe quais tarefas ele gostaria de estar alocado. No início do projeto tem-se somente o *Sprint Backlog* da primeira *Sprint* e os outros *Sprint Backlogs* são criados nas *Planning Meetings* de cada *Sprint*.

Tanto nas *Planning Meetings* quanto nas *Review Meetings* de cada *Sprint*, o *Product Backlog* pode ser alterado e provocar mudanças no *Release Plan*. Nestes casos, o *Release Plan* deve ser atualizado com as mudanças para obter-se a visualização real dos releases do projeto, ou seja, este plano deve ser revisado sempre que o *Product Backlog* for alterado e aprovado durante as reuniões.

O *Scrum* não formaliza onde o orçamento do projeto deve estar disponível. Portanto, sugerimos agregar esta informação nos dois níveis de cronograma que o *Scrum* possui, ou seja, o orçamento macro, de todo o projeto no *Release Plan* e o orçamento por iteração no *Sprint Backlog*.

Assim, com estes artefatos e a inclusão do orçamento nos mesmos, entendemos que esta prática específica seja atendida.

### ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Permitir a criação de *Sprints*, com duração e datas de início e fim;
- Permitir a alocação de *User Stories* em uma determinada *Sprint*;
- Gerar uma sugestão de *Release Plan*, a partir das *User Stories* já devidamente pontuadas e priorizadas e da velocidade do time, permitir alterações neste plano de forma simples e facilitar a visualização das *User Stories* que não estão vinculadas a uma *Sprint*;
- Permitir a criação de tarefas para uma determinada *User Story*, com prioridade, estimativa, recursos alocados e dependência entre esta e outras tarefas;
- Permitir a inclusão dos orçamentos macro do projeto e do orçamento por iteração.

#### 3.3.1.2.2. SP 2.2 Identificar os Riscos do Projeto

##### **Descrição**

Esta prática tem como objetivo identificar e analisar os riscos do projeto (Chrissis, et al., 2003).

Segundo o PMBOK (PMBOK, 2004), um risco de projeto é um evento ou condição incerta que, se ocorrer, terá um efeito positivo ou negativo em pelo menos um objetivo do projeto.

Os riscos de um projeto devem ser identificados ou descobertos e analisados para apoiar o planejamento do projeto. Todos os planos que afetam o projeto devem ter seus riscos analisados para assegurar que o tratamento necessário seja dado a cada risco, diminuindo assim, a probabilidade de um risco negativo prejudicar o projeto.

Os produtos de trabalho típicos desta prática são: Riscos identificados, prioridade dos riscos, impacto dos riscos e probabilidade de sua ocorrência.

## ***Integração com Scrum***

Processos cujo ciclo de vida é iterativo e incremental, como o *Scrum*, são orientados a riscos, isto é, os elementos mais complexos, difíceis e com maior risco associado são realizados nas primeiras iterações (Larman, 2004).

Além disto, nas reuniões de revisão, que ocorrem no final de cada *Sprint*, pode-se identificar se os objetivos do projeto estão sendo atingidos e com o *feedback* do cliente é possível perceber se o produto está ou não satisfazendo suas necessidades. Nos casos em que o produto não esteja se adequando às expectativas do cliente, ou este deseje efetuar mudanças no produto, estas mudanças poderão ser realizadas durante o processo de desenvolvimento do mesmo, evitando que o cliente se decepcione com o produto final.

No *Scrum*, a análise dos riscos do projeto bem como o controle apropriado para os mesmos é realizada na fase de planejamento (Schwaber). Entretanto, os criadores do *Scrum* reconhecem que este possui diversas lacunas, como o gerenciamento de riscos, e indicam que estas devem ser preenchidas com as melhores práticas (AgileCollab, 2008). Ou seja, o *Scrum* não formaliza uma abordagem para identificação e análise dos riscos em seu processo. Cada organização adota a abordagem que mais se encaixa com a sua realidade.

Para preencher a lacuna de gerenciamento de riscos do *Scrum*, sugerimos a criação um artefato chamado *Risk Backlog*, criado na fase de planejamento com os riscos iniciais e atualizado e acompanhado durante as *Planning Meetings* e *Review Meetings* de cada *Sprint*.

Este artefato seria de responsabilidade de todo o time, incluindo o *Product Owner*, e deve ter ligação com os itens do *Product Backlog* para proporcionar melhor visualização do que é afetado, além de auxiliar na decisão do tratamento a ser usado para cada risco.

O *Risk Backlog* proposto seria um documento, baseando-se em alguns conceitos de gerenciamento de riscos do PMBOK (PMBOK, 2004), contendo as seguintes informações:

- Prioridade: a prioridade associada a cada risco;
- Categoria: tipificação do risco de acordo com a realidade do projeto, quando necessário;

- Rastreabilidade: *User Stories* afetadas pelo risco, quando necessário;
- Descrição: a descrição do risco identificado;
- Análise Qualitativa:
  - Avaliação de Probabilidade: a probabilidade de cada risco específico ocorrer;
  - Avaliação de Impacto: o efeito potencial sobre um objetivo do projeto, como tempo, custo, escopo ou qualidade, inclusive os efeitos negativos das ameaças e os efeitos positivos das oportunidades.
- Estratégia de resposta aos riscos:
  - Prevenir/Explorar: realizar mudanças no planejamento de projeto para eliminar uma ameaça apresentada por um risco adverso ou eliminar a incerteza de ocorrência associada a um risco positivo;
  - Transferir/Compartilhar: realizar a passagem do impacto negativo de uma ameaça para terceiros, juntamente com a propriedade de resposta ou atribuir a propriedade a terceiros que possam capturar melhor a oportunidade em benefício do projeto;
  - Mitigar/Melhorar: reduzir a probabilidade e/ou impacto de um evento de risco adverso ou aumentar de um risco positivo;
  - Aceitar: não modificar o planejamento de projeto para tratar um risco.
- Ação: a ação a ser tomada de acordo com a estratégia escolhida;
- Responsável: pessoa responsável pelo risco.

Nas reuniões previstas no *Scrum* como as *Planning Meetings* e as *Review Meetings*, propõe-se a discussão dos itens do *Risk Backlog*, o levantamento de novos riscos e a aprovação do artefato.

Portanto, com a formalização do gerenciamento de riscos, adoção do *Risk Backlog* e integração do mesmo no processo, esta prática específica pode ser satisfeita.

## ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Possuir um Risk *Backlog*, para conter os riscos, que podem ser criados com prioridade, categoria, descrição, impacto, probabilidade de ocorrência, estratégia, ação corretiva e usuário responsável pelo mesmo;
- Criar um risco a partir de uma *User Story* ou tarefa e associá-lo automaticamente;

### 3.3.1.2.3. SP 2.3 Planejar o Gerenciamento de Dados

#### ***Descrição***

Os dados de um projeto são todos os artefatos gerados durante o ciclo de vida do mesmo. Estes dados podem estar na forma de relatórios, códigos-fonte, especificações, e-mails, etc.

Os requisitos de dados para o projeto deveriam ser estabelecidos para os itens de dados a serem criados e seus conteúdos e formas, baseados em um conjunto padrão de requisitos de dados. Requisitos de formato e uniformidade de conteúdo para itens de dados facilitam o entendimento e contribuem para o gerenciamento consistente dos recursos de dados (CMMI Product Team, 2006).

Os artefatos do projeto devem possuir uma razão para serem criados. Segundo o CMMI for Development (CMMI Product Team, 2006), a criação de documentos sem um entendimento claro de como o mesmo será utilizado é muito comum. Dados são custosos e deveriam ser criados apenas quando necessário.

Alguns dos produtos de trabalho típicos desta prática são: Plano de gerenciamento de dados, lista de dados gerenciados, requisitos de privacidade, mecanismos para obtenção, reprodução e distribuição de dados, etc.

## ***Integração com Scrum***

O *Scrum* não prevê em seu processo a formalização de como os dados serão mantidos e como a comunicação destes será realizada durante o ciclo de vida do projeto.

No processo *Scrum* estão presentes as seguintes regras que auxiliam na identificação de níveis de acesso aos artefatos do projeto e o desenvolvimento de um plano de comunicação:

- Somente o *Product Owner* pode modificar o *Product Backlog* e este deve estar visível para todos os comprometidos com o projeto;
- O *Sprint Backlog* é criado nas *Planning Meetings* pelo *Scrum Master*, o *Product Owner*, pelo time e, às vezes, por outros envolvidos e este documento deve estar visível para todos os comprometidos com o projeto;
- O *Iteration Burndown chart* é atualizado pelo *Scrum Master* durante as *Daily Scrums* e este também deve estar visível para todos os comprometidos com o time.

A fase de preparação pode ser utilizada para a definição e formalização de como os dados do projeto serão gerenciados ao longo de seu ciclo de vida, bem como as regras de disponibilidade e acessibilidade dos artefatos e outros dados. Como evidência desta prática, sugere-se a utilização de um sistema de informação que se encarregue de versionar os artefatos do projeto e possua as regras de acesso e disponibilidade intrínsecas em suas funcionalidades. Além disto, uma tabela ilustrando os *Stakeholders* do projeto, os artefatos disponíveis para o mesmo e seus níveis de acesso, como e onde estes dados estarão disponíveis.

## ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Habilitar o gerenciamento do *Product Backlog* apenas pelo o *Product Owner* e a visualização do mesmo para todo o time;

- Habilitar o gerenciamento do *Sprint Backlog* por todo o time durante uma Planning Meeting;
- Habilitar a visualização do *Sprint Backlog* por todo o time;
- Habilitar a atualização do Iteration *Burndown chart* pelo *Scrum Master* e a visualização do mesmo por todo o time;
- Proporcionar o versionamento dos artefatos do projeto;
- Possibilitar a criação de regras de acesso para os artefatos do projeto;
- Proporcionar a visualização de uma tabela contendo os *Stakeholders* do projeto, os artefatos disponíveis para o mesmo, seus níveis de acesso, além de como e onde estes dados estarão disponíveis;

#### 3.3.1.2.4. SP 2.4 Planejar os Recursos do Projeto

##### **Descrição**

A definição dos recursos do projeto (mão de obra, equipamentos, materiais e métodos) e das quantidades necessárias para a execução de atividades do projeto, estabelece estimativas iniciais e fornece informações adicionais que podem ser aplicadas na expansão da estrutura analítica utilizada para a gerência do projeto (Chrissis, et al., 2003).

Produtos típicos de trabalho desta área são: requisitos dos recursos humanos necessários com base no tamanho e escopo do projeto, lista de equipamentos e instalações críticas, definição do processo e diagramas de fluxo e lista de requisitos da administração do programa.

##### **Integração com Scrum**

Conforme descrito na seção 3.1.6 o *Scrum* prevê uma fase de preparação realizada antes da primeira *Sprint* e após a definição da visão do projeto e da composição do *Product Backlog* e possui como objetivo a preparação do ambiente de trabalho necessário para a realização do projeto. Se o time de projeto ainda não

foi definido, esta fase pode ser utilizada para isto, bem como para definir os recursos materiais, ferramentas e tecnologias a serem utilizadas no projeto.

Na fase de preparação, os requisitos já estão definidos por meio de *User Stories* e o escopo e tamanho do projeto estão visíveis. Isto auxilia na definição dos recursos necessários a serem alocados para o projeto.

Como evidência, ao final da fase de preparação, o projeto terá o time definido, bem como as ferramentas, máquinas e recursos físicos alocados.

### ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Módulo para a gerência dos recursos humanos no projeto, contendo nome do recurso, papel e responsabilidade;
- Possuir um repositório para a gerência de documentos sobre equipamentos, instalações elétricas, definições de processos, diagramas e outros artefatos necessários ao projeto para o gerenciamento de recursos.

#### 3.3.1.2.5. SP 2.5 Planejar Conhecimento e Habilidades Necessárias

### ***Descrição***

A aquisição de conhecimento e habilidades necessárias para a execução do projeto envolve treinamentos do time de projeto ou contratação de novos recursos de fontes externas.

Os requisitos do time de trabalho são dependentes do conhecimento e habilidades disponíveis para execução do projeto (CMMI Product Team, 2006).

Os produtos de trabalho típicos desta área são: inventário de habilidades necessárias, planejamento de novas contratações e treinamentos.

## ***Integração com Scrum***

O planejamento dos conhecimentos e habilidades necessárias pode ser realizado na fase de preparação. Após a avaliação dos requisitos do projeto e das tecnologias e ferramentas envolvidas no mesmo, é possível obter um inventário de habilidades necessárias ao projeto e mapear este inventário às habilidades dos recursos disponíveis.

Com este mapeamento, é possível verificar a necessidade de contratação de treinamentos ou de outros recursos humanos.

Se não for possível realizar todos os treinamentos necessários na fase de preparação, pode-se utilizar *Spikes* para gerenciar esta necessidade nas outras *Sprints*, ou seja, um Spike pode ser adicionado em uma determinada *Sprint* que prevê o treinamento em alguma tecnologia a ser utilizada.

## ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Possuir um inventário de habilidades necessárias ao projeto, para identificar as habilidades que estão faltando para completar o projeto e auxiliar na contratação de pessoas ou treinamentos que proporcionarão as habilidades necessárias para o projeto;
- Permitir a criação de *Spikes* específicos para treinamento, informando quais os recursos humanos envolvidos e em qual período o treinamento seria necessário;
- Permitir a criação de *Spikes* específicos para contratação de recursos humanos, informando as habilidades necessárias e a partir de que período seria necessário ter este recurso humano contratado;

### 3.3.1.2.6. SP 2.6 Planejar o Envolvimento dos *Stakeholders*

Segundo o PMBOK (PMBOK, 2004), os *Stakeholders* são pessoas ou organizações, como clientes, patrocinadores, etc. que estejam envolvidas ativamente no projeto ou tenham seu interesse afetado positiva ou negativamente pela execução ou término do mesmo. *Stakeholders* podem exercer influência sobre o projeto e suas entregas.

Os *Stakeholders* de um projeto são identificados durante todas as fases do ciclo de vida do mesmo. Eles são identificados pelo grau de envolvimento ou interesse nas atividades específicas do projeto.

Uma matriz bidimensional contendo em um eixo os *Stakeholders* e no outro eixo as atividades do projeto é um formato conveniente para o acompanhamento desta identificação (CMMI Product Team, 2006).

Exemplos do tipo de material a ser incluído no plano para a interação com o *Stakeholder* incluem o seguinte: Lista de todos os *Stakeholders* relevantes, motivo do envolvimento do mesmo, regras e responsabilidades dos *Stakeholders*, relacionamento entre eles, importância dos mesmos e recursos necessários para garantir a interação de uma destas pessoas.

O produto de trabalho típico desta prática é o plano de envolvimento do *Stakeholder*.

### ***Integração com Scrum***

No *Scrum*, os papéis dentro do time bem como suas responsabilidades são bem definidos, conforme descrito na seção 4.3.

Na fase de planejamento, é necessária a identificação do *Product Owner* bem como a identificação do *Scrum Master*. Se o time ainda não foi definido, na fase de preparação é possível identificar os papéis técnicos de cada integrante do time e atribuir responsabilidades aos mesmos.

Além disto, nas reuniões de revisão, realizadas no final de cada *Sprint*, todos os *Stakeholders* são convidados a participar de uma demonstração do produto. As reuniões de revisão são acordadas durante o planejamento de cada *Sprint* e garantem o envolvimento contínuo dos *Stakeholders* durante todo o ciclo de vida do projeto.

Durante a *Planning Meeting* de uma *Sprint*, pode-se identificar a necessidade do envolvimento de algum *Stakeholder* para esclarecer dúvidas do time sobre algum assunto que o *Product Owner* não tem conhecimento.

Nestes casos, o *Product Owner* sabe indicar quem é o contato para sanar a dúvida e pode-se, então, utilizar *Spikes* para planejar o envolvimento do *Stakeholder* requerido. *Spikes* são tarefas incluídas no *Sprint Backlog* que tem por objetivo responder dúvidas ou obter conhecimento sobre algum assunto, conforme descrito na seção 2.1.4.5.

O *Scrum* não prevê a formalização da identificação preliminar dos *Stakeholders*, para suprir esta limitação, sugere-se a criação de um documento criado na fase de planejamento do projeto, ligando pessoas aos seus papéis e responsabilidades. O *Product Owner* irá auxiliar a identificar as pessoas que não estão envolvidas diretamente com o projeto e não possuem papéis formalizados no *Scrum*. Este documento poderá ser atualizado durante as reuniões de planejamento das *Sprints* se um novo *Stakeholder* for envolvido.

Assim, com a adoção deste documento bem como as reuniões de revisão e criação de *Spikes*, entende-se que o *Scrum* é capaz de atender esta prática específica.

### ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Possuir uma lista contendo os *Stakeholders* relevantes e seu envolvimento no projeto;
- Permitir a criação e atualização de usuários no sistema que representam os *Stakeholders*;
- Permitir a criação e atualização de funções, para serem ligadas aos *Stakeholders*. Estas funções poderiam possuir um texto ou documento anexo informando as regras e responsabilidades daquela função;
- Permitir o envio de e-mails ou solicitações de reunião para todos ou parte dos *Stakeholders*;

- Permitir a criação de *Spikes* específicos para o envolvimento de um ou mais *Stakeholders*;

#### 3.3.1.2.7. SP 2.7 Estabelecer o Plano de Projeto

##### **Descrição**

Um plano documentado que trate todos os itens relevantes para o planejamento é necessário para conseguir entendimento, compromisso e desempenho mútuo dos indivíduos, dos grupos e das organizações que devem executar ou dar suporte aos planos. O plano gerado para o projeto define todos os aspectos do esforço, unindo tudo de uma maneira lógica: considerações sobre o ciclo de vida do projeto, tarefas técnicas e de gestão, orçamentos e cronogramas marcos, gerenciamento de dados, identificação de riscos, requisitos de recursos e de habilidades e identificação de *Stakeholder* e interação com o mesmo. As descrições da infra-estrutura incluem relacionamentos de responsabilidades e autoridades para a equipe de projeto, para a equipe de gerenciamento e para as organizações de suporte (CMMI Product Team, 2006).

O produto típico de trabalho desta área é o plano geral de projeto.

##### **Integração com Scrum**

Por ser um processo iterativo, o plano de um projeto *Scrum*, possuirá dois níveis: um plano macro e os planos das iterações. Ambos os planos, descritos a seguir, são compostos de artefatos previstos no *Scrum* e de artefatos que foram propostos neste trabalho para atender a algumas práticas específicas do CMMI.

O Plano de Projeto em nível macro conterá todos os artefatos gerados e aprovados na fase de planejamento e/ou fase de preparação:

- A visão do projeto;
- O *Product Backlog*, priorizado, escrito em forma de *User Stories* e com os pontos associados;

- O *Release Plan*, contendo as datas de início e fim de cada *Sprint* bem como o orçamento do projeto;
- O *Risk Backlog*, contendo os riscos identificados, priorizados, analisados e com ações já definidas;
- Documento de formalização do plano de gerenciamento de dados do projeto;
- A identificação das habilidades necessárias, o mapeamento destas necessidades com os recursos já disponíveis e o plano de aquisição de recursos humanos e/ou treinamentos;
- Documento dos *Stakeholders* do projeto bem como suas responsabilidades, incluindo a identificação do *Product Owner*, do *Scrum Master* e dos componentes do time.

O Plano em nível de iterações é criado e aprovado nas *Planning Meetings* de cada *Sprint* e envolve os seguintes artefatos:

- O *Product Backlog* para alguma possível alteração;
- O *Sprint Backlog*, contendo as tarefas da *Sprint* em horas com os recursos associados;
- O *Risk Backlog* para atualização;

O *Scrum* não prevê a criação de um documento chamado Plano de Projeto, porém todos estes artefatos descritos, alguns oriundos do *Scrum* outros sugeridos neste trabalho compõe um plano para o projeto, atendendo o objetivo desta prática específica.

### ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Produzir um documento que contenha todos os artefatos do projeto que sejam necessários para formar o plano de projeto em nível macro;
- Produzir um documento que contenha todos os artefatos do projeto que sejam necessários para formar o plano de projeto em nível de iteração;

- Produzir um documento que contenha todos os artefatos do *Scrum* que sejam necessários para formar o plano de projeto, tanto em nível macro como em nível de iteração. Ele poderia conter artefatos como a visão do projeto, o *Product Backlog* priorizado, o *Sprint Backlog*, etc.

A geração de um documento pela ferramenta contendo as últimas atualizações dos artefatos que compõem o plano de projeto é crucial para que este nunca esteja desatualizado e para que não seja necessária a criação de vários documentos contendo a mesma informação, já que uma das boas práticas dos métodos ágeis é evitar a criação de documentos desnecessários ou replicados.

### 3.3.1.3. SG 3 Obter Comprometimento com o Plano

Este objetivo específico visa estabelecer e manter o comprometimento com o plano de projeto (Chrissis, et al., 2003).

#### 3.3.1.3.1. SP 3.1 Revisar os Planos que Afetam o Projeto

##### **Descrição**

Esta prática visa revisar todos os planos que afetam o projeto para entender os comprometimentos do projeto (Chrissis, et al., 2003).

Planos desenvolvidos em outras áreas de processo podem conter informações similares aos documentos da área de Planejamento de Projeto.

Todos os planos que afetam o projeto devem ser revistos para assegurar um entendimento comum do escopo, objetivos, regras e relacionamentos que são requeridos para o projeto ser bem sucedido (CMMI Product Team, 2006).

O produto de trabalho típico desta prática é: registro das revisões dos planos que afetam o projeto.

## ***Integração com Scrum***

Há duas reuniões previstas no *Scrum* que visam, entre outros objetivos, revisar e atualizar os planos que afetam o projeto: a *Planning Meeting* e a *Review Meeting*.

Na *Planning Meeting*, que ocorre no início de cada *Sprint*, o *Product Backlog* e o *Release Plan* são revisados pelo *Product Owner*, o *Scrum Master* e o time. Após a revisão, estes artefatos são atualizados de acordo com a realidade do projeto. Esta reunião pode ser utilizada para revisão e atualização do *Risk Backlog* também.

Na *Review Meeting* é comum a identificação de novas *User Stories* a serem agregadas no projeto bem como a remoção de outras e repriorização do *Product Backlog*. Nesta reunião, é realizada a revisão do *Product Backlog* e do *Release Plan* pelos integrantes da reunião e o *Product Owner* atualiza estes artefatos de acordo com a decisão do projeto.

## ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Possuir um versionamento dos artefatos que afetam o projeto, para gerar evidência da revisão dos mesmos;
- Permitir a visualização de uma tabela contendo todos os planos que afetam o projeto, a quantidade de versões de cada plano e as datas da primeira e das últimas versões de cada plano;
- Permitir a visualização das versões de um plano, com a data e descrição de cada mudança;
- Permitir a visualização de uma determinada versão de um plano;

### 3.3.1.3.2. SP 3.2 Conciliar Níveis de Trabalho e Recursos

#### **Descrição**

Esta prática visa ajustar o plano do projeto para refletir recursos estimados e disponíveis (Chrissis, et al., 2003).

O objetivo desta prática é obter o comprometimento dos *Stakeholders* e conciliar as diferenças entre os recursos que foram estimados e os que estão realmente disponíveis.

A conciliação normalmente termina com a negociação de mais recursos, quando for encontrado um modo de aumentar a produtividade com a realização de outsourcing, com o ajuste do perfil da equipe ou com a atualização de todos os planos que afetam o projeto ou os cronogramas (CMMI Product Team, 2006).

Os produtos de trabalho típicos desta prática são: métodos e parâmetros de estimativa correspondentes atualizados, orçamentos revisados, cronogramas revisados, lista de requisitos revisada e acordos com *Stakeholders* renegociados.

#### **Integração com Scrum**

Na *Review Meeting*, os resultados da *Sprint* são revisados conforme o *Sprint Backlog* acordado na *Planning Meeting*.

Muitas vezes, por problemas técnicos ou de estimativas, algumas *User Stories* podem não ficar prontas ao final da *Sprint*. Nestes casos, deve haver uma negociação entre o *Product Owner*, o *Scrum Master* e possivelmente outros *Stakeholders* do projeto buscando rever o que não foi alcançado e como isto irá ser tratado no plano do projeto.

Além disto, é comum notar a necessidade de novas funcionalidades para o produto que não foram identificadas anteriormente. Nestes casos, uma negociação de escopo, tempo ou custos também será necessária.

O resultado desta negociação irá variar de acordo com o que é melhor para o projeto e a decisão será tomada com o consentimento de todos os *Stakeholders* presentes na reunião.

Como evidência desta negociação tem-se o *Product Backlog* revisado e atualizado com as novas *User Stories*, as que não ficaram prontas na *Sprint* e com a remoção de algumas *User Stories* quando necessário. Estas alterações no *Product Backlog* afetam diretamente o *Release Plan* que também deve ser revisto e atualizado.

### ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Possibilitar o gerenciamento do *Product Backlog*, fornecendo uma forma simples de remoção e adição de novas *User Stories*;
- Possuir uma funcionalidade para colocar *User Stories* com baixa prioridade como fora do escopo do projeto;
- Mostrar horas estimadas e realizadas, para que o time possa atualizar as horas estimadas para futuras *User Stories* de acordo com uma base histórica;

#### 3.3.1.3.3. SP 3.3 Obter Comprometimento do Plano

### ***Descrição***

Esta prática possui o objetivo de obter o comprometimento dos *Stakeholders* relevantes responsáveis pela execução e suporte à execução do plano (Chrissis, et al., 2003).

Para se obter o comprometimento dos *Stakeholders* é preciso haver interação entre todas as pessoas relevantes que estejam ligadas, internamente ou externamente, ao projeto. Os *Stakeholders* comprometidos com o projeto precisam ter a segurança de que o trabalho possa ser feito dentro das restrições existentes para custo, tempo e desempenho.

Os produtos de trabalho típicos desta prática são: requisitos documentados para o comprometimento e comprometermos documentados.

## ***Integração com Scrum***

Obter o comprometimento com o plano de projeto faz parte do processo *Scrum* visto que este possui iterações curtas e contato constante com o cliente e demais interessados no produto. Estes fatores auxiliam a identificar se os objetivos estão sendo alcançados e, conseqüentemente, se os envolvidos estão realmente comprometidos com o plano de projeto e corrigir desvios relacionados a isto.

Nas *Planning Meetings*, o plano para a próxima iteração do projeto é desenvolvido, baseado no plano macro do projeto com o consentimento do time, do *Scrum Master*, do *Product Owner* e de outros envolvidos, quando necessário. Para evidenciar o comprometimento dos envolvidos com o plano desenvolvido, pode-se agregar no processo a aprovação deste plano por todos os integrantes da reunião.

## ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Enviar uma notificação solicitando a aprovação do plano desenvolvido durante a *Planning Meeting* de uma *Sprint*.
- Possuir uma funcionalidade para que os usuários do sistema possam visualizar as tarefas das *User Stories* que estão associadas a eles;
- Possuir uma funcionalidade para que os usuários do sistema possam visualizar o andamento das tarefas das *User Stories*;
- Possuir uma funcionalidade para que os usuários do sistema possam visualizar as *User Stories* prontas, em andamento ou ainda não inicializadas;
- Enviar um alerta aos *Stakeholders* quando uma *User Story* com prioridade mais baixa tenha sido iniciada antes de outra com prioridade maior;

### **3.3.2. Monitoramento e Controle de Projeto (*Project Monitoring and Control - PMC*)**

#### 3.3.2.1. SG 1 Monitorar o Projeto em Relação ao Plano

O propósito deste objetivo específico é monitorar o progresso e o desempenho do projeto em relação ao seu plano (Chrissis, et al., 2003).

##### 3.3.2.1.1. SP 1.1 Monitorar os Parâmetros de Planejamento do Projeto

#### ***Descrição***

Esta prática específica visa monitorar os valores reais dos parâmetros de planejamento do projeto de acordo com o plano do mesmo (Chrissis, et al., 2003).

Os parâmetros de planejamento de projeto constituem os indicadores típicos de desempenho e de progresso do projeto e incluem atributos de produtos de trabalho e de tarefas, custo, esforço e cronograma (CMMI Product Team, 2006).

O monitoramento normalmente envolve medir os valores reais dos parâmetros de planejamento de projeto, comparar os valores reais com os estimados no plano e identificar os desvios significativos. (Chrissis, et al., 2003).

Os produtos de trabalho típicos são: registros de desempenho de projeto e registros de desvios significativos.

#### ***Integração com Scrum***

No *Scrum*, há dois artefatos utilizados para monitorar os parâmetros de planejamento do projeto: o *Release Burndown chart* e o *Iteration Burndown chart*.

Conforme comentado na seção 2.1.4.11, o *Release Burndown chart* é um gráfico composto pelo número de pontos restantes para finalizar o projeto no eixo vertical e as iterações previstas no eixo horizontal. Este gráfico é atualizado no final de cada iteração e mostra a quantidade de trabalho restante para a conclusão do projeto (Cohn, 2005).

Conforme descrito na seção 2.1.4.13, o *Iteration Burndown chart* é um gráfico composto pelas horas restantes de esforço estimado para a finalização da *Sprint* e pelos dias que compõe a mesma. Este gráfico é atualizado diariamente pelo o *Scrum Master* com o progresso do time naquele dia (Cohn, 2005).

Ambos os gráficos são simples, porém contém informações muito significativas para o monitoramento do projeto, visto que é fácil identificar os desvios do projeto em relação ao que foi planejado. Além disto, é possível identificar se os desvios são causados por erros de estimativas ou por interferências externas que estejam afetando a produtividade do time.

A partir destes gráficos é possível criar planos e realizar negociações para tratar os desvios significantes identificados.

O *Scrum* não formaliza em seu processo nenhuma prática para o acompanhamento do orçamento do projeto. Para tanto, sugere-se a criação de um *Burndown chart* que reflita a situação do projeto em relação ao orçamento estimado e o total gasto. A figura 9 mostra como seria um *Burndown chart* de custos, onde no eixo vertical tem-se o total estimado e no eixo horizontal as *Sprints* previstas para o projeto.

Este gráfico poderia ser atualizado no final de cada *Sprint* a partir das horas realizadas, com o intuito de perceber desvios em relação ao planejado e realizar negociações e/ou tomar ações corretivas para os desvios identificados.

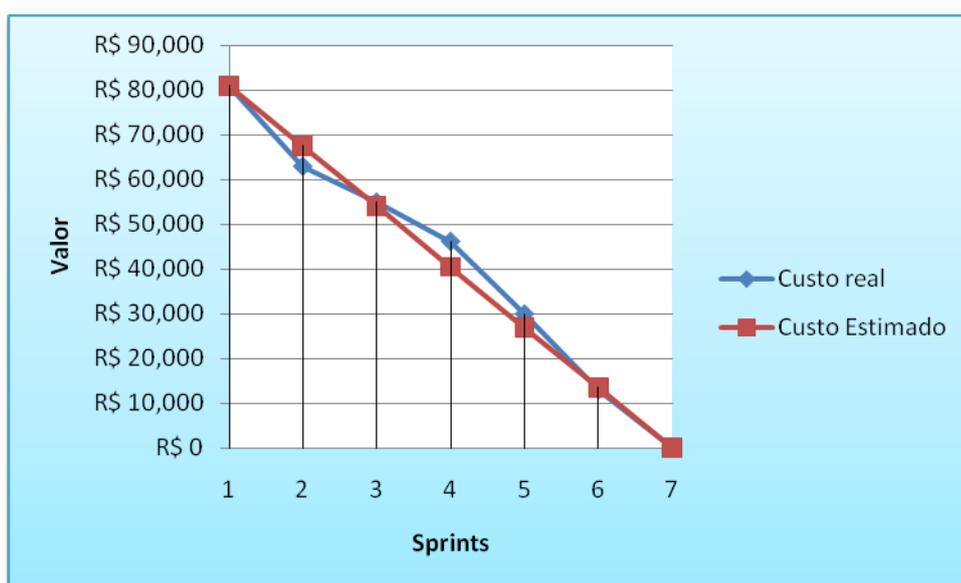


Figura 9: *Burndown chart* ilustrando o andamento dos custos do projeto

Esta sugestão poderia ser estendida de forma a possuir também o conceito de valor agregado (PMBOK, 2004) que contempla o valor do trabalho terminado em termos do orçamento aprovado atribuído a este trabalho.

### ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Proporcionar a inserção de horas realizadas e atualização das horas restantes nas tarefas de uma *User Story* e atualizar o *Iteration Burndown chart* automaticamente após cada mudança.
- Gerar um *Burndown chart* de custos baseado no orçamento estimado e no número de iterações do projeto;
- Permitir a inserção do orçamento realizado em uma *Sprint* e atualizar o custo real da *Sprint* automaticamente no *Burndown chart* de custos;
- Enviar uma notificação ao *Product Owner* e *Scrum Master* caso o custo, esforço ou pontos reais possuam algum desvio significativo de acordo com o que havia sido estimado;
- Atualizar o *Release Burndown chart* automaticamente quando uma *User Story* for dada como pronta.

#### 3.3.2.1.2. SP 1.2 Monitorar os Comprometimentos

### ***Descrição***

O objetivo desta prática específica é monitorar os comprometimentos em relação com aqueles identificados no plano de projeto (Chrissis, et al., 2003).

O produto típico de trabalho é o registro das revisões dos comprometimentos.

## ***Integração com Scrum***

O *Scrum* prevê as *Daily Scrums* entre o time e o *Scrum Master* que auxiliam no monitoramento dos compromettimentos feitos em relação ao plano de projeto, ou seja, a realidade do projeto é analisada de acordo com o plano da *Sprint*, possíveis desvios ao que foi comprometido são identificados e ações corretivas podem ser tomadas.

Além disto, no final de cada *Sprint* a *Review Meeting* também fornece uma forma de monitoração dos compromettimentos realizados, envolvendo o *Product Owner*, o *Scrum Master*, o time e outros envolvidos.

## ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Possuir uma funcionalidade para que, durante a *Daily Scrum*, o *Scrum Master* insira, para cada membro do time, o que aquele membro fez desde a última *Daily Scrum*, o que ele planeja fazer até a próxima *Daily Scrum* e quais são os seus impedimentos. Estes impedimentos são atualizados na lista de Impediments *Backlog* e o relato do time é salvo para servir como evidência;
- Enviar um alerta ao *Scrum Master* quando as tarefas de uma *User Story* com prioridade mais baixa tenham sido iniciadas antes de outra com prioridade maior;

### 3.3.2.1.3. SP 1.3 Monitorar os Riscos do Projeto

#### ***Descrição***

Esta prática específica tem por objetivo monitorar os riscos identificados no plano de projeto.

O produto típico de trabalho é os registros de monitoramento de riscos do projeto.

### ***Integração com Scrum***

Conforme apresentado na seção 3.3.2.2, foi proposta a criação de um documento chamado *Risk Backlog* para dar apoio ao gerenciamento dos riscos do projeto.

Este documento, contendo a identificação e análise dos riscos, deverá ser revisado e atualizado nas *Planning Meetings* e *Review Meetings*.

Durante a *Planning Meeting* deve-se verificar quais riscos ainda estão ativos, se os responsáveis estão realizando as ações identificadas de acordo com a estratégia escolhida, se a prioridade, probabilidade e o impacto dos riscos mudaram e deve-se também adicionar novos riscos identificados e analisados.

Durante as *Review Meetings*, podem ser identificados novos riscos com o *feedback* do cliente e com o resultado da *Sprint*. Esses novos riscos devem ser analisados e inseridos no *Risk Backlog*.

Assim, como evidência do monitoramento dos riscos do projeto tem-se as revisões do *Risk Backlog* realizadas nas reuniões previstas no *Scrum*.

### ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Possuir um *Gadget*<sup>6</sup> que mostre os riscos mais prioritários ou da iteração para que os usuários possam colocar na página inicial do sistema;
- Enviar os riscos por e-mail semanalmente para o *Product Owner*, o *Scrum Master* e o time;

---

<sup>6</sup> Mini-aplicativos criados para proporcionar informações e dados úteis ou para melhorar uma aplicação ou um serviço.

- Enviar alertas sobre riscos que tiveram sua data de expiração atingida e não foram solucionados;
- Guardar um histórico de riscos para mostrar a quantidade de riscos encontrada no início do projeto, a quantidade de riscos fechados, os riscos que foram adicionados durante o projeto e a quantidade de riscos aberta;
- Mostrar um gráfico contendo a quantidade de riscos abertos e fechados no decorrer do projeto;

#### 3.3.2.1.4. SP 1.4 Monitorar o Gerenciamento de Dados

##### ***Descrição***

Esta prática específica visa monitorar o gerenciamento de dados em relação ao plano de projeto (Chrissis, et al., 2003).

O produto típico de trabalho são os registros de monitoramento de dados.

##### ***Integração com Scrum***

Um dos objetivos do *Scrum Master* é garantir que o processo esteja sendo seguido pelos envolvidos no projeto. Portanto, o *Scrum Master* seria indicado para monitorar regularmente se os dados do projeto estão acessíveis conforme planejado e atualizar possíveis mudanças no gerenciamento dos artefatos, bem como inclusão de novos. Como evidência desta prática, tem-se a revisão e atualização do plano de gerenciamento de dados.

Porém, possivelmente seria apropriada a adoção de uma prática alternativa a esta solução, visto que se propõe um formalismo inadequado para os propósitos do *Scrum*.

## ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter a seguinte funcionalidade que incentivariam o cumprimento desta prática específica:

- Proporcionar o versionamento do plano de gerenciamento de dados com as revisões de alteração e visualização feitas pelo *Scrum Master*.

### 3.3.2.1.5. SP 1.5 Monitorar o Envolvimento de *Stakeholders*

#### ***Descrição***

Esta prática específica visa monitorar o envolvimento dos *Stakeholders* em relação ao plano de projeto (Chrissis, et al., 2003).

Uma vez que os *Stakeholders* sejam identificados e seu envolvimento no projeto esteja especificado no plano, esse envolvimento deve ser monitorado para assegurar que as interações apropriadas estejam ocorrendo (CMMI Product Team, 2006).

O produto típico de trabalho é o registro do envolvimento dos *Stakeholders*.

#### ***Integração com Scrum***

Como já mencionado, o *Scrum* prevê reuniões periódicas, ao final de cada *Sprint*, que tem como um dos objetivos monitorar o envolvimento dos *Stakeholders* do projeto através de seu *feedback* e da monitoração do comprometimento com o plano. Nestas reuniões, chamadas de *Review Meetings*, estão presentes o *Product Owner*, o *Scrum Master*, o time e outros *Stakeholders* que poderão expressar sua opinião sobre o andamento do projeto e agregar sugestões e melhorias.

Além disto, é o *Scrum Master* pode monitorar o envolvimento do time durante as *Daily Scrums*, através do *feedback* do time.

## ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Mostrar uma lista das *Daily Scrums* feitas no decorrer do projeto para que isto possa ser utilizado como evidência de registro do envolvimento dos *Stakeholders*;
- Possuir um repositório para que sejam colocados documentos criados durante a *Review Meeting*, como uma ata ou um vídeo mostrando a demonstração do incremento funcional.

### 3.3.2.1.6. SP 1.6 Conduzir Revisões de Progresso

#### ***Descrição***

Esta prática específica visa à revisão periódica do progresso, desempenho e problemas do projeto.

Revisões de progresso são revisões no projeto para manter os *Stakeholders* informados. Estas revisões de projeto podem ser informais e não precisam estar especificadas nos planos de projeto.

O produto típico de trabalho é os resultados documentados das revisões de projeto.

#### ***Integração com Scrum***

Conforme descrito anteriormente, o *Scrum* prevê a criação de dois gráficos que proporcionam uma visão do progresso do projeto: o *Iteration Burndown chart* e o *Release Burndown chart*.

Ambos gráficos são atualizados periodicamente pelo *Scrum Master* e devem estar disponíveis para todos os envolvidos com o projeto.

O *Iteration Burndown chart* é atualizado diariamente e para evidenciar sua revisão pelos principais envolvidos com o projeto, pode-se monitorar o acesso ao

mesmo, se tiver acessível a partir de algum sistema de informação, ou enviar o artefato por e-mail com solicitação de confirmação de leitura.

O *Release Burndown chart* é exposto e revisado na *Review Meeting*, para todos os envolvidos do projeto.

### **Funcionalidades Sugeridas**

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Enviar o *Iteration Burndown chart* periodicamente por e-mail para o *Product Owner*, solicitando confirmação de leitura.
- Enviar o *Release Burndown chart* por e-mail ao final de cada *Sprint* para os *Stakeholders*, solicitando confirmação de leitura.
- Possuir um repositório para que sejam colocados documentos criados durante a reunião de Review, como uma ata ou um vídeo mostrando a demo;

#### 3.3.2.1.7. SP 1.7 Conduzir Revisões em Marcos

### **Descrição**

Esta prática específica visa revisar as realizações e resultados do projeto em marcos selecionados do projeto (Chrissis, et al., 2003).

Revisões de um marco do projeto são planejadas durante a fase de planejamento e são tipicamente revisões formais (CMMI Product Team, 2006).

O produto típico de trabalho é os resultados documentados das revisões de marcos.

### **Integração com Scrum**

A *Review Meeting* pode ser considerada como um marco no projeto, que indica o término de uma *Sprint* e a finalização de um incremento funcional.

As *Review Meetings* podem ser planejadas e agendadas durante a fase de planejamento do projeto devido ao *Release Plan*, que indica as datas de início e fim estimadas para cada *Sprint*.

Estas reuniões têm por objetivo justamente revisar o que foi desenvolvido na última *Sprint*, através de uma demonstração e discutir possíveis mudanças no curso do projeto.

Como evidência desta prática tem-se a atualização do *Product Backlog* com as *User Stories* que foram concluídas na *Sprint* e com possíveis novas *User Stories* identificadas.

### ***Funcionalidades Sugeridas***

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Possuir uma funcionalidade para colocar as *User Stories* que não foram completadas na *Sprint* como incompletas no *Product Backlog*, para que as estimativas do projeto sejam revistas;
- Possuir um *Gadget* para lembrar o *Product Owner* caso ele precise rever as estimativas;
- Enviar uma notificação informando o *Product Owner* caso ele precise rever as estimativas;
- Mostrar a quantidade de *User Stories* planejadas, a quantidade das que foram entregues e a quantidade de novas *User Stories* identificadas ao final de cada *Sprint* para que isto sirva como evidência;

#### 3.3.2.2. SG 2 Gerenciar as Ações Corretivas até o Encerramento

Este objetivo específico procura gerenciar as ações corretivas quando o desempenho do projeto e seus resultados desviam significativamente do planejado (Chrissis, et al., 2003).

### 3.3.2.2.1. SP 2.1 Analisar *Issues*

#### **Descrição**

Esta prática específica tem por objetivo coletar e analisar as *issues* e determinar as ações corretivas necessárias para endereçar as mesmas (Chrissis, et al., 2003).

O produto de trabalho típico é uma lista de *issues* que necessitam de ações corretivas.

#### **Integração com Scrum**

Nas *Daily Scrums*, todos os integrantes do time e o *Scrum Master* devem falar sobre os impedimentos identificados em seu trabalho que estão obstruindo o progresso do projeto (Schwaber, et al., 2001).

Conforme descrito na seção 2.1.4.12, estes impedimentos são registrados em uma lista, conhecida como *Impediments Backlog*, e o *Scrum Master* é o responsável pela priorização e gerência dos itens da lista, além de assegurar que os impedimentos sejam solucionados e removidos do *Impediments Backlog* (Sutherland, 2006).

Portanto, como o *Scrum* já possui um artefato que contém as *issues* do projeto encontrados ao longo de seu ciclo de vida, pode-se considerar que esta prática esteja satisfeita.

#### **Funcionalidades Sugeridas**

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Proporcionar o cadastro de itens dentro do *Impediments Backlog*, associando o membro do time bloqueado (quando houver), prioridade e membro que deve tomar alguma ação para corrigir o mesmo;
- Possuir um *Gadget* que mostre os impedimentos mais prioritários para que os usuários possam colocar na página inicial do sistema;

- Enviar uma notificação ao usuário endereçado a tomar alguma ação corretiva para um impedimento informando-o sobre o impedimento e a data final para que o mesmo seja resolvido;
- Habilitar uma funcionalidade que auxilie na criação de impedimentos durante a *Daily Scrum*, associando automaticamente o membro do time que está bloqueado por este impedimento;

#### 3.3.2.2.2. SP 2.2 Tomar Ações Corretivas

##### **Descrição**

Esta prática específica tem por objetivo tomar as ações corretivas para as *issues* identificadas (Chrissis, et al., 2003).

O produto típico de trabalho é o plano de ações corretivas.

##### **Integração com Scrum**

Conforme descrito anteriormente, o *Impediments Backlog* é um artefato presente no *Scrum* cujos impedimentos do projeto são descritos e priorizados, entretanto, somente estas informações não são suficientes para acompanhar as ações corretivas a serem realizadas para a remoção destes impedimentos.

Assim, sugerimos a adição de duas informações no *Impediments Backlog*: a ação corretiva associada aos itens do mesmo e uma data estimada de correção.

Estas informações podem ser adicionadas, revisadas e atualizadas pelo *Scrum Master* nas *Daily Scrums*.

Portanto, com a adição destas informações no *Impediments Backlog*, esta prática específica pode ser satisfeita.

##### **Funcionalidades Sugeridas**

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Permitir a inserção de uma data estimada para correção e ações corretivas a serem tomadas dentro de cada impedimento no *Impediments Backlog*;

#### 3.3.2.2.3. SP 2.3 Gerenciar as Ações Corretivas

##### **Descrição**

Esta prática específica tem por objetivo gerenciar as ações corretivas até que estas solucionem as *issues* (Chrissis, et al., 2003).

O produto típico de trabalho é os resultados das ações corretivas tomadas.

##### **Integração com Scrum**

Novamente, somente o *Impediments Backlog* da maneira que ele é proposto no *Scrum*, não proporciona as informações necessárias e evidências de quais ações corretivas foram tomadas para a resolução dos problemas.

Para tanto, sugerimos que os itens do *Impediment Backlog* não sejam removidos do mesmo quando forem resolvidos e sim que eles sejam destacados de alguma forma, indicando que o problema já foi resolvido e não é necessária nenhuma ação adicional.

Além disto, se a ação corretiva planejada não foi eficaz e outra foi tomada para a resolução da *issue*, o item do *Impediment Backlog* deve ser atualizado com a ação que realmente produziu os resultados esperados. Esta atualização pode ser realizada pelo *Scrum Master* nas *Daily Scrums*.

Assim, com estas mudanças no *Impediments Backlog*, esta prática específica pode ser satisfeita.

##### **Funcionalidades Sugeridas**

Um sistema colaborativo de gestão de projetos poderia conter as seguintes funcionalidades que incentivariam o cumprimento desta prática específica:

- Armazenar todos os impedimentos fechados no Impediment *Backlog*;
- Aceitar a reabertura de um impedimento;
- Ao fechar um impedimento que havia sido reaberto, obrigar a especificação de uma nova ação corretiva;

### 3.4. DESCRIÇÃO DO PROJETO

O projeto do sistema colaborativo contemplado neste trabalho utilizou o Scrum e os processos descritos na integração proposta entre a metodologia *Scrum* e as práticas de PP e PMC do CMMI, permitindo assim uma experimentação efetiva do processo. Ao utilizar a metodologia *Scrum*, o projeto incorporou um ciclo de vida iterativo e incremental

O projeto possuiu as fases de planejamento, preparação, desenvolvimento e entrega.

A fase de planejamento foi executada durante a disciplina de Estágio Profissional. As mais importantes funcionalidades sugeridas descritas na proposta de integração se tornaram as *User Stories* do *Product Backlog*. Estas *User Stories* foram priorizadas e utilizou-se a técnica de *User Story Points* para estimativa de tamanho das mesmas. Ao final da fase de planejamento, foram definidas a visão, o *Product Backlog*, o *Release Plan*, o *Risk Backlog* e a arquitetura do projeto. O projeto foi dividido em 6 sprints de 2 semanas, tendo como base uma velocidade estimada de 35 pontos por sprint.

Durante a fase de preparação, foram adquiridos todos os recursos físicos necessários ao projeto. Além disto, houve um período de aprendizado da tecnologia envolvida e aquisição das habilidades e conhecimentos necessários para o desenvolvimento do projeto.

Durante a fase de desenvolvimento, o time implementou, testou e integrou as *User Stories* de acordo com suas prioridades, além de fazer as reuniões definidas no processo. Durante a fase de planejamento, o projeto havia sido dividido em 6 Sprints, porém, na fase de desenvolvimento, uma das Sprints precisou ser postergada por duas semanas e fez com que a última Sprint precisasse ser cancelada. A fase de desenvolvimento explica estas mudanças com maiores detalhes.

As User Stories foram sendo desenvolvidas de acordo com suas prioridades no decorrer das Sprints.

Na fase de entrega, o produto foi estabilizado e o trabalho foi preparado para entrega e apresentação.

### **3.4.1. Processo de trabalho**

O time procurou seguir todas as práticas propostas na integração que condiziam com o escopo do trabalho.

Por ser um time muito pequeno e estar sempre em constante comunicação, foi acordado que as *Daily Scrums* não seriam aplicadas. Além disto, não realizamos o planejamento nem monitoramento dos custos envolvidos, já que este é um trabalho acadêmico e não possui características comerciais.

Todas as funcionalidades desenvolvidas durante uma Sprint eram utilizadas na próxima, contendo as informações do projeto. Por exemplo, a funcionalidade de gerenciamento de Sprints foi desenvolvida na Sprint 1 e a partir da Sprint 2 todas as Sprints do projeto foram mapeadas na ferramenta e esta funcionalidade foi utilizada para o gerenciamento das mesmas.

Enquanto as funcionalidades ainda não estavam prontas para utilização, o time utilizou ferramentas do Microsoft Office para realizar o planejamento, monitoramento e controle das Sprints.

### **3.4.2. Fase de planejamento**

#### **3.4.2.1. Visão**

Desenvolver um sistema colaborativo para gestão de projetos integrando as práticas do *Scrum* e os objetivos específicos das áreas de processos Planejamento de Projeto e Monitoramento e Controle de Projeto do CMMI.

### 3.4.2.2. Product Backlog

A tabela 4 expõe o *Product Backlog* inicial do projeto, devidamente priorizado e pontuado conforme a técnica de *User Story Points*.

A priorização do *Product Backlog* foi baseada nas práticas do CMMI a fim de obter funcionalidades que auxiliem na satisfação das áreas de processos Planejamento de Projeto e Monitoramento e Controle de Projeto. Assim, a *User Story* mais prioritária de cada prática específica está entre as mais prioritárias do projeto, contemplando primeiramente as práticas do Planejamento de Projeto, e, após, as práticas do Monitoramento e Controle de Projeto.

A pontuação foi realizada utilizando a técnica de *User Story Points*, onde o *Product Backlog* foi analisado em busca da *User Story* menos complexa. A seguir, todas as *User Stories* foram analisadas, em comparação com a menos complexa, ou seja, para cada uma a seguinte pergunta foi feita: “quantas vezes a *User Story* x é mais complexa em relação a y. Como resposta, foi utilizado um número da seqüência de fibonacci de 1 a 13.

ID	<i>User Story</i>	Pontos
PP1.1.1	Proporcionar um <i>template</i> para a criação de <i>User Stories</i> e uma forma simples e prática de manutenção	5
PP1.2.1	Permitir a pontuação das <i>User Stories</i> no <i>Product Backlog</i>	2
PP1.3.1	Disponibilizar <i>checklists</i> com o que deve ser feito em cada fase para seguir o ciclo de vida do <i>Scrum</i>	5
PP1.4.1	Gerar automaticamente a base histórica, com a velocidade do time em pontos por <i>Sprint</i> a partir dos dados das <i>Sprints</i> já realizadas, a fim de proporcionar dados consistentes com a realidade para uso nas estimativas da próxima <i>Sprint</i> .	8
PP2.1.1	Permitir a criação de tarefas para uma determinada <i>User Story</i> , com prioridade, estimativa, recursos alocados e dependência entre esta e outras tarefas	5
PP2.2.1	Possuir um <i>Risk Backlog</i> , para conter os riscos, que podem ser criados com prioridade, categoria, descrição, impacto, probabilidade de ocorrência, estratégia, ação corretiva e usuário responsável pelo mesmo	3

<b>PP2.3.1</b>	Proporcionar o versionamento dos artefatos do projeto	1
<b>PP2.4.1</b>	Possuir um repositório para a gerência de documentos sobre equipamentos, instalações elétricas, definições de processos, diagramas e outros artefatos necessários ao projeto para o gerenciamento de recursos	2
<b>PP2.5.1</b>	Possuir um inventário de habilidades necessárias ao projeto, para identificar as habilidades que estão faltando para completar o projeto e auxiliar na contratação de pessoas ou treinamentos que proporcionarão as habilidades necessárias para o projeto	5
<b>PP2.6.1</b>	Possuir uma lista contendo os <i>Stakeholders</i> relevantes e seu envolvimento no projeto	2
<b>PP2.7.1</b>	Produzir um documento que contenha todos os artefatos do projeto que sejam necessários para formar o plano de projeto em nível macro	8
<b>PP3.2.1</b>	Possibilitar o gerenciamento do <i>Product Backlog</i> , fornecendo uma forma simples de remoção e adição de novas <i>User Stories</i>	5
<b>PP3.3.1</b>	Enviar uma notificação solicitando a aprovação do plano desenvolvido durante a <i>Planning Meeting</i> de uma <i>Sprint</i>	3
<b>PMC1.1.1</b>	Proporcionar a inserção de horas realizadas e atualização das horas restantes nas tarefas de uma <i>User Story</i> e atualizar o <i>Iteration Burndown chart</i> automaticamente após cada mudança	13
<b>PMC1.2.1</b>	Possuir uma funcionalidade para que, durante a <i>Daily Scrum</i> , o <i>Scrum Master</i> insira, para cada membro do time, o que aquele membro fez desde a última <i>Daily Scrum</i> , o que ele planeja fazer até a próxima <i>Daily Scrum</i> e quais são os seus impedimentos. Estes impedimentos são atualizados na lista de <i>Impediments</i>	5
<b>PMC1.3.1</b>	Guardar um histórico de riscos para mostrar a quantidade de riscos encontrados no início do projeto, a quantidade de riscos fechados, os riscos que foram adicionados durante o projeto e a quantidade de riscos abertos	8
<b>PMC1.4.1</b>	Proporcionar o versionamento do plano de gerenciamento de dados com as revisões de alteração e visualização feitas pelo <i>Scrum Master</i>	2
<b>PMC1.5.1</b>	Mostrar uma lista das <i>Daily Scrums</i> feitas no decorrer do projeto para que isto possa ser utilizado como evidência de registro do envolvimento dos <i>Stakeholders</i>	2

<b>PMC1.6.1</b>	Possuir um repositório para que sejam colocados documentos criados durante a reunião de Review, como uma ata ou um vídeo com a demonstração do incremento funcional	2
<b>PMC1.7.1</b>	Mostrar a quantidade de <i>User Stories</i> planejadas, a quantidade das que foram entregues e a quantidade de novas <i>User Stories</i> identificadas ao final de cada <i>Sprint</i> para que isto sirva como evidência	3
<b>PMC2.1.1</b>	Proporcionar o cadastro de itens dentro do <i>Impediments Backlog</i> , associando o membro do time bloqueado (quando houver), prioridade e membro que deve tomar alguma ação para corrigir o mesmo	3
<b>PMC2.2.1</b>	Permitir a inserção de uma data estimada para correção e ações corretivas a serem tomadas dentro de cada impedimento no <i>Impediments Backlog</i>	2
<b>PMC2.3.1</b>	Armazenar todos os impedimentos já resolvidos no <i>Impediment Backlog</i>	1
<b>PP1.1.2</b>	Facilitar a priorização das <i>User Stories</i> dentro do <i>Product Backlog</i>	3
<b>PP1.4.2</b>	Gerar alertas quando o <i>Sprint Backlog</i> estiver alocando mais ou menos pontos que a velocidade da equipe.	5
<b>PP2.1.2</b>	Permitir a alocação de <i>User Stories</i> em uma determinada <i>Sprint</i>	2
<b>PP2.3.2</b>	Habilitar o gerenciamento do <i>Product Backlog</i> apenas pelo o <i>Product Owner</i> e a visualização do mesmo para todo o time	2
<b>PP2.6.2</b>	Permitir a criação e atualização de usuários no sistema que represente os <i>Stakeholders</i>	2
<b>PP2.7.2</b>	Produzir um documento que contenha todos os artefatos do projeto que sejam necessários para formar o plano de projeto em nível de iteração	5
<b>PP3.1.1</b>	Permitir a visualização de uma determinada versão de um plano	2
<b>PMC1.1.2</b>	Gerar um <i>Burndown chart</i> de custos baseado no orçamento estimado e no número de iterações do projeto	5
<b>PMC1.3.2</b>	Enviar os riscos por e-mail semanalmente para o <i>Product Owner</i> , o <i>Scrum Master</i> e o time	2
<b>PMC1.5.2</b>	Possuir um repositório para que sejam colocados documentos criados durante a reunião de Review, como uma ata ou um vídeo mostrando a demonstração do incremento funcional	2
<b>PMC1.6.2</b>	Enviar o <i>Iteration Burndown chart</i> periodicamente por e-mail para	2

	o <i>Product Owner</i> , solicitando confirmação de leitura	
<b>PMC1.7.2</b>	Possuir uma funcionalidade para colocar as <i>User Stories</i> que não foram completadas na <i>Sprint</i> como incompletas no <i>Product Backlog</i> , para que as estimativas do projeto sejam revistas	2
<b>PMC2.1.2</b>	Habilitar uma funcionalidade que auxilie na criação de impedimentos durante a <i>Daily Scrum</i> , associando automaticamente o membro do time que está bloqueado por este impedimento	3
<b>PMC2.3.2</b>	Ao fechar um impedimento que havia sido reaberto, obrigar a especificação de uma nova ação corretiva	5
<b>PP1.1.3</b>	Realizar o versionamento do <i>Product Backlog</i> para que as informações nele contidas desde o início do projeto estejam disponíveis durante todo seu ciclo de vida	8
<b>PP2.1.3</b>	Permitir a criação de <i>Sprints</i> , com duração e datas de início e fim	3
<b>PP2.3.3</b>	Proporcionar a visualização de uma tabela contendo os <i>Stakeholders</i> do projeto, os artefatos disponíveis para o mesmo, seus níveis de acesso, além de como e onde estes dados estarão disponíveis	3
<b>PP2.6.3</b>	Permitir a criação e atualização de funções, para serem ligadas aos <i>Stakeholders</i> . Estas funções poderiam possuir um texto ou documento anexo informando as regras e responsabilidades daquela função	3
<b>PP3.1.2</b>	Permitir a visualização das versões dos planos que afetam o projeto, com a data e descrição de cada mudança	5
<b>PMC1.1.3</b>	Atualizar o Release <i>Burndown chart</i> automaticamente quando uma <i>User Story</i> for dada como pronta	8
<b>PMC1.3.3</b>	Enviar alertas sobre riscos que tiveram sua data de expiração atingida e não foram solucionados	5
<b>PMC1.6.3</b>	Enviar o Release <i>Burndown chart</i> por e-mail ao final de cada <i>Sprint</i> para os <i>Stakeholders</i> , solicitando confirmação de leitura	2
<b>PP1.1.4</b>	Disponibilizar o <i>Product Backlog</i> de maneira fácil a todos os envolvidos no projeto para que todos possam acessá-lo, visualizá-lo e saber quais os itens do <i>Product Backlog</i> possuem maior prioridade e precisam ser trabalhados primeiro	2
<b>PP2.1.4</b>	Permitir a inclusão dos orçamentos macro do projeto e do orçamento por iteração	3

<b>PP2.3.4</b>	Possibilitar a criação de regras de acesso para os artefatos do projeto	2
<b>PMC1.1.4</b>	Permitir a inserção do orçamento realizado em uma <i>Sprint</i> e atualizar o custo real da <i>Sprint</i> automaticamente no <i>Burndown chart</i> de custos	8
<b>PP2.1.5</b>	Gerar uma sugestão de <i>Release Plan</i> , a partir das <i>User Stories</i> já devidamente pontuadas e priorizadas e da velocidade do time, permitir alterações neste plano de forma simples e facilitar a visualização das <i>User Stories</i> que não estão vinculadas a uma <i>Sprint</i>	13
<b>Total</b>		202

**Tabela 4: Product Backlog**

### 3.4.2.3. Release Plan

O *Release Plan*, foi criado com as seguintes informações:

Pontos Totais	Velocidade	Número de Sprints
202	/ 35	= 6

**Figura 10: Informações utilizadas no Release Plan**

A tabela 10 mostra o *Release Plan* inicial do projeto, contendo as *User Stories* separadas nas 6 *Sprints* planejadas.

<i>Sprint 1</i>	<i>Sprint 2</i>	<i>Sprint 3</i>	<i>Sprint 4</i>	<i>Sprint 5</i>	<i>Sprint 6</i>
PP1.1.1	PP2.5.1	PMC1.2.1	PP2.1.2	PP1.1.3	PMC1.6.3
PP1.2.1	PP2.6.1	PMC1.3.1	PP2.3.2	PP2.1.3	PP1.1.4
PP1.3.1	PP2.7.1	PMC1.4.1	PP2.6.2	PP2.3.3	PP2.1.4
PP1.4.1	PP3.2.1	PMC1.5.1	PP2.7.2	PP2.6.3	PP2.3.4
PP2.1.1	PP3.3.1	PMC1.6.1	PP3.1.1	PP3.1.2	PMC1.1.4
PP2.2.1	PMC1.1.1	PMC1.7.1	PMC1.1.2	PMC1.1.3	PP2.1.5
PP2.3.1		PMC2.1.1	PMC1.3.2	PMC1.3.3	
PP2.4.1		PMC2.2.1	PMC1.5.2		
		PMC2.3.1	PMC1.6.2		
		PP1.1.2	PMC1.7.2		
		PP1.4.2	PMC2.1.2		
			PMC2.3.2		

Tabela 5: *Release Plan*

#### 3.4.2.4. Release Burndown Chart

O Release Burndown Chart inicial do projeto contém apenas os pontos estimados e pode ser visualizado na figura 11.

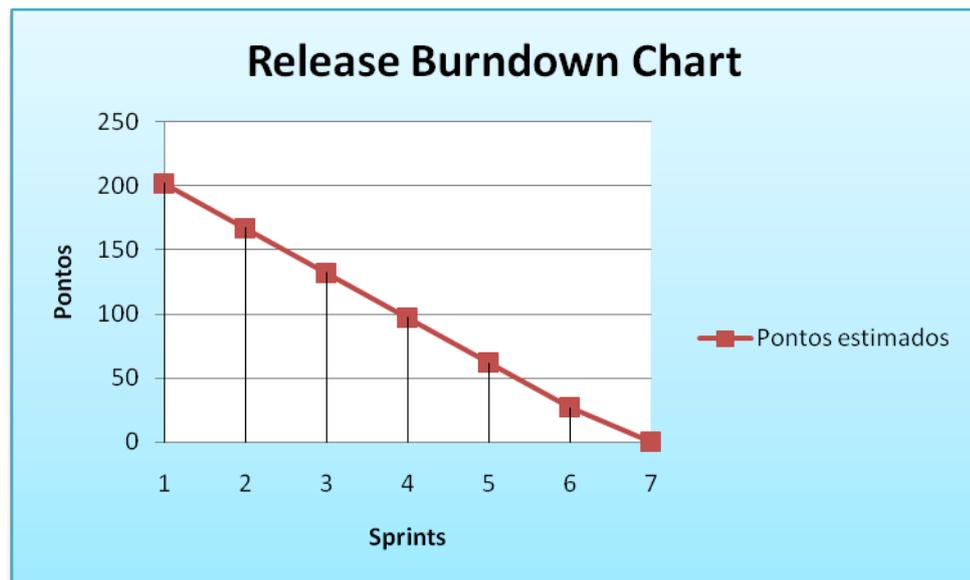


Figura 11: Release Burndown Chart inicial

#### 3.4.2.5. Risk Backlog

O Risk Backlog inicial do projeto pode ser visualizado na tabela 6.

Os índices de probabilidade e impacto utilizados foram de números de 1 a 5 onde o número 1 indica a menor probabilidade/impacto e o número 5 indica a maior probabilidade/impacto.

Descrição			Então	Ação	Responsável
Os pontos de uma <i>User Story</i> não terem sido estimados corretamente.	3	4	Mitigar	Rever a pontuação das <i>User Stories</i> planejadas para a próxima <i>Sprint</i> na <i>Planning Meeting</i> e ajustar o <i>Product Backlog</i> e o <i>Release Plan</i> com as estimativas corretas.	Leonardo e Fernanda.
Não ter sido identificada uma dependência de uma <i>User Story</i> prioritária com uma com prioridade mais baixa.	2	1	Mitigar	Rever se as <i>User Stories</i> da próxima <i>Sprint</i> possuem alguma dependência e ajustar o <i>Product Backlog</i> e o <i>Release Plan</i> com nova priorização.	Leonardo e Fernanda.
O time não possuir a velocidade estimada.	4	5	Mitigar	Rever a velocidade do time no final de cada <i>Sprint</i> e atualizar o <i>Product Backlog</i> e o <i>Release Plan</i> para refletir a nova realidade	Leonardo e Fernanda.
O time conseguir atingir uma velocidade superior à esperada.	1	3	Aceitar	Rever os itens do <i>Product Backlog</i> , procurando a inserir mais <i>User Stories</i> .	Leonardo e Fernanda.
Não ser possível contemplar uma <i>User Story</i> com a tecnologia utilizada.	2	5	Mitigar	Modificar a <i>User Story</i> para que esta possa ser contemplada pela tecnologia.	Leonardo e Fernanda.

Identificar que <i>Sprints</i> de 2 semanas não são adequadas	1	2	Mitigar	Aumentar o tempo das próximas <i>Sprints</i> , atualizando os artefatos impactados	Leonardo e Fernanda.
---	---	---	---------	--	----------------------

**Tabela 6: Risk Backlog inicial**

### 3.4.2.6. Microsoft Office Sharepoint Server 2007

O Microsoft Office Sharepoint Server 2007 (MOSS) é uma ferramenta corporativa, composta de diversos componentes de software, que tem por objetivo fornecer gerenciamento de conteúdo empresarial facilitando a colaboração entre pessoas e times através de um ambiente web.

O MOSS é composto de diversos componentes que podem ser agrupados em seis grandes áreas, representadas na figura 12.



**Figura 12: Áreas de Funcionalidades**

- Colaboração: funcionalidades que permitem que o trabalho entre times se torne efetivo, provendo mecanismos intuitivos, flexíveis e seguros para o compartilhamento de informações através de wikis, blogs, publicação de documentos, formulários de pesquisa, desenvolvimento de workflows e customização de sites.

- Portal: funcionalidades que permitem a personalização e customização dos sites, para que os conteúdos possam ser direcionados aos usuários, grupos ou perfis corretos e para facilitar automaticamente a navegação entre os sites.
- Pesquisa: funcionalidades que permitem localizar facilmente conteúdo relevante distribuído entre uma grande variedade de sites, documentos, dados de outras aplicações e outros recursos.
- Gerenciamento de Conteúdo Empresarial: funcionalidades que permitem a criação, publicação e gerenciamento de informações empresariais.
- Processos e formulários comerciais: funcionalidades que permitem a implementação de formulários que representam processos de negócios, desde sua estruturação até sua publicação.
- Inteligência Comercial (Business Intelligence): funcionalidades que permitem oferecer informações críticas aos objetivos de negócio através de diferentes mecanismos, como análises sofisticadas sobre os dados do sharepoint ou de fontes externas, buscando por padrões pré-determinados.

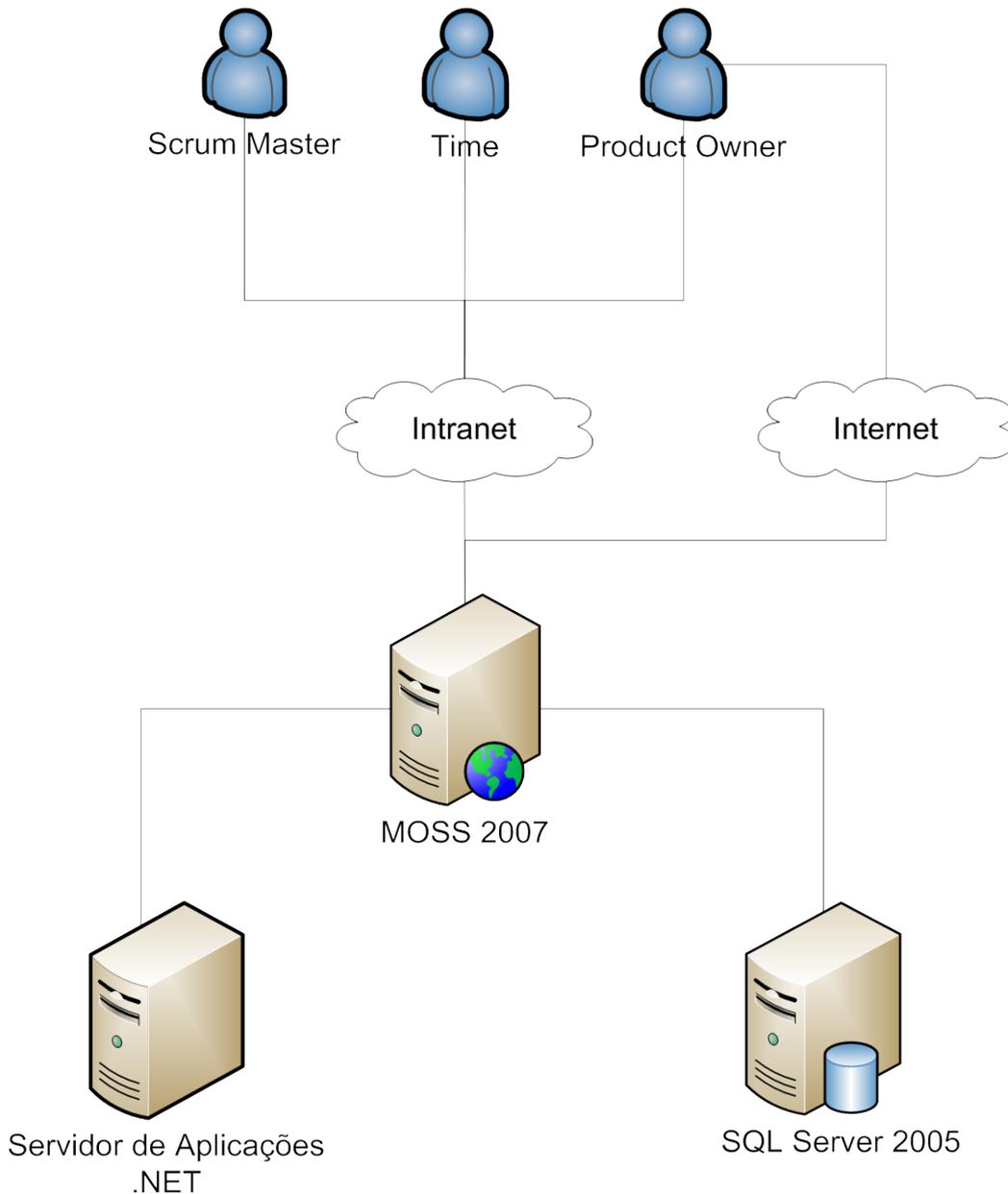
O Sharepoint é completamente integrado com outras ferramentas da plataforma Office, como o Word, para a criação de documentos variados, Excel, para a criação de planilhas e o InfoPath para a criação de formulários customizados. Além disto, o sharepoint fornece mecanismos simples para criação de sites, bibliotecas de documentos, imagens e formulários além de listas personalizadas.

#### 3.4.2.7. Arquitetura do Sistema

O sistema proposto fornece um ambiente web colaborativo onde os usuários podem usufruir de diversas funcionalidades que centralizam as informações do projeto, facilitando seu planejamento, monitoramento e controle.

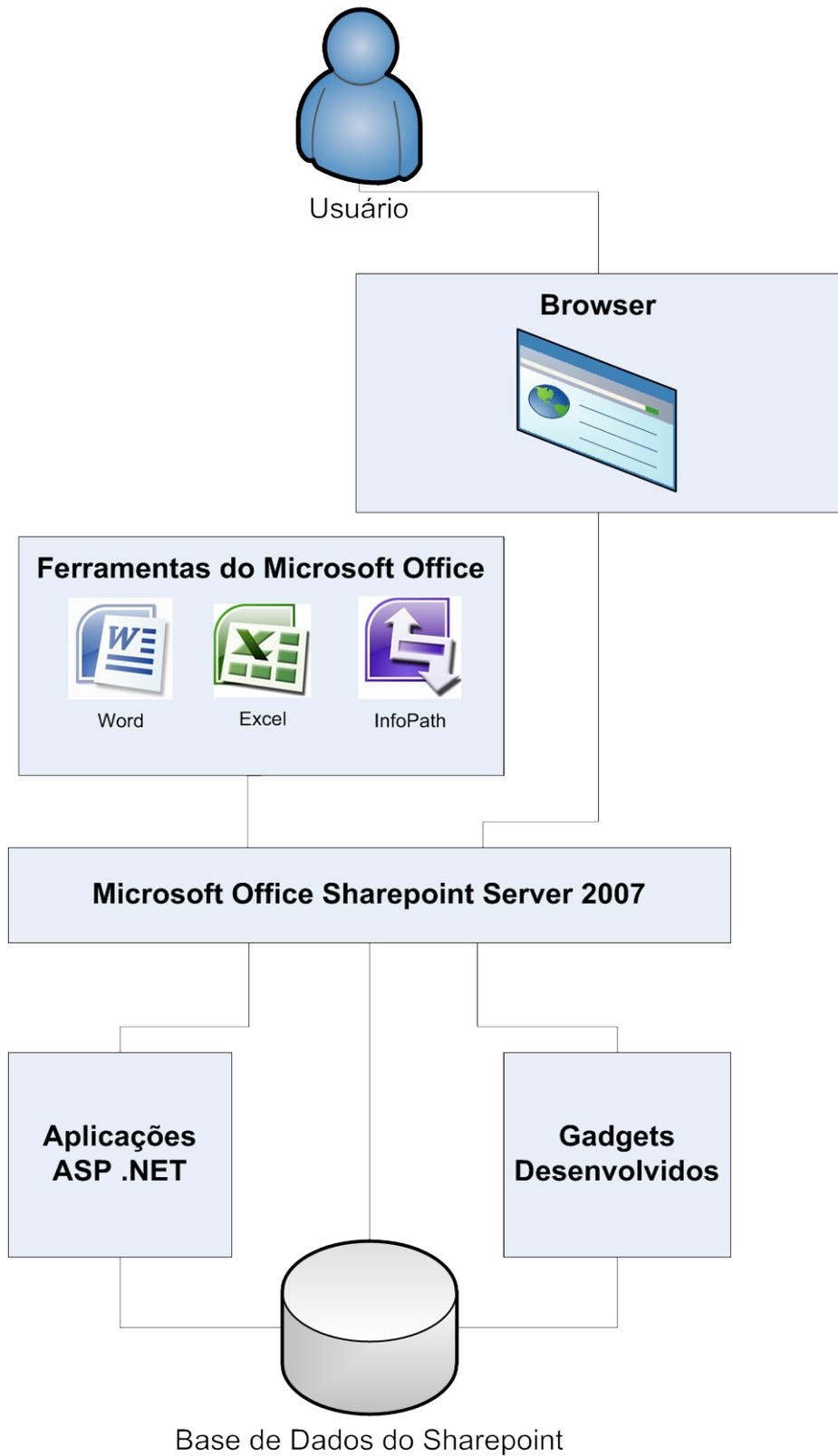
O sistema foi desenvolvido utilizando o Microsoft Office Sharepoint Server 2007 (MOSS) para implementação do ambiente colaborativo e a plataforma .NET para desenvolvimento de aplicações para o sistema.

A arquitetura física do sistema, ilustrada na figura 12, é cliente/servidor, onde o acesso ao mesmo pode ser feito tanto em uma intranet, para o time e o *Scrum Master* quanto através da internet, nos casos em que o *Product Owner* não fique na empresa onde o produto esteja sendo desenvolvido.



**Figura 13: Arquitetura física do sistema**

A arquitetura lógica do sistema está representada na figura 14. Ela mostra a integração entre os módulos que compõe a solução, integrando aplicações ASP .NET e desenvolvimento de novos gadgets com MOSS e as ferramentas do Microsoft Office.



**Figura 14: Arquitetura Lógica do Sistema**

As User Stories do projeto foram desenvolvidas utilizando as funcionalidades de customização dos objetos fornecidos pelo Sharepoint, como web parts, sites e

listas. Nos casos em que o Sharepoint não atendia plenamente o desenvolvimento de alguma User Story ou esta foi modificada para que sua implementação fosse possível ou então foram desenvolvidas aplicações à parte, que possuísse integração com sharepoint, como aplicações ASP .NET ou web parts novas.

#### 3.4.2.8. Recursos utilizados

##### 3.4.2.8.1. Software

- Windows Server 2003 como sistema operacional do servidor;
- Windows XP ou Vista como sistema operacional do cliente e ambiente de desenvolvimento;
- Microsoft Office Sharepoint Server (MOSS) 2007 para a criação do portal;
- Sharepoint Designer 2007 para a customização do design do portal;
- Visual Studio 2008 para o desenvolvimento de novas funcionalidades a serem agregadas ao portal;
- Microsoft Office como ferramenta de apoio à elaboração dos documentos;
- Software No-IP como software utilitário para acesso externo à rede onde se encontra o servidor;

##### 3.4.2.8.2. Hardware

- Uma máquina servidora para hospedagem do portal e que atenda os requisitos mínimos para instalação do MOSS;
- Duas máquinas para o desenvolvimento das funcionalidades e para a criação dos documentos necessários;
- Um roteador wireless para acesso ao servidor tanto pela rede interna quanto pela internet;
- Um modem para acesso pela rede externa;

### 3.4.3. Fase de preparação

Na fase de preparação, os recursos físicos foram adquiridos, os softwares foram instalados e as máquinas configuradas.

Além disto, o time estudou instalação, configuração e gerenciamento do Microsoft Office Sharepoint 2007 além do desenvolvimento de aplicações .NET.

### 3.4.4. Fase de desenvolvimento

A fase de desenvolvimento foi dividida em 5 Sprints de duas semanas cada. Inicialmente, foram planejadas 6 sprints para o trabalho, porém, durante o desenvolvimento, houve uma Sprint que, por indisponibilidade do time, o mesmo não conseguiu desenvolver nenhuma funcionalidade. Esta *Sprint* foi então postergada em 2 semanas e foi acordado que a última *Sprint* do projeto seria removida, já que a fase de entrega não poderia ser postergada.

#### 3.4.4.1. Sprint 1

##### 3.4.4.1.1. Planning Meeting

Durante o planejamento da Sprint 1, detectou-se que algumas User Stories com prioridade mais baixa eram pré-requisitos para o desenvolvimento de outras User Stories com maior prioridade. Assim, o time precisou implementar as User Stories que eram pré-requisito.

Para respeitar a velocidade do time inicialmente planejada, as User Stories menos prioritárias da Sprint foram removidas do Sprint Backlog e recolocadas no Product Backlog, com a maior prioridade.

A figura 15 ilustra o Sprint Backlog da Sprint 1. Como, durante as primeiras *Sprints*, a ferramenta não possuía funcionalidades para gerenciamento de tarefas das *User Stories*, os primeiros *Sprint Backlogs* foram realizados em um documento do Microsoft Project 2007.

A lista de *User Stories* está ordenada por prioridade.

	Task Name	Work	Predecessors	Resource Names
1	☐ <b>Sprint 1</b>	<b>40 hrs</b>		
2	☐ <b>PP 1.1.1</b>	<b>7 hrs</b>		
3	Criação de uma biblioteca para gerenciamento de User Stories	1 hr		Fernanda
4	CRUD das User Stories	2 hrs	3	Fernanda
5	Modificação do design do formulário para que este se pareça com um post it	4 hrs	4	Fernanda
6	☐ <b>PP 1.2.1</b>	<b>2 hrs</b>	<b>2</b>	
7	Visualização das User Stories como lista para simular o Product Backlog	1 hr		Fernanda
8	Adição de um campo na User Stories para armazenar sua pontuação	1 hr		Fernanda
9	☐ <b>PP 1.3.1</b>	<b>5 hrs</b>	<b>6</b>	
10	Levantar quais atividades necessitam serem feitas para o cumprimento do ciclo de	3 hrs		Fernanda
11	Criar uma lista que tenha as atividades levantadas	2 hrs	10	Fernanda
12	☐ <b>PP 2.1.3*</b>	<b>2 hrs</b>		
13	Criar uma lista de Sprints para um projeto	1 hr		Leonardo
14	CRUD Sprints	1 hr	13	Leonardo
15	☐ <b>PP 2.1.2*</b>	<b>4 hrs</b>	<b>12</b>	
16	Permitir a alocação de Stories em uma determinada Sprint	4 hrs		Leonardo
17	☐ <b>PMC 1.7.2*</b>	<b>2 hrs</b>	<b>15</b>	
18	Criar uma forma de marcar uma user story de uma sprint como completa	2 hrs		Leonardo
19	☐ <b>PP 1.4.1</b>	<b>9 hrs</b>	<b>17</b>	
20	Acessar os dados das users stories de uma sprint	6 hrs		Leonardo
21	Verificar quais user stories de uma sprint passada foram finalizadas	1 hr		Leonardo
22	Calcular a velocidade do time a partir das user stories completadas em uma sprint	1 hr		Leonardo
23	Mostrar a velocidade atual do time	1 hr		Leonardo
24	☐ <b>PP 2.6.2*</b>	<b>2 hrs</b>	<b>9</b>	
25	Criação dos grupos: time, scrum master e product owner	1 hr		Fernanda
26	Permitir o vínculo de usuários nestes grupos	0,5 hrs		Fernanda
27	Permitir a criação de novos grupos	0,5 hrs		Fernanda
28	☐ <b>PP 2.1.1</b>	<b>7 hrs</b>	<b>24</b>	
29	CRUD de tarefas	1 hr		Fernanda
30	Priorização de tarefas de uma User Story	2 hrs		Fernanda
31	Permitir dependência entre tarefas	4 hrs		Fernanda

\* User Stories não planejadas originalmente para esta sprint

**Figura 15: Sprint 1 Backlog**

#### 3.4.4.1.2. Review Meeting

De todas as User Stories planejadas para a Sprint, apenas a User Story PP 2.1.3 foi finalizada.

A velocidade do time nesta Sprint foi de: 3 pontos.

A falta de conhecimento do funcionamento das listas do Sharepoint impactou diretamente neste resultado. O time possuía muitas incertezas em relação à implementação e teve que estudar e procurar outros recursos para entendimento do desenvolvimento das funcionalidades no Sharepoint. Assim, as User Stories não finalizadas nesta sprint foram colocadas novamente no product backlog.

Foi concordado que a User Story PP 1.1.1, que solicitava um template para criação de User Stories, irá ser dividida em duas: a primeira (PP 1.1.1) será apenas

o template em forma de lista do Sharepoint e a segunda (TC1.1) seria a modificação dos itens desta lista para que as User Stories apareçam como post it.

Além da TC1.1, duas outras User Stories foram identificadas e adicionadas no Product Backlog:

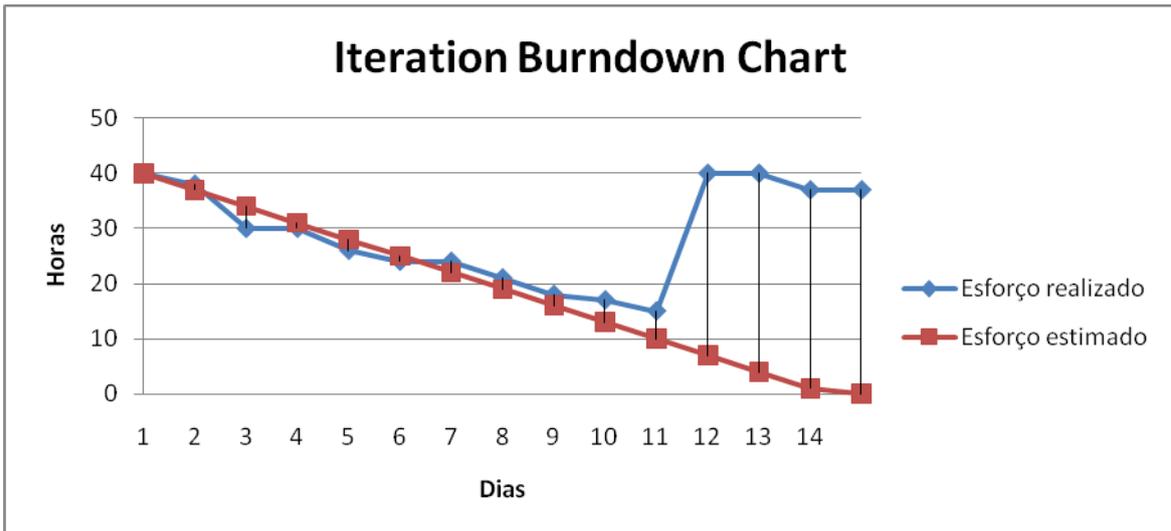
- TC1.2: configurar log de eventos para listas do Sharepoint
- TC1.3: modificar a apresentação do portal, para que este se torne mais prático e intuitivo para seus usuários

O Risk Backlog foi revisado e foi concordado aumentar a probabilidade do risco que relata a não identificação de dependências entre *User Stories*, já que ele aconteceu nesta Sprint e o plano de mitigação foi tomado conforme planejado. Além disto, um novo risco foi identificado e este é ilustrado na tabela 7. Como a *User Story* de criação de um *Risk Backlog* foi criada apenas na Sprint 3, as modificações feitas durante as Sprints 1 e 2 no Risk Backlog, foram feitas utilizando o programa Microsoft Excel 2007.

Descrição			Estratégia de	Ação	Responsável
Pouco conhecimento da tecnologia implementação das próximas User Stories.	4	5	Mitigar	Rever o product backlog para possível remoção de escopo.	Leonardo e Fernanda

**Tabela 7: Novo risco identificado na Sprint 1**

Como, durante a Sprint 1, a ferramenta não possuía funcionalidades para a visualização de um Iteration Burndown Chart, o mesmo foi feito utilizando o programa Microsoft Excel 2007. O mesmo pode ser visualizado na figura 16.



**Figura 16: Iteration Burndown Chart da Sprint 1**

Pode-se notar, com a análise deste gráfico, que o time estava tendo um bom rendimento na sprint, porém foi percebido que as últimas *User Stories* da iteração não poderiam ser desenvolvidas se não modificássemos a forma de implementação das *User Stories* já desenvolvidas durante a Sprint. Assim, as *User Stories* já desenvolvidas foram reabertas e o esforço realizado voltou ao início.

#### 3.4.4.1.3. Retrospective

*O que o time fez certo durante a Sprint?*

O time conseguiu seguir o processo definido.

*O que o time deveria melhorar nas próximas Sprints?*

O time imaginava que as *User Stories* iniciais poderiam ser feitas utilizando as listas do Sharepoint, porém, durante a sprint, descobriu-se que algumas *User Stories* não poderiam ser contempladas desta maneira.

As *User Stories* que haviam sido já iniciadas foram reiniciadas após esta descoberta.

O time estudou melhor a tecnologia para conseguir terminar as *User Stories* da Sprint 2 sem os mesmos problemas encontrados durante a Sprint 1.

O time percebeu que muitas das User Stories deveriam ter uma tarefa para integração. Isto foi adicionado no Sprint Backlog das demais Sprints, sempre que aplicável.

### 3.4.4.2. Sprint 2

#### 3.4.4.2.1. Planning Meeting

Mesmo tendo feito apenas 3 pontos durante a Sprint 1. O time acreditava que na Sprint 2 não teria tantos problemas quanto na Sprint 1 e decidiu não utilizar apenas 3 pontos como parâmetro de velocidade da Sprint 2. Mesmo assim, 35 pontos pareciam acima da velocidade que o time conseguiria atingir e o mesmo decidiu colocar a velocidade estimada para a Sprint de 25 pontos.

As User Stories mais prioritárias foram selecionadas para serem desenvolvidas na Sprint. A figura 17 ilustra o Sprint Backlog da Sprint 2. Ela também foi feita utilizando o Microsoft Project 2007.

Task Name	Work	Predecessors	Resource Names
<input type="checkbox"/> Sprint 2	<b>46 hrs</b>		
<input type="checkbox"/> PP 1.1.1	<b>8 hrs</b>		
Criação de uma biblioteca para gerenciamento de User Stories	2 hrs		Leonardo
CRUD das User Stories	3 hrs	3	Leonardo
Integração	3 hrs	4	
<input type="checkbox"/> PP 1.2.1	<b>5 hrs</b>	<b>2</b>	
Visualização das User Stories como lista para simular o Product Backlog	1 hr		Leonardo
Adição de um campo na User Stories para armazenar sua pontuação	1 hr		Leonardo
Integração	3 hrs	7;8	
<input type="checkbox"/> PP 1.3.1	<b>9 hrs</b>	<b>6</b>	
Levantar quais atividades necessitam serem feitas para o cumprimento do ciclo de vida do scrum	3 hrs		Fernanda
Criar uma lista que tenha as atividades levantadas	3 hrs	11	Fernanda
Integração	3 hrs	12	
<input type="checkbox"/> PP 2.1.2	<b>7 hrs</b>		
Permitir a alocação de Stories em uma determinada Sprint	4 hrs		Leonardo
Integração	3 hrs	15	
<input type="checkbox"/> PMC 1.7.2	<b>5 hrs</b>	<b>14</b>	
Criar uma forma de marcar uma user story de uma sprint como completa	2 hrs		Leonardo
Integração	3 hrs	18	
<input type="checkbox"/> PP 1.4.1	<b>12 hrs</b>	<b>17</b>	
Acessar os dados das users stories de uma sprint	6 hrs		Leonardo
Verificar quais user stories de uma sprint passada foram finalizadas	1 hr	21	Leonardo
Calcular a velocidade do time a partir das user stories completadas em uma sprint	1 hr	22	Leonardo
Mostrar a velocidade atual do time	1 hr	23	Leonardo
Integração	3 hrs	24	

Figura 17: Sprint 2 Backlog

#### 3.4.4.2.2. Review Meeting

Todas as User Stories planejadas para a Sprint foram finalizadas.

A velocidade do time nesta Sprint foi de: 24 pontos.

Para acessar os dados das sprints e gerar automaticamente a base histórica (PP 1.4.1), foi necessário utilizar o Visual Studio. Por este motivo, a velocidade aparece com uma interface simples na tela. Foi criada uma *User Story* e adicionada ao Product Backlog para melhorar a interface gráfica desta User Story.

Ao criar as User Stories no Product backlog, verificou-se a necessidade de uma melhoria no produto para que seja mostrada uma dica de prioridade a ser utilizada com base na maior prioridade do Product Backlog, para que o usuário que esteja cadastrando as *User Stories* na ordem de prioridade não precise saber a última prioridade colocada.

Outra melhoria proposta e adicionada ao final do Product Backlog foi do produto não aceitar prioridades iguais para duas User Stories.

Não houve atualização do Risk Backlog.

A figura 18 mostra o Iteration Burndown Chart da Sprint 2. O mesmo foi feito utilizando o programa Microsoft Excel 2007.

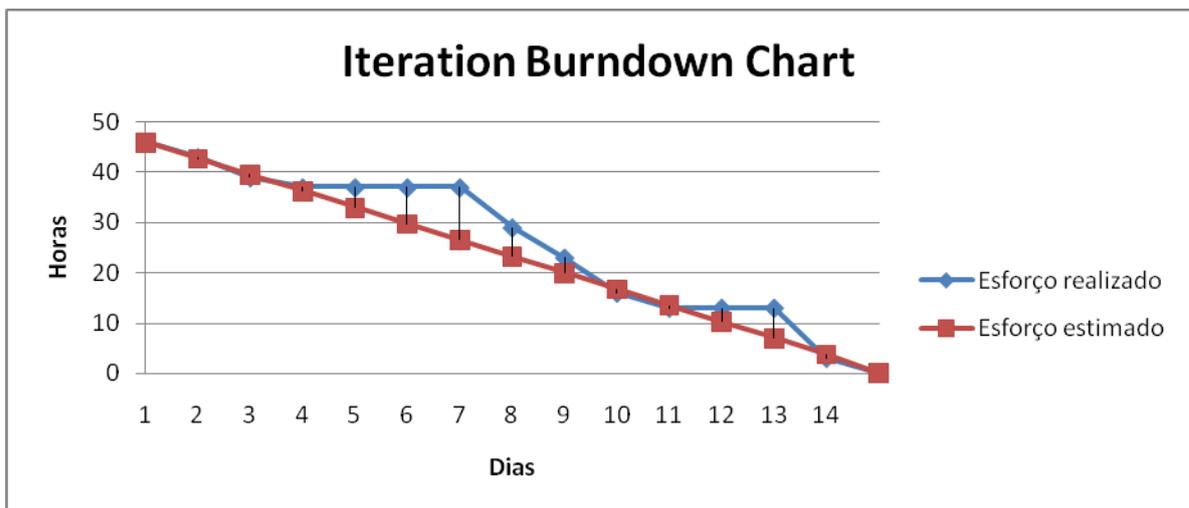


Figura 18: *Iteration Burndown Chart da Sprint 2*

#### 3.4.4.2.3. Retrospective

*O que o time fez certo durante a Sprint?*

O time verificou que era possível acessar a base de dados do sharepoint e voltou a fazer as User Stories sem utilizar um banco de dados externo ao sharepoint.

As User Stories que haviam sido prontas foram revistas para não acessar outros bancos de dados. Isto foi feito de forma rápida e não houve impacto no projeto. Na verdade, as User Stories depois destas ficaram mais rápidas de serem feitas.

A organização desta Sprint foi melhor do que a anterior.

Utilizar apenas o banco de dados do sharepoint foi muito mais vantajoso do que criar um próprio e o processo fluiu mais naturalmente.

*O que o time deveria melhorar nas próximas Sprints?*

O time não encontrou nenhum ponto a melhorar.

#### 3.4.4.3. Sprint 3

##### 3.4.4.3.1. Planning Meeting

Devido alguns problemas enfrentados pelo time, a Sprint 3 foi postergada por duas semanas. Todas as Sprints depois desta também foram postergadas.

O time havia estimado 25 pontos na Sprint 2 e fez todas as User Stories. Para a Sprint 3, a mesma pontuação foi planejada.

As User Stories mais prioritárias foram selecionadas para serem desenvolvidas na Sprint. A figura 19 ilustra o Sprint Backlog da Sprint 3. Ela também foi feita utilizando o Microsoft Project 2007.

Task Name	Work	Predecessors	Resource Names
<b>[-] Sprint 3</b>	<b>31 hrs</b>		
<b>[-] PP 2.1.1</b>	<b>5 hrs</b>		
Criar um lista no sharepoint para gerenciamento de tarefas	2 hrs		Leonardo[25%]
Fazer links dos campos da lista com campos de outras listas	1 hr	3	Leonardo[25%]
Fazer views das tasks por user story	2 hrs	4	Fernanda[25%]
<b>[-] PP 2.2.1</b>	<b>3 hrs</b>		
Criar uma lista no sharepoit para gerenciamento dos riscos	3 hrs		Fernanda[25%]
<b>[-] PP 2.3.1</b>	<b>5 hrs</b>		
Identificar os artefatos a serem versionado	3 hrs		Leonardo[38%]
Configurar os artefatos para serem versionados	2 hrs	9	Leonardo[25%]
<b>[-] PP 2.4.1</b>	<b>2 hrs</b>		
Criar uma biblioteca de documentos	2 hrs		Fernanda[25%]
<b>[-] PP 2.6.2</b>	<b>3 hrs</b>		
Mapear os stakeholders	1 hr		Fernanda[13%]
Criar um grupo de usuários para cada stakeholder	1 hr	14	Fernanda[13%]
Permitir a criação e edição de usuários para os grups	1 hr	15	Fernanda
<b>[-] PP 2.5.1</b>	<b>2 hrs</b>		
Desenvolver um inventário de habilidades	1 hr		Leonardo[13%]
Preencher o inventário	1 hr		Leonardo[13%]
<b>[-] PP 2.6.1</b>	<b>2 hrs</b>		
Criar uma biblioteca de documentos para os stakeholders	1 hr		Fernanda[13%]
Preencher documento com o mapeamento dos stakeholders	1 hr		Fernanda[13%]
<b>[-] TC 1.3</b>	<b>9 hrs</b>		
Fazer wire frames do portal buscando arquitetar melhor nave	3 hrs		Fernanda[50%]
Modificar o design e arquitetura do portal	6 hrs		Fernanda[75%]

**Figura 19: Sprint 3 Backlog**

#### 3.4.4.3.2. Review Meeting

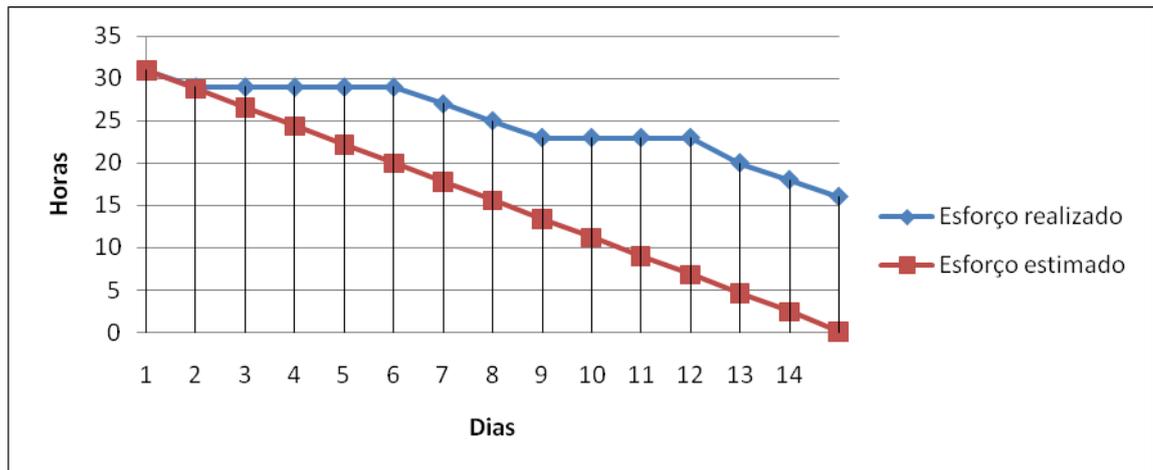
As seguintes User Stories foram finalizadas: PP2.1.1, PP2.2.1, PP2.3.1 e PP.2.4.1. As demais não foram finalizadas.

A velocidade do time nesta Sprint foi de: 11 pontos.

A User Story TC1.3 saiu da sprint, pois o Product Owner verificou que ela não tinha tanta prioridade quanto as demais.

Não houve atualização do Risk Backlog.

A figura 20 mostra o Iteration Burndown Chart da Sprint 3. O mesmo foi feito utilizando o programa Microsoft Excel 2007.



**Figura 20: Iteration Burndown Chart da Sprint 3**

#### 3.4.4.3.3. Retrospective

*O que o time fez certo durante a Sprint?*

Nenhum ponto a considerar.

*O que o time deveria melhorar nas próximas Sprints?*

Devido aos problemas pessoais do time, algumas User Stories da Sprint não foram completas.

O time verificou que precisava fazer mais User Stories nas demais Sprints para conseguir atingir um produto aceitável.

#### 3.4.4.4. Sprint 4

##### 3.4.4.4.1. Planning Meeting

A velocidade atual do time foi atualizada para 11 pontos (mesmo valor em pontos de User Stories finalizadas na Sprint 3).

A quantidade total em pontos de User Stories já finalizadas até o início da Sprint 4 era de 38 pontos.

O time necessitou contornar este problema para tentar finalizar todas as User Stories a tempo. Como ainda tinham 3 Sprints (na Planning Meeting da Sprint 4, o

time ainda não havia decidido remover a Sprint 6), ele tentou fazer 55 pontos por Sprint e decidiu utilizar 55 pontos como a velocidade atual do time.

Um nova User Story foi criada, para criar um repositório para a Planning Meeting, Review Meeting e Retrospective Meeting, pois o time viu que isto não existia no Product Backlog e era importante. Esta User Story foi planejada para ser feita na Sprint 4.

O time verificou que a User Story PMC1.1.1 poderia ser dividida em duas. A mesma foi dividida em uma User Story de 5 pontos de outra de 8 pontos.

- PMC1.1.1 A: Proporcionar a inserção de horas realizadas e atualização das horas restantes nas tarefas de uma User Story.
- PMC1.1.1 B: atualizar o Iteration Burndown chart automaticamente após cada mudança nas tarefas de uma User Story.

O time verificou que a User Story PMC2.1.1 era pré-requisito para a User Story PMC1.2.1. A mesma foi colocada na Sprint 4 como mais prioritária que a PMC 1.2.1.

A tabela 21 ilustra o Sprint Backlog da Sprint 4. Desta vez, foi possível visualizar o Sprint Backlog com o auxílio da ferramenta, pois na Sprint 4 ela já suportava a entrada de tarefas para as *User Stories*.

Union > User Story Tasks

## User Story Tasks

Tasks from a User Story.

New Actions Settings View: **Sprint 4 Backlog**

Edit	Task	Estimated hours	Allocated resource	Dependency
<b>Sprint : Sprint 4 (33)</b>				
<b>User Story : PMC1.1.1 A (1)</b>				
	Update the Tasks list with the new fields <b>NEW</b>	1	JOKER\Leonardo	
<b>User Story : PMC1.1.1 B (8)</b>				
	Investigate asp net charts.	4	JOKER\nanda	
	Create an iteration burndown chart mock using ASP	8	JOKER\nanda	
	Create the real line for estimated hours	4	JOKER\nanda	
	Create the real line for remainin hours per day	4	JOKER\nanda	
	Update the chart to work with different sprints	4	JOKER\nanda	
	Insert the chart on sharepoint	1	JOKER\nanda	
	Create a trigger to get the remaining hours everyday	2	JOKER\nanda	
	Add a button to update the chart	3	JOKER\nanda	
<b>User Story : PMC1.2.1 (1)</b>				
	Create the Daily Scrums list.	1	JOKER\Leonardo	
<b>User Story : PMC1.3.1 (3)</b>				
	Create the column "Creation Date" on "Risk Backlog"	1	JOKER\Leonardo	
	Create the column "Closed" on "Risk Backlog"	1	JOKER\Leonardo	Create the column "Creation Date" on "Risk Backlog"
	Create the Web part	6	JOKER\Leonardo	Create the column "Closed" on "Risk Backlog"
<b>User Story : PMC2.1.1 (2)</b>				
	Create the List	1	JOKER\Leonardo	
	Test the CRUD.	1	JOKER\Leonardo	Create the List
<b>User Story : PP2.5.1 (2)</b>				
	Create the inventory	1	JOKER\Leonardo	
	Insert the necessary capabilities	1	JOKER\Leonardo	Create the inventory
<b>User Story : PP2.6.1 (2)</b>				
	Create a document library for stakeholders <b>NEW</b>	1	JOKER\nanda	
	Map the stakeholders <b>NEW</b>	2	JOKER\nanda	Create a document library for stakeholders <b>NEW</b>
<b>User Story : PP2.6.2 (3)</b>				
	Map all the stakeholders	1	JOKER\nanda	
	Create a user group for each one	1	JOKER\nanda	Map all the stakeholders
	Allow the creation and update of users for any group	1	JOKER\nanda	Create a user group for each one
<b>User Story : PP2.7.1 (3)</b>				
	Verify how to insert sharepoint data on a word doc	4	JOKER\Leonardo	
	Verify what data must go to the doc	2	JOKER\Leonardo	
	Create the doc default	2	JOKER\Leonardo	
<b>User Story : PP3.2.1 (1)</b>				
	Allow the maintain of Product Backlog <b>NEW</b>	1	JOKER\Leonardo	
<b>User Story : PP3.3.1 (1)</b>				
	Create the workflow	4	JOKER\Leonardo	
<b>User Story : TC4.1 (3)</b>				
	Create the documents repository	1	JOKER\Leonardo	
	Fill the documents repository with the docs.	1	JOKER\Leonardo	Create the documents repository
	Create a new document library <b>NEW</b>	1	JOKER\nanda	
<b>User Story : TC4.4 (3)</b>				
	Create template for Planning Meeting	2	JOKER\Leonardo	
	Create the template for Review meeting	2	JOKER\Leonardo	
	Create the template for Retrospective meeting.	2	JOKER\Leonardo	

Figura 21: Sprint 4 Backlog

### 3.4.4.4.2. Review Meeting

As seguintes User Stories não foram finalizadas: PP 2.7.1 e PMC 1.1.1 B. As demais foram finalizadas.

A User Story TC 4.4 foi criada durante a Sprint 4. Como ela tinha alta prioridade, ela foi desenvolvida na própria Sprint.

A velocidade do time nesta Sprint foi de: 44 pontos.

O Risk Backlog foi revisado e os seguintes riscos foram fechados:

- Identificar que Sprints de 2 semanas não são adequadas: O time chegou à conclusão que a duração de 2 semanas estava adequada.
- O time conseguir atingir uma velocidade superior à esperada: O time concluiu que não era capaz de fazer mais pontos que o estimado.

Um novo risco foi identificado e este é ilustrado na figura 22:

The screenshot shows a web interface for a Risk Backlog. At the top, there is a breadcrumb trail: 'Union > Risk Backlog > Unfinished stories'. Below this is a yellow header with the title 'Risk Backlog: Unfinished stories' and a 'Close' button. A blue bar contains actions: 'Delete this version', 'Restore this version', and 'Version History'. The main content is a table with the following data:

<b>Risk</b>	Unfinished stories
<b>Priority</b>	0
<b>Category</b>	None
<b>Description</b>	The team might not be able to complete all planned user stories with the time frame of 2 sprints
<b>Impact</b>	Medium
<b>Probability</b>	Very high
<b>Strategy</b>	Accept
<b>Action</b>	The user stories that can't be finished will be developed in the next product's versin
<b>User responsible</b>	JOKER\scrum.master
<b>Closed</b>	No
<b>Creation date</b>	5/18/2009

At the bottom, it shows 'Version: 1.0' and 'Created at 5/18/2009 9:36 PM by System Account', with another 'Close' button.

**Figura 22: Novo Risco Identificado**

A figura 23 mostra o Iteration Burndown Chart da Sprint 4. O mesmo foi feito utilizando o programa Microsoft Excel 2007.

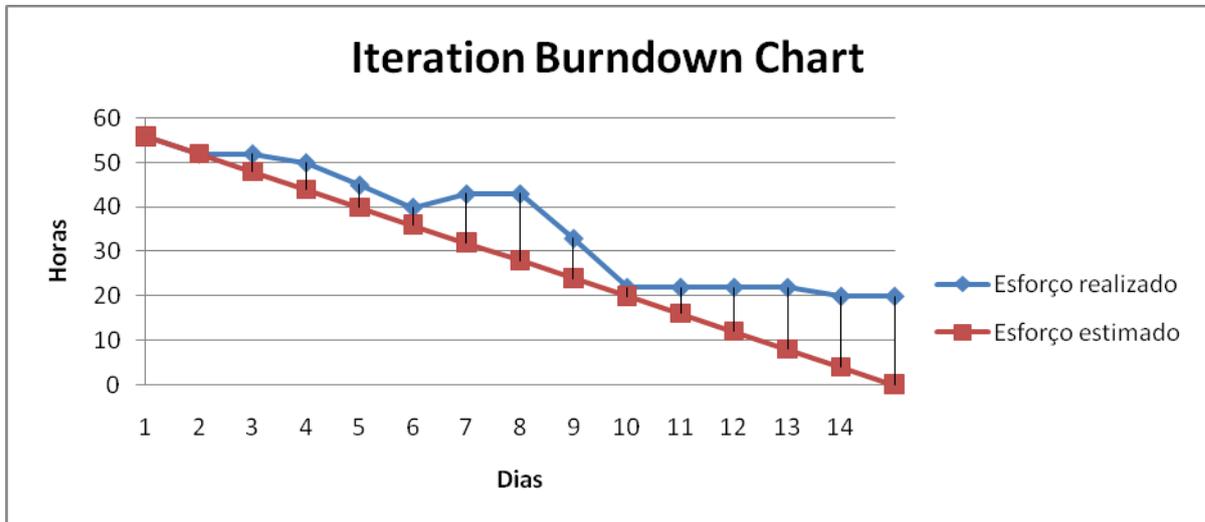


Figura 23: Iteration Burndown chart da Sprint 4

#### 3.4.4.4.3. Retrospective

*O que o time fez certo durante a Sprint?*

O time aumentou consideravelmente a sua velocidade.

*O que o time deveria melhorar nas próximas Sprints?*

O time precisava quebrar as User Stories em tarefas e melhorar a estimativa destas.

#### 3.4.4.5. Sprint 5

##### 3.4.4.5.1. Planning Meeting

A velocidade atual do time foi atualizada para 44 pontos (mesmo valor em pontos de User Stories finalizadas na Sprint 4).

A quantidade total em pontos de User Stories já finalizadas até o início da Sprint 5 era de 82 pontos.

Verificou-se que, com a atual velocidade, para que o time conseguisse finalizar todas as User Stories restantes, seriam necessários mais 4 Sprints. Como o projeto tinha apenas 2 Sprints restantes, as seguintes User Stories foram retiradas desta versão do produto: PP1.1.4, PP2.1.4, PP2.3.4, PMC1.1.4, PP2.1.5, C1.1, TC1.2, TC2.1, TC2.2, TC2.3, TC3.1, TC3.2, TC4.2.

A tabela 24 ilustra o Sprint Backlog da Sprint 5.

# User Story Tasks

Tasks from a User Story.

New Actions Settings View: **Sprint 5 Backlog**

Edit	Task	Estimated hours	Allocated resource	Dependency
<b>Sprint : Sprint 5 (31)</b>				
User Story : Bug Fixing (2)				
	Fix the Team's velocity web part issue	2	JOKER\nanda	
	Fix the workflow issue	3	JOKER\Leonardo	
User Story : PMC1.1.1 B (8)				
	Investigate asp net charts.	4	JOKER\nanda	
	Create an iteration burndown chart mock using ASP	8	JOKER\nanda	
	Create the real line for estimated hours	4	JOKER\nanda	
	Create the real line for remainin hours per day	4	JOKER\nanda	
	Update the chart to work with different sprints	4	JOKER\nanda	
	Insert the chart on sharepoint	1	JOKER\nanda	
	Create a trigger to get the remaining hours everyday	2	JOKER\nanda	
	Add a button to update the chart	3	JOKER\nanda	
User Story : PMC1.1.3 (3)				
	Develop an ASP .net site to provide the graphic	2	JOKER\nanda	
	Retrieve sharepoint data to fill the graphic.	4	JOKER\nanda	
	Deploy the site and integrate with sharepoint	1	JOKER\nanda	
User Story : PMC1.4.1 (2)				
	Create the Data management plan	1	JOKER\Leonardo	
	Allow its versioning	1	JOKER\Leonardo	
User Story : PMC1.5.1 (1)				
	Create a way to show the list	2	JOKER\Leonardo	
User Story : PMC2.2.1 (2)				
	Create the column Estimated date	1	JOKER\Leonardo	
	Create the column actions to be taken	1	JOKER\Leonardo	
User Story : PMC2.3.1 (1)				
	Create the column solved	1	JOKER\Leonardo	
User Story : PP1.4.2 (2)				
	Investigate how to send alerts	4	JOKER\Leonardo	
	Send alerts	2	JOKER\Leonardo	
User Story : PP2.3.2 (2)				
	Allow to only the PO to change the PB	2	JOKER\Leonardo	
	Allow the team to see it	1	JOKER\Leonardo	
User Story : PP2.7.1 (3)				
	Verify how to insert sharepoint data on a word doc	4	JOKER\Leonardo	
	Verify what data must go to the doc	2	JOKER\Leonardo	
	Create the doc default	2	JOKER\Leonardo	
User Story : PP2.7.2 (2)				
	Verify what data must go to the doc	3	JOKER\Leonardo	
	Create the doc default	3	JOKER\Leonardo	
User Story : PP3.1.1 (1)				
	Allow the visualization	2	JOKER\Leonardo	
User Story : TC4.3 (2)				
	Create the statuses	1	JOKER\Leonardo	
	Insert the statuses	1	JOKER\Leonardo	

Figura 24: Sprint 5 Backlog

### 3.4.4.5.2. Review Meeting

A velocidade do time nesta Sprint foi de: 41 pontos.

As seguintes User Stories não foram finalizadas: PP2.7.1, PP1.4.2, PP2.7.2.

Todas as demais foram finalizadas.

Ao final da Sprint 5, o time verificou que era necessário se concentrar no desenvolvimento do volume final e que não seria possível realizar a Sprint 6. Todas as User Stories que não foram finalizadas até o final da Sprint 5 não farão parte desta versão.

O Risk Backlog foi revisado e todos os riscos desta versão que ainda estavam abertos foram fechados.

A figura 25 mostra o Iteration Burndown Chart da Sprint 5. Como a User Story para visualização do Iteration Burndown Chart foi finalizada nesta Sprint, o mesmo será mostrado utilizando a própria tecnologia desenvolvida no projeto.

#### Iteration Burndown chart

Choose:

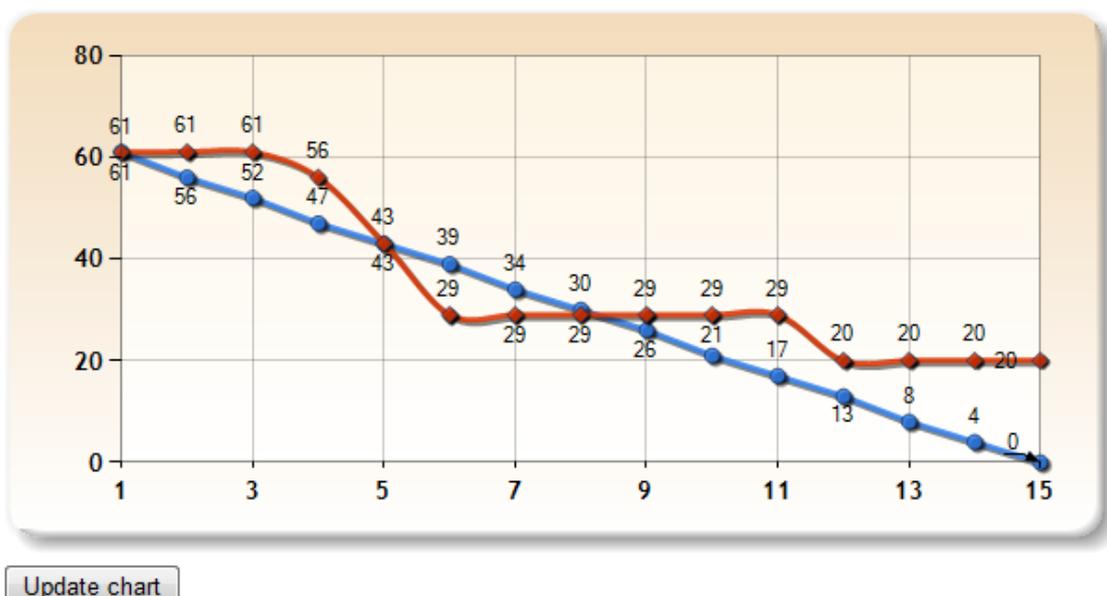


Figura 25: Iteration Burndown Chart da Sprint 5

### 3.4.4.5.3. Retrospective

*O que o time fez certo durante a Sprint?*

O time obteve um bom rendimento na Sprint.

A User Story de Iteration Burndown Chart pôde ser finalizada.

*Comentários:*

Nas próximas versões, é preciso verificar com antecedência se a tecnologia suporta as User Stories, e caso não suporte, deve ser descoberta uma maneira de fazê-la suportar.

### 3.4.4.6. Fase de entrega

Durante a fase de entrega, este volume foi escrito, a ferramenta foi estabilizada e a apresentação do trabalho foi desenvolvida.

### 3.4.4.7. Resumo dos Resultados

A tabela 8 ilustra a relação entre as *User Stories* planejadas e as que realmente foram desenvolvidas durante o projeto.

User Story	Sprint Planejada							Sprint Realizada					
	1	2	3	4	5	6		1	2	3	4	5	NF <sup>7</sup>
PP1.1.1	X								X				
PP1.2.1	X								X				
PP1.3.1	X								X				
PP1.4.1	X								X				
PP2.1.1	X									X			
PP2.2.1	X									X			

---

<sup>7</sup> NF: Não feita.

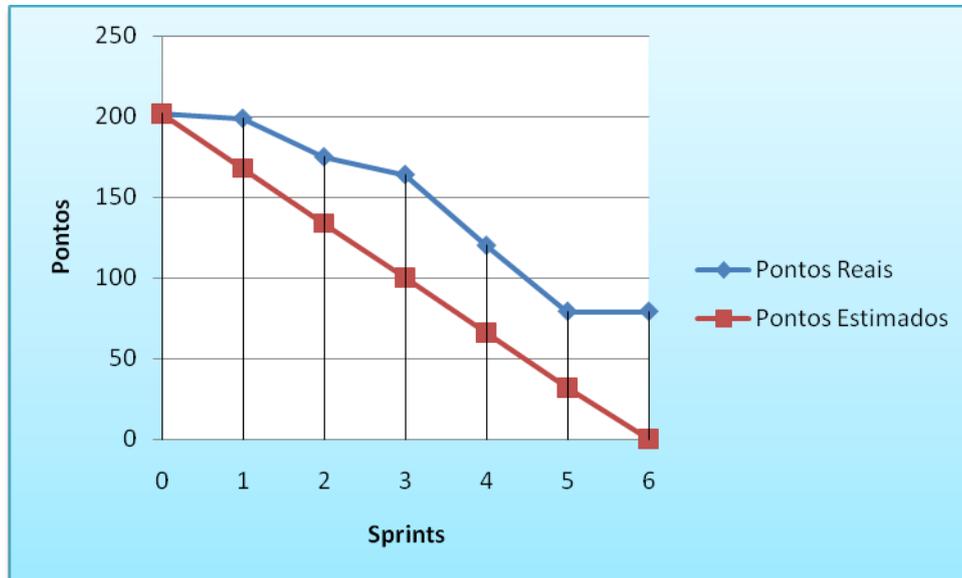


PMC2.3.2				X									X
PP1.1.3					X							X	
PP2.1.3					X		X						
PP2.3.3					X								X
PP2.6.3					X								X
PP3.1.2					X								X
PMC1.1.3					X								X
PMC1.3.3					X								X
	<b>Sprint Planejada</b>							<b>Sprint Realizada</b>					
	1	2	3	4	5	6		1	2	3	4	5	NF
PMC1.6.3						X							X
PP1.1.4						X						X	
PP2.1.4						X							X
PP2.3.4						X							X
PMC1.1.4						X							X
PP2.1.5						X							X
TC1.1													X
TC1.2													X
TC1.3													X
TC2.1													X
TC3.1												X	
TC4.1										X			
TC4.2													X
TC4.3												X	
TC4.4										X			

Tabela 8: Relação entre *User Stories* planejadas e *User Stories* realizadas

Durante a fase de planejamento, a priorização das *User Stories* foi feita buscando com que a ferramenta tivesse ao menos uma funcionalidade de cada área de processo. É possível notar na tabela 8, que apenas duas áreas de processo não tiveram suas funcionalidades mais prioritárias desenvolvidas. As duas áreas foram: PP2.7.1 e PMC1.7.1. As demais *User Stories* mais prioritárias para cada área foram finalizadas.

A figura 26 ilustra o Release Burndown chart final do projeto, que mostra a relação entre os pontos planejados e os pontos que realmente foram completados durante o projeto.



**Figura 26: Release Burndown Chart final**

Devido ao projeto ser atípico e o time não se dedicar a ele em tempo integral, houve uma grande diferença na quantidade (em pontos) das User Stories entregues em cada Sprint. Por estes motivos, o time não conseguiu encontrar um padrão de velocidade ideal.

O escopo entregue foi menor do que o planejado, porém, como houve uma priorização das User Stories durante o planejamento do projeto, todas aquelas que não foram desenvolvidas tinham baixa prioridade e não agregavam tanto valor ao produto quanto aquelas que foram de fato implementadas.

A figura 27 mostra o Risk Backlog final do projeto.

## Risk Backlog

Backlog containing all the project risks.

View: <b>All Items</b>											
Edit	Risk	Priority	Category	Description	Impact	Probability	Strategy	Action	User responsible	Closed	Creation date
	Unfinished stories	0	None	The team might not be able to complete all planned user stories with the time frame of 2 sprints	Medium	Very high	Accept	The user stories that can't be finished will be developed in the next product's version	JOKER\scrum.master	Yes	5/18/2009
	Wrong user story points	1		The user story points can be wrongly estimated.	High	Medium	Mitigate/Improve	Review the planned User Story points for the next Sprint on its Planning Meeting and adjust the Product Backlog and Release Plan with the right estimations.	JOKER\nanda; JOKER\Leonardo	Yes	12/1/2008
	User Story dependency	2		Dependency from a User Story with high priority to another one with less priority.	Very low	Low	Mitigate/Improve	Review if the User Stories for the next Sprint have a dependency to another Story with less priority and adjust the Product Backlog and Release Plan with the new prioritization.	JOKER\nanda; JOKER\Leonardo	Yes	12/1/2008
	Less team's velocity	3		Team can have less velocity than the estimated.	Very high	High	Mitigate/Improve	Review team's velocity in the end of each Sprint and update the Product Backlog and Release Plan to reflect the new reality.	JOKER\nanda; JOKER\Leonardo	Yes	12/1/2008
	More team's velocity	4		Team can have more velocity than the estimated.	Medium	Very low	Accept	Review the Product Backlog items, trying to insert more User Stories.	JOKER\nanda; JOKER\Leonardo	Yes	12/1/2008
	User Story and technology	5		User Story not contemplate with the technology used.	Very high	Low	Mitigate/Improve	Modify the User Story to be made with the technology.	JOKER\nanda; JOKER\Leonardo	Yes	3/22/2009
	Sprint time	6		Identify that 2 weeks Sprints are inadequate.	Low	Very low	Mitigate/Improve	Change the next Sprints time, updating the impacted artifacts.	JOKER\nanda; JOKER\Leonardo	Yes	12/1/2008

**Figura 27: Risk Backlog final**

#### 4. CONCLUSÃO

Este trabalho foi desenvolvido com o intuito de analisar se a integração entre *Scrum* e CMMI traria uma melhoria na execução e gerenciamento de um projeto de software, tanto para times que já utilizam *Scrum* e pretendem realizar uma avaliação do CMMI quanto para equipes que seguem o CMMI e procuram uma abordagem ágil de gerência de projetos. Utilizando estas abordagens em conjunto, acreditávamos que seria possível obter a agilidade oriunda do *Scrum* e a maturidade agregada ao CMMI.

Durante a integração entre as metodologias, verificamos que algumas práticas específicas do CMMI podem ser satisfeitas pelo *Scrum*. Há casos em que o *Scrum* não satisfaz uma prática ou a satisfaz apenas parcialmente. Nestes casos foi sugerida a adoção de novos conceitos e técnicas ao *Scrum* que tornassem viável a integração deste com as áreas de processos selecionadas.

Uma vez que o processo proposto foi utilizado de forma efetiva no desenvolvimento e gerenciamento do projeto, foi possível analisar de forma prática esta integração. Além disto, podemos constatar que a ferramenta auxiliou nas práticas propostas para a adequação do *Scrum* ao CMMI, não criando sobrecarga ao time, já que muitas práticas foram automatizadas através da ferramenta. Como exemplos de práticas automatizadas temos: a criação de workflows, a criação e atualização dos Iteration Burndown Charts, versionamentos e geração de logs para evidência. A ferramenta se mostrou efetiva mesmo para projetos que utilizam o *Scrum*, mas não buscam realizar uma avaliação CMMI. A criação da ferramenta, não apenas possui funcionalidades que integram o *Scrum* às práticas do CMMI, como traz maior controle a um projeto ágil, pois integra seus processos de forma a fazer com que o Scrum Master e o time deixem de se preocupar com a integridade e confiabilidade dos dados e possam se focar em suas atividades específicas dentro do projeto. Este tipo de controle se torna muito mais difícil em projetos *Scrum* tradicionais, onde as atividades são controladas através de quadros contendo post-its em uma parede e planilhas distintas com os diversos dados do projeto.

A aplicabilidade da ferramenta foi validada através do seu próprio uso durante o desenvolvimento do projeto. Com esta experiência, foi possível perceber que a mesma pode cumprir com seus objetivos propostos ao ser utilizada em um projeto de desenvolvimento de software. Os conceitos e técnicas adotados para os casos

em que o Scrum não satisfaz uma prática ou a satisfaz parcialmente não oneraram o controle do projeto de maneira significativa, já que os mesmos, ou são simples de manter, como é o caso do Risk Backlog, ou são totalmente automatizados, como é o caso do histórico de riscos.

Como resultado destes estudos, podemos concluir que é viável integrar Scrum com CMMI, que o uso de princípios ágeis nas práticas do CMMI pode criar processos mais apropriados para a realidade das empresas e a incorporação dos objetivos do CMMI nas atividades de um projeto ágil pode auxiliar o desenvolvimento da maturidade dentro do time. Porém, esta integração, caso realizada sem a instrumentação adequada, pode gerar uma quantidade grande de documentações, evidências e outras práticas indicadas pelo CMMI que acabam por onerar um processo ágil. Neste ponto, o foco em uma ferramenta foi fundamental para tornar o processo proposto o mais leve e ágil possível, proporcionando ao mesmo tempo as evidências necessárias em uma avaliação CMMI.

A proposta agrega valor ao gerenciamento do projeto sem onerar o processo com eventuais sobrecargas de documentação, formalização e evidências que estão acopladas ao modelo de avaliação dos níveis de maturidade do CMMI, ou seja, a proposta visa simplificar a forma de seguir o processo e automatizar a forma de gerar as evidências de cumprimento do mesmo, para que este não seja sobrecarregado com o único intuito de mostrar que o processo está sendo seguido.

#### 4.1. DIFICULDADES ENCONTRADAS

Entre as dificuldades encontradas durante o desenvolvimento deste trabalho, podemos citar:

- A falta de consenso na bibliografia consultada sobre o Scrum dificultou a tomada de decisão sobre os padrões, processos e nomenclaturas do Scrum a serem utilizadas no trabalho, principalmente quanto à nomenclatura sobre seu ciclo de vida;
- Tivemos dificuldades em entender a aplicabilidade de algumas práticas do CMMI, fazendo com que fosse difícil descobrir a melhor maneira de integrar com o Scrum e de gerar evidências do uso destas práticas. Por exemplo: A prática de planejar e monitorar o gerenciamento de dados;

- Apesar ter facilitado em alguns casos e ser uma ótima ferramenta para prototipação de uma solução, o uso do Sharepoint dificultou a implementação de algumas User Stories e impossibilitou o desenvolvimento de outras.
- Pouca bibliografia sobre integração entre Scrum e CMMI.

#### 4.2. TRABALHOS FUTUROS

Todas as User Stories que não foram contempladas por esta versão da ferramenta poderão ser utilizadas como requisitos para próximas versões do sistema.

O Sharepoint foi uma ótima ferramenta para a prototipação e prova de conceito desta integração, porém, ela possui limitações consideráveis, limitando as funcionalidades e não recomendamos seu uso em trabalhos futuros.

Uma boa maneira de validar a ferramenta seria apresentá-la a avaliadores do CMMI para que os mesmos avaliem a proposta de integração. Outra maneira seria com seu uso prático em um projeto de software dentro de uma empresa.

A tabela 9 mostra um possível Product Backlog inicial que poderá ser utilizado em trabalhos futuros.

ID	User Story	Pontos
PP2.7.1	Produzir um documento que contenha todos os artefatos do projeto que sejam necessários para formar o plano de projeto em nível macro	8
PMC1.7.1	Mostrar a quantidade de User Stories planejadas, a quantidade das que foram entregues e a quantidade de novas User Stories identificadas ao final de cada Sprint para que isto sirva como evidência	3
PP1.1.2	Facilitar a priorização das User Stories dentro do Product Backlog	3
PP1.4.2	Gerar alertas quando o Sprint Backlog estiver alocando mais ou menos pontos que a velocidade da equipe.	5
PP2.7.2	Produzir um documento que contenha todos os artefatos do	5

	projeto que sejam necessários para formar o plano de projeto em nível de iteração	
<b>PMC1.1.2</b>	Gerar um Burndown chart de custos baseado no orçamento estimado e no número de iterações do projeto	5
<b>PMC1.3.2</b>	Enviar os riscos por e-mail semanalmente para o Product Owner, o Scrum Master e o time	2
<b>PMC1.6.2</b>	Enviar o Iteration Burndown chart periodicamente por e-mail para o Product Owner, solicitando confirmação de leitura	2
<b>PMC2.1.2</b>	Habilitar uma funcionalidade que auxilie na criação de impedimentos durante a Daily Scrum, associando automaticamente o membro do time que está bloqueado por este impedimento	3
<b>PP2.3.3</b>	Proporcionar a visualização de uma tabela contendo os Stakeholders do projeto, os artefatos disponíveis para o mesmo, seus níveis de acesso, além de como e onde estes dados estarão disponíveis	3
<b>PP2.6.3</b>	Permitir a criação e atualização de funções, para serem ligadas aos Stakeholders. Estas funções poderiam possuir um texto ou documento anexo informando as regras e responsabilidades daquela função	3
<b>PP3.1.2</b>	Permitir a visualização das versões dos planos que afetam o projeto, com a data e descrição de cada mudança	5
<b>PMC1.1.3</b>	Atualizar o Release Burndown chart automaticamente quando uma User Story for dada como pronta	8
<b>PMC1.3.3</b>	Enviar alertas sobre riscos que tiveram sua data de expiração atingida e não foram solucionados	5
<b>PMC1.6.3</b>	Enviar o Release Burndown chart por e-mail ao final de cada Sprint para os Stakeholders, solicitando confirmação de leitura	2
<b>PP2.1.4</b>	Permitir a inclusão dos orçamentos macro do projeto e do orçamento por iteração	3
<b>PP2.3.4</b>	Possibilitar a criação de regras de acesso para os artefatos do projeto	2
<b>PMC1.1.4</b>	Permitir a inserção do orçamento realizado em uma Sprint e atualizar o custo real da Sprint automaticamente no Burndown chart de custos	8
<b>PP2.1.5</b>	Gerar uma sugestão de Release Plan, a partir das User Stories	13

	já devidamente pontuadas e priorizadas e da velocidade do time, permitir alterações neste plano de forma simples e facilitar a visualização das User Stories que não estão vinculadas a uma Sprint	
<b>TC1.1</b>	Modificar a lista de User Stories do projeto mostrando as mesmas como post-its	8
<b>TC1.2</b>	Habilitar funcionalidades avançadas de log no sharepoint	3
<b>TC1.3</b>	Melhorar a arquitetura de informação e layout do portal	3
<b>TC2.3</b>	Melhorar a interface das tabelas de dados da Velocidade do Time e do Histórico de Riscos para que sejam mostrados em forma de gráficos	5
<b>TC4.2</b>	Criar uma lista, associada com os riscos, para guardar as ações tomadas durante o monitoramento dos riscos	3

**Tabela 9: *Product Backlog* inicial para trabalhos futuros**

## REFERÊNCIAS

AHERN, Dennis M.; CLOUSE, Aaron; TURNER, Richard. 2004. **CMMI Distilled**. Boston: Addison-Wesley, 2004.

ANDERSON, David J. 2005. **Agile Management**. Disponível em [http://www.agilemanagement.net/Articles/Papers/Agile\\_2005\\_Paper\\_DJA\\_v1\\_6.pdf](http://www.agilemanagement.net/Articles/Papers/Agile_2005_Paper_DJA_v1_6.pdf). Acesso em 05 de Setembro de 2008.

(Boehm, 2006) Boehm, Barry. **Center for Systems and Software Engineering**. Disponível em: <http://csse.usc.edu/csse/TECHRPTS/2006/usccsse2006-626/usccsse2006-626.pdf>. Acesso em 23 de Novembro de 2008.

(Chrissis, et al., 2003) Chrissis, Mary Beth, Konrad, Mike e Shrum, Sandy. **CMMI®: Guidelines for Process Integration and Product Improvement**. Boston : Addison-Wesley, 2003.

(CMMI Product Team, 2006) CMMI Product Team. **CMMI for Development, Version 1.2. Software Engineering Institute**. Disponível em: <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>.

(Cohn, 2005) Cohn, Mike. **Agile Estimating and Planning**. Prentice Hall, 2005.

(Cohn, 2004) Cohn, Mike. **Agile Estimating and Planning**. Addison Wesley, 2004.

(InfoQ, 2006) InfoQ. **Interview: Jim Johnson of the Standish Group**. Disponível em: <http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>. Acesso em 29 de novembro de 2008.

(Koch, 2005) Koch, Alan. **Agile Software Development**. Artech House, 2005.

(Kulpa, et al., 2003) Kulpa, Margaret e Kent, Johnson A. **Interpreting the CMMI: A Process Improvement Approach**. Auerbach, 2003.

(Larman, 2004) Larman, C. **Agile & Iterative Development, A Manager's Guide**. Addison-Wesley, 2004.

(Leffingwell, 2008) Leffingwell, Dean. **Scaling Software Agility**. Addison-Wesley, 2008.

(2001) **Manifesto for Agile Software Development**. Disponível em: <http://agilemanifesto.org>. Acesso em: 07 de setembro de 2008.

(PMBOK, 2004) **A Guide to the Project Management Body of Knowledge**. Project Management Institute, 2004.

(Schuh, 2005) Schuh, Peter. **Integrating Agile Development in the Real World**. Charles River Media, 2005.

(Schwaber, 2004) Schwaber, Ken. **Agile Project Management with Scrum**. Microsoft Press, 2004.

(Schwaber, et al., 2001) Schwaber, Ken e Beedle, Mike. **Agile Software Development with Scrum**. Prentice Hall, 2001.

(AgileCollab, 2008) AgileCollab. **Interview with Ken Schwaber**. Disponível em: <http://www.agilecollab.com/interview-with-ken-schwaber>. Acesso em 19 de Fevereiro de 2008.

(Schwaber) Schwaber, Ken. **SCRUM Development Process**. Disponível em: <http://jeffsutherland.com/oops/la/schwapub.pdf>. Acesso em: 30 de outubro de 2008.

(SEI, 2008) SEI. **CMMI® or Agile: Why Not Embrace Both!** Disponível em: <http://www.sei.cmu.edu/publications/documents/08.reports/08tn003.html>. Acesso em: 23 de Novembro de 2008.

(Standish Group, 1994) Standish Group. **Chaos Report**. Disponível em: [http://www.standishgroup.com/sample\\_research/chaos\\_1994\\_1.php](http://www.standishgroup.com/sample_research/chaos_1994_1.php). Acesso em: 22 de Novembro de 2008.

(Sutherland, 2006) Sutherland, Jeff. **Scrum Log Jeff Sutherland**. Disponível em: <http://jeffsutherland.com/scrum/2006/06/why-three-questions-in-daily-scrum.html>. Acesso em: 20 de Novembro de 2008

(Takeuchi, et al., 1986) Takeuchi, Hirotaka e Nonaka, Ikujiro. **Agile Project Leadership Network (APLN)**. Disponível em: <http://aplrichmond.pbwiki.com/f/New%20New%20Prod%20Devel%20Game.pdf>. Acesso em: 07 de setembro de 2008.

(Turner, et al., 2002) Turner, R.; Jain, A. Agile Meets CMMI: **Culture Clash or Common Cause**, In proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods XP/Agile Universe, pp. 153165, 2002.

## **ANEXO A – MANUAL DO SISTEMA**

## ÍNDICE

<b>1.</b>	<b>Introdução .....</b>	<b>128</b>
<b>2.</b>	<b>Funcionalidades do Sistema .....</b>	<b>128</b>
2.1.	PP2.1.3 - Permitir a criação de Sprints, com datas de início e fim.....	128
2.2.	PP1.1.1 - Proporcionar um template para a criação de <i>User Stories</i> e uma forma simples e prática de manutenção .....	131
2.3.	PP1.2.1 - Permitir a pontuação das <i>User Stories</i> no <i>Product Backlog</i> .....	136
2.4.	PP1.3.1 - Disponibilizar <i>checklists</i> com o que deve ser feito em cada fase para seguir o ciclo de vida do <i>Scrum</i> .....	136
2.5.	PP2.1.2 - Permitir a alocação de <i>User Stories</i> em uma determinada <i>Sprint</i> .	138
2.6.	PMC1.7.2 - Possuir uma funcionalidade para colocar as <i>User Stories</i> que foram completadas na <i>Sprint</i> como completas no <i>Product Backlog</i> , para que as estimativas do projeto sejam revistas .....	138
2.7.	PP1.4.1 - Gerar automaticamente a base histórica, com a velocidade do time em pontos por <i>Sprint</i> a partir dos dados das <i>Sprints</i> já realizadas, a fim de proporcionar dados consistentes com a realidade para uso nas estimativas da próxima <i>Sprint</i> .....	138
2.8.	PP2.1.1 - Permitir a criação de tarefas para uma determinada <i>User Story</i> , com prioridade, estimativa, recursos alocados e dependência entre esta e outras tarefas.....	139
2.9.	PP2.2.1 - Possuir um <i>Risk Backlog</i> , para conter os riscos, que podem ser criados com prioridade, categoria, descrição, impacto, probabilidade de ocorrência, estratégia, ação corretiva e usuário responsável pelo mesmo...	143
2.10.	PP2.3.1 - Proporcionar o versionamento dos artefatos do projeto .....	148
2.11.	PP2.4.1 - Possuir um repositório para a gerência de documentos sobre equipamentos, instalações elétricas, definições de processos, diagramas e outros artefatos necessários ao projeto para o gerenciamento de recursos	150
2.12.	TC4.1 – Criar um repositório para conter os documentos necessários ao <i>Scrum</i> .....	150
2.13.	PP2.6.2 - Permitir a criação e atualização de usuários no sistema que representem os <i>Stakeholders</i> .....	152

2.14.	PP2.5.1 - Possuir um inventário de habilidades necessárias ao projeto, para identificar as habilidades que estão faltando para completar o projeto e auxiliar na contratação de pessoas ou treinamentos que proporcionarão as habilidades necessárias para o projeto.....	155
2.15.	PP2.6.1 - Possuir uma lista contendo os <i>Stakeholders</i> relevantes e seu envolvimento no projeto.....	158
2.16.	TC4.4 – Criar Templates para documentos Scrum.....	158
2.17.	PP3.3.1 - Enviar uma notificação solicitando a aprovação do plano desenvolvido durante a <i>Planing Meeting</i> de uma <i>Sprint</i> .....	159
2.18.	PMC1.1.1 A - Proporcionar a inserção de horas realizadas e atualização das horas restantes nas tarefas de uma User Story.....	162
2.19.	PMC2.1.1 - Proporcionar o cadastro de itens dentro do Impediments <i>Backlog</i> , associando o membro do time bloqueado (quando houver), prioridade e membro que deve tomar alguma ação para corrigir o mesmo.....	162
2.20.	PMC1.2.1 - Possuir uma funcionalidade para que, durante a <i>Daily Scrum</i> , o <i>Scrum Master</i> insira, para cada membro do time, o que aquele membro fez desde a última <i>Daily Scrum</i> , o que ele planeja fazer até a próxima <i>Daily Scrum</i> e quais são os seus impedimentos.....	166
2.21.	PMC1.3.1 - Guardar um histórico de riscos para mostrar a quantidade de riscos encontrados no início do projeto, a quantidade de riscos fechados, os riscos que foram adicionados durante o projeto e a quantidade de riscos abertos.....	169
2.22.	PMC1.1.1B - Atualizar o Iteration <i>Burndown chart</i> automaticamente após cada mudança em uma tarefa.....	170
2.23.	TC4.3 – Inserir um status para cada tarefa de uma User Story.....	170
2.24.	PMC1.4.1 – Proporcionar o versionamento do plano de gerenciamento de dados com as revisões de alteração e visualização feitas pelo <i>Scrum Master</i> 171	
2.25.	PMC1.5.1 – Mostrar uma lista das <i>Daily Scrums</i> feitas no decorrer do projeto para que isto possa ser utilizado como evidência de registro do envolvimento dos <i>Stakeholders</i> .....	171
2.26.	PMC1.6.1 – Possuir um repositório para que sejam colocados documentos criados durante a reunião de Review, como uma ata ou um vídeo com a demonstração do incremento funcional .....	171

2.27. PMC2.2.1 – Permitir a inserção de uma data estimada para correção e ações corretivas a serem tomadas dentro de cada impedimento no Impediments Backlog.....	171
2.28. PMC2.3.1 – Armazenar todos os impedimentos já resolvidos no Impediment Backlog.....	172
2.29. PP2.3.2 - Habilitar o gerenciamento do <i>Product Backlog</i> apenas pelo o <i>Product Owner</i> e a visualização do mesmo para todo o time.....	172
2.30. PP3.1.1 – Permitir a visualização de uma determinada versão de um plano	173
2.31. PP1.1.3 – Realizar o versionamento do <i>Product Backlog</i> para que as informações nele contidas desde o início do projeto estejam disponíveis durante todo seu ciclo de vida .....	174

## 1. INTRODUÇÃO

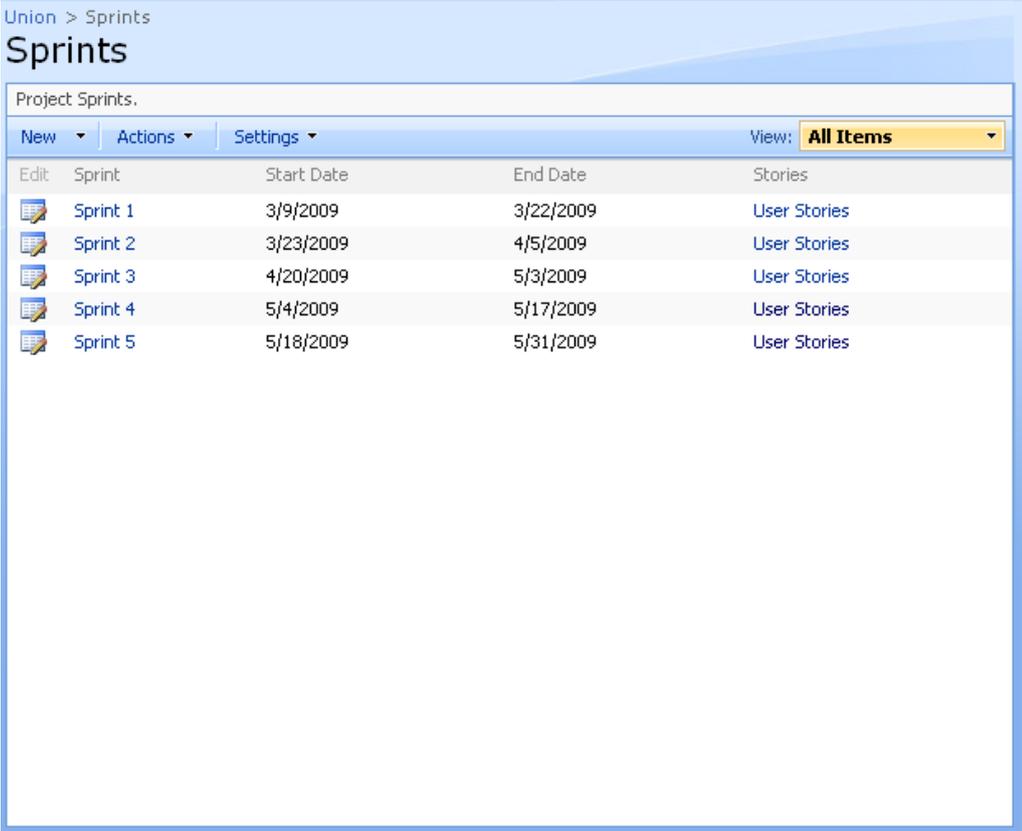
Este manual apresenta-se na forma de tutorial para explicar as opções e funcionalidades do sistema.

Estas funcionalidades são apresentadas na ordem em que foram desenvolvidas.

## 2. FUNCIONALIDADES DO SISTEMA

### 2.1. PP2.1.3 - PERMITIR A CRIAÇÃO DE SPRINTS, COM DATAS DE INICIO E FIM

Ao acessar a opção “Sprints” do menu “Lists”, o sistema irá exibir a lista de Sprints do projeto, como pode ser visto na figura 28:



Union > Sprints

### Sprints

Project Sprints.

New ▾ Actions ▾ Settings ▾ View: **All Items** ▾

Edit	Sprint	Start Date	End Date	Stories
	Sprint 1	3/9/2009	3/22/2009	<a href="#">User Stories</a>
	Sprint 2	3/23/2009	4/5/2009	<a href="#">User Stories</a>
	Sprint 3	4/20/2009	5/3/2009	<a href="#">User Stories</a>
	Sprint 4	5/4/2009	5/17/2009	<a href="#">User Stories</a>
	Sprint 5	5/18/2009	5/31/2009	<a href="#">User Stories</a>

**Figura 28: Lista de Sprints**

O sistema proporciona funcionalidades para criar, editar e remover uma Sprint.

Para criar uma nova Sprint, é necessário clicar no botão **New**. Neste momento o sistema irá exibir o formulário ilustrado na figura 29, que permitirá a criação de uma nova Sprint.

Union > Sprints > New Item

## Sprints: New Item

OK Cancel

Attach File | Spelling... \* indicates a required field

**Sprint \*** Sprint [Sprint number]  
Sprint name.

**Start Date** Sprint's start date.

**End Date** Sprint's End Date.

**Stories** Type the Web address: (Click here to test)  
http://  
Type the description:

OK Cancel

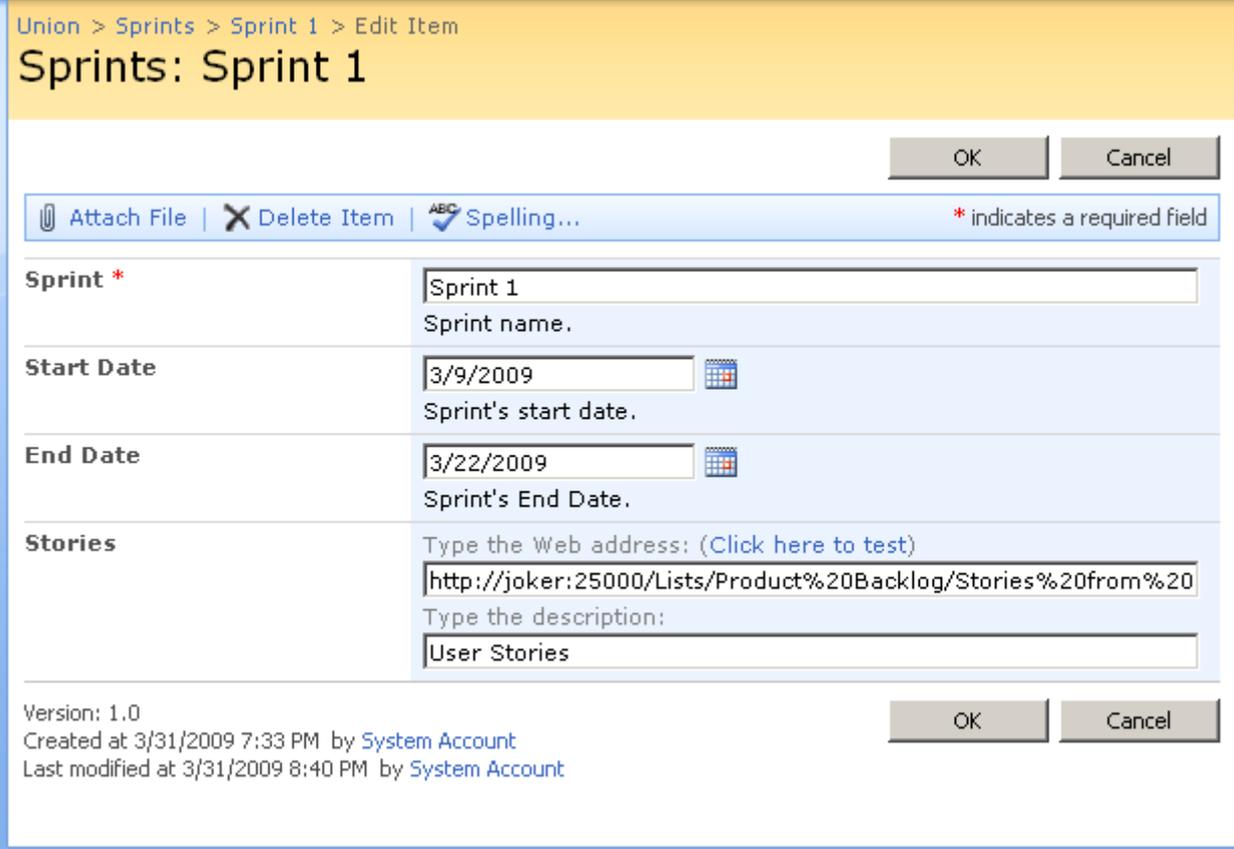
**Figura 29: Formulário de Criação de uma Nova Sprint**

O processo de criação de uma Sprint consiste em:

- Atribuir um nome a Sprint. Este é o único campo obrigatório no formulário. Por padrão, basta trocar a expressão “[Sprint number]” pelo número da Sprint.
- Inserir uma data de início e uma data de fim para a Sprint.
- Inserir um link que leve às User Stories daquela Sprint.

A única maneira de visualizar as User Stories de uma Sprint pelo Sharepoint é criando uma View que mostre somente as User Stories de uma determinada Sprint. Esta View pode ser acessada por um link específico. Este é o link que é colocado na criação de uma Sprint.

Para editar uma Sprint existente, é necessário clicar no botão  ao lado do nome da Sprint que se deseja editar. Neste momento o sistema irá exibir o formulário ilustrado na figura 30, que permitirá a edição desta Sprint.



Union > Sprints > Sprint 1 > Edit Item

## Sprints: Sprint 1

OK Cancel

 Attach File |  Delete Item |  Spelling... \* indicates a required field

<b>Sprint *</b>	<input type="text" value="Sprint 1"/> Sprint name.
<b>Start Date</b>	<input type="text" value="3/9/2009"/>  Sprint's start date.
<b>End Date</b>	<input type="text" value="3/22/2009"/>  Sprint's End Date.
<b>Stories</b>	Type the Web address: ( <a href="#">Click here to test</a> ) <input type="text" value="http://joker:25000/Lists/Product%20Backlog/Stories%20from%20"/> Type the description: <input type="text" value="User Stories"/>

Version: 1.0  
Created at 3/31/2009 7:33 PM by System Account  
Last modified at 3/31/2009 8:40 PM by System Account

OK Cancel

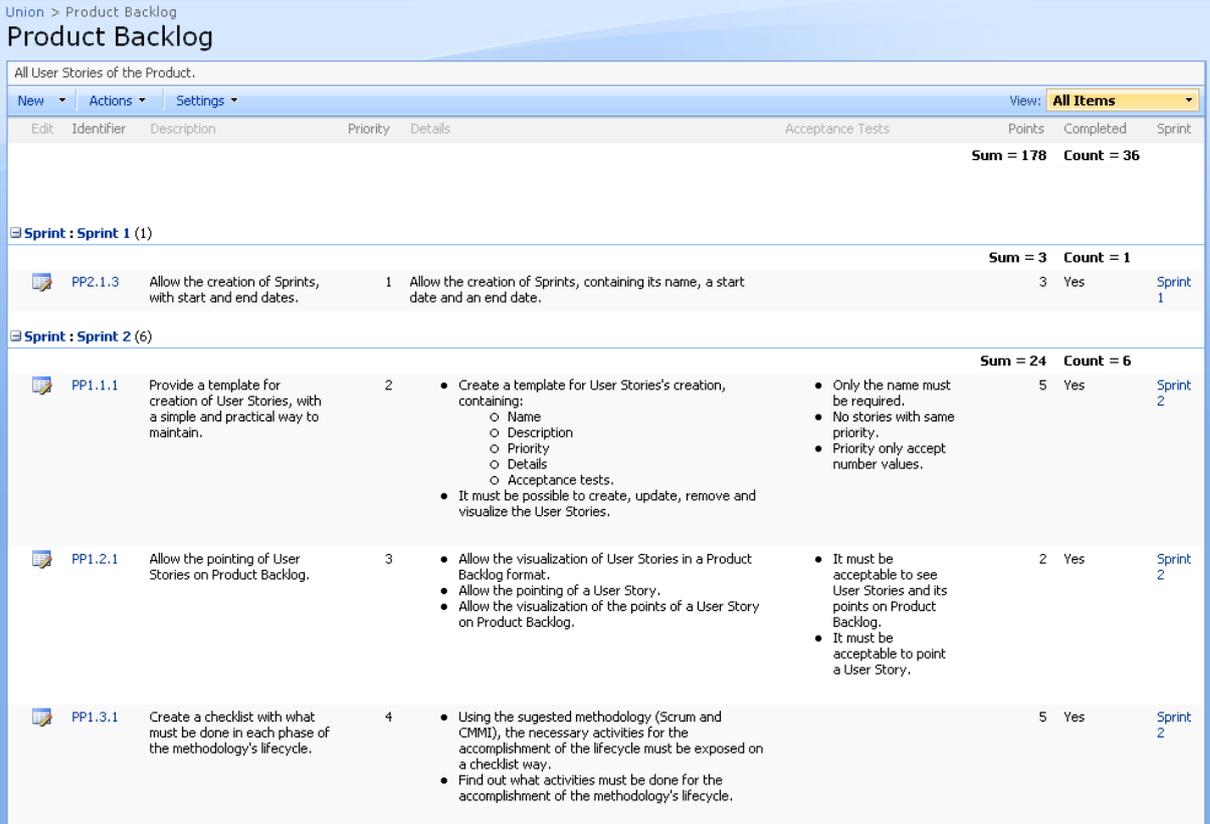
**Figura 30: Formulário de Edição de uma Sprint**

O processo de edição de uma Sprint consiste no mesmo processo que o da criação de uma Sprint.

Para remover uma Sprint existente, é necessário clicar no botão  ao lado do nome da Sprint que se deseja remover. Este botão levará ao formulário ilustrado na imagem 30. A partir deste formulário, para remover a Sprint, é necessário clicar no botão  **Delete Item** e confirmar a remoção.

## 2.2. PP1.1.1 - PROPORCIONAR UM TEMPLATE PARA A CRIAÇÃO DE *USER STORIES* E UMA FORMA SIMPLES E PRÁTICA DE MANUTENÇÃO

Ao acessar a opção “Product Backlog” do menu “Lists”, o sistema irá exibir a lista de User Stories do projeto, como pode ser visto na figura 31:



The screenshot shows the 'Product Backlog' interface. At the top, there is a navigation bar with 'Union > Product Backlog' and a title 'Product Backlog'. Below the title, it says 'All User Stories of the Product.' There are menu options for 'New', 'Actions', and 'Settings', and a 'View: All Items' dropdown. The main table has columns for 'Edit', 'Identifier', 'Description', 'Priority', 'Details', 'Acceptance Tests', 'Points', 'Completed', and 'Sprint'. A summary row shows 'Sum = 178' and 'Count = 36'. The table is grouped into two sprints: 'Sprint 1 (1)' and 'Sprint 2 (6)'. Each row represents a user story with its identifier, description, priority, details, acceptance tests, points, completion status, and assigned sprint.

Edit	Identifier	Description	Priority	Details	Acceptance Tests	Points	Completed	Sprint
						<b>Sum = 178</b>	<b>Count = 36</b>	
<b>Sprint : Sprint 1 (1)</b>								
	PP2.1.3	Allow the creation of Sprints, with start and end dates.	1	Allow the creation of Sprints, containing its name, a start date and an end date.		3	Yes	Sprint 1
						<b>Sum = 3</b>	<b>Count = 1</b>	
<b>Sprint : Sprint 2 (6)</b>								
	PP1.1.1	Provide a template for creation of User Stories, with a simple and practical way to maintain.	2	<ul style="list-style-type: none"><li>Create a template for User Stories's creation, containing:<ul style="list-style-type: none"><li>Name</li><li>Description</li><li>Priority</li><li>Details</li><li>Acceptance tests.</li></ul></li><li>It must be possible to create, update, remove and visualize the User Stories.</li></ul>	<ul style="list-style-type: none"><li>Only the name must be required.</li><li>No stories with same priority.</li><li>Priority only accept number values.</li></ul>	5	Yes	Sprint 2
	PP1.2.1	Allow the pointing of User Stories on Product Backlog.	3	<ul style="list-style-type: none"><li>Allow the visualization of User Stories in a Product Backlog format.</li><li>Allow the pointing of a User Story.</li><li>Allow the visualization of the points of a User Story on Product Backlog.</li></ul>	<ul style="list-style-type: none"><li>It must be acceptable to see User Stories and its points on Product Backlog.</li><li>It must be acceptable to point a User Story.</li></ul>	2	Yes	Sprint 2
	PP1.3.1	Create a checklist with what must be done in each phase of the methodology's lifecycle.	4	<ul style="list-style-type: none"><li>Using the suggested methodology (Scrum and CMMI), the necessary activities for the accomplishment of the lifecycle must be exposed on a checklist way.</li><li>Find out what activities must be done for the accomplishment of the methodology's lifecycle.</li></ul>		5	Yes	Sprint 2

**Figura 31: Product Backlog**

O sistema proporciona funcionalidades para criar, editar, visualizar e remover uma User Story.

Para criar uma nova User Story, é necessário clicar no botão **New**. Neste momento o sistema irá exibir o formulário ilustrado na figura 32:

Union > Product Backlog > New Item

## Product Backlog: New Item

OK Cancel

Attach File | Spelling... \* indicates a required field

<b>Identifier *</b>	<input type="text"/>
	Attribute that identifies the User Story.
<b>Description</b>	<input type="text"/>
	Describes the User Story.
<b>Priority</b>	<input type="text"/>
<b>Details</b>	<div style="border: 1px solid gray; padding: 5px;"> <div style="font-size: small; border-bottom: 1px solid gray; margin-bottom: 5px;"> <span>A A1</span>   <span>B I U</span>   <span>☰ ☷ ☹</span>   <span>☰ ☷ ☹</span> </div> <input style="width: 100%; height: 100%;" type="text"/> </div>
<b>Acceptance Tests</b>	<div style="border: 1px solid gray; padding: 5px;"> <div style="font-size: small; border-bottom: 1px solid gray; margin-bottom: 5px;"> <span>A A1</span>   <span>B I U</span>   <span>☰ ☷ ☹</span>   <span>☰ ☷ ☹</span> </div> <input style="width: 100%; height: 100%;" type="text"/> </div>
<b>Points</b>	<input type="text"/>
<b>Completed</b>	<input type="checkbox"/>
<b>Sprint</b>	<input type="text" value="(None)"/> <ul style="list-style-type: none"> <li>(None)</li> <li>Sprint 1</li> <li>Sprint 2</li> <li>Sprint 3</li> <li>Sprint 4</li> <li>Sprint 5</li> </ul>
	OK Cancel

Figura 32: Formulário de Criação de uma *User Story*

O processo de criação de uma User Story consiste em:

- Atribuir um identificador para a User Story. Este é o único campo obrigatório no formulário.
- Colocar uma descrição para a User Story.
- Colocar uma prioridade para a User Story, que pode ser qualquer número inteiro.
- Adicionar detalhes para a User Story.
- Adicionar testes de aceitação para a User Story.
- Informar os pontos da User Story.
- Informar se a User Story foi completada ou não.
- Informar a Sprint que a User Story será desenvolvida. As Sprints aparecem em um ComboBox, vindas da própria lista de Sprints do projeto. Qualquer adição, edição ou remoção de uma Sprint refletirá aqui.

Para editar uma User Story existente, é necessário clicar no botão  ao lado do nome da User Story que se deseja editar. Neste momento o sistema irá exibir o formulário ilustrado na figura 33, que permitirá a edição desta User Story.

Union > Product Backlog > PP1.1.1 > Edit Item

## Product Backlog: PP1.1.1

OK Cancel

Attach File | Delete Item | Spelling... \* indicates a required field

<b>Identifier *</b>	PP1.1.1 Attribute that identifies the User Story.
<b>Description</b>	Provide a template for creation of User Stories, with a simple and practical way to maintain. Describes the User Story.
<b>Priority</b>	2
<b>Details</b>	<ul style="list-style-type: none"> <li>• Create a template for User Stories's creation, containing: <ul style="list-style-type: none"> <li>○ Name</li> <li>○ Description</li> <li>○ Priority</li> <li>○ Details</li> <li>○ Acceptance tests.</li> </ul> </li> <li>• It must be possible to create, update, remove and visualize the User Stories.</li> </ul>
<b>Acceptance Tests</b>	<ul style="list-style-type: none"> <li>• Only the name must be required.</li> <li>• No stories with same priority.</li> <li>• Priority only accept number values.</li> </ul>
<b>Points</b>	5
<b>Completed</b>	<input checked="" type="checkbox"/>
<b>Sprint</b>	Sprint 2

Version: 1.0  
Created at 3/31/2009 8:22 PM by System Account  
Last modified at 3/31/2009 8:23 PM by System Account

OK Cancel

**Figura 33: Formulário de Edição de uma User Story**

O processo de edição de uma User Story consiste no mesmo processo que o da criação.

Para visualizar uma User Story, basta clicar no identificador da mesma na lista de User Stories. A figura 34 ilustra a visualização de uma User Story:

Union > Product Backlog > PP1.1.1

## Product Backlog: PP1.1.1

Close

[New Item](#) | 
 [Edit Item](#) | 
 [Delete Item](#) | 
 [Manage Permissions](#) | 
 [Alert Me](#) | 
 [Version History](#)

<b>Identifier</b>	PP1.1.1
<b>Description</b>	Provide a template for creation of User Stories, with a simple and practical way to maintain.
<b>Priority</b>	2
<b>Details</b>	<ul style="list-style-type: none"> <li>• Create a template for User Stories's creation, containing:             <ul style="list-style-type: none"> <li>○ Name</li> <li>○ Description</li> <li>○ Priority</li> <li>○ Details</li> <li>○ Acceptance tests.</li> </ul> </li> <li>• It must be possible to create, update, remove and visualize the User Stories.</li> </ul>
<b>Acceptance Tests</b>	<ul style="list-style-type: none"> <li>• Only the name must be required.</li> <li>• No stories with same priority.</li> <li>• Priority only accept number values.</li> </ul>
<b>Points</b>	5
<b>Completed</b>	Yes
<b>Sprint</b>	<a href="#">Sprint 2</a>

Version: 1.0  
 Created at 3/31/2009 8:22 PM by System Account  
 Last modified at 3/31/2009 8:23 PM by System Account

Close

**Figura 34: Formulário de Visualização de uma *User Story***

Não é possível editar a User Story que está sendo visualizada nesta mesma tela, para editá-la, é necessário clicar no botão [Edit Item](#).

Para remover uma User Story existente, é necessário acessar o formulário de edição ou visualização de User Stories. A partir de um destes formulários, para remover a User Story, é necessário clicar no botão [Delete Item](#) e confirmar a remoção.

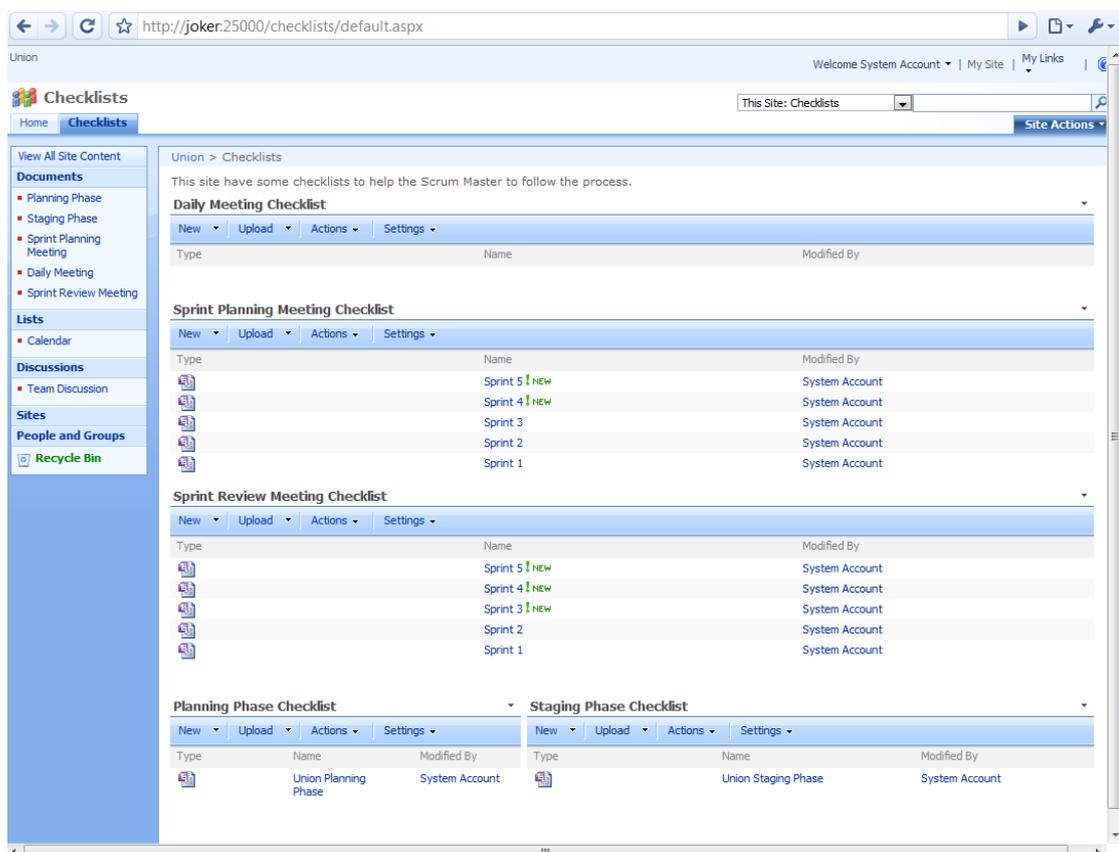
## 2.3. PP1.2.1 - PERMITIR A PONTUAÇÃO DAS *USER STORIES* NO *PRODUCT BACKLOG*

É possível pontuar uma User Story quando se cria ou edita uma. A funcionalidade PP1.1.1, detalhada na seção 2.2, mostra como criar e editar *User Stories*.

## 2.4. PP1.3.1 - DISPONIBILIZAR *CHECKLISTS* COM O QUE DEVE SER FEITO EM CADA FASE PARA SEGUIR O CICLO DE VIDA DO *SCRUM*

A ferramenta proporciona *checklists* para auxiliar o Scrum Master e o Time no cumprimento do processo.

Para acessar os *checklists*, é necessário clicar na aba Checklists do menu superior.



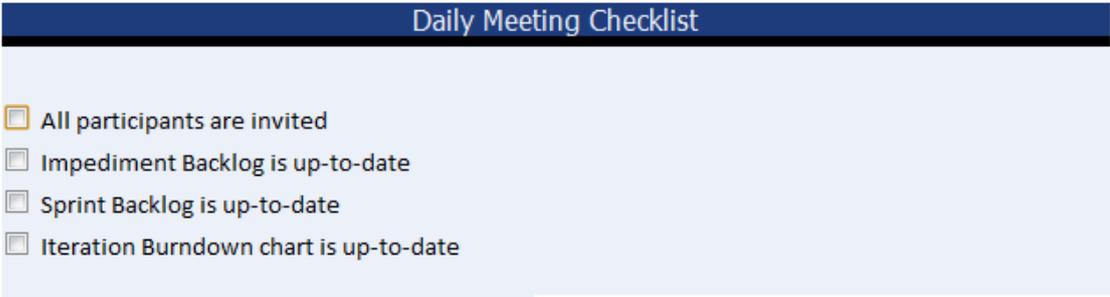
**Figura 35: Página de *Checklists***

Há cinco tipos diferentes de *checklists*:

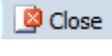
- Daily Meeting: checklist para ser usado no planejamento e execução da Planning Meeting
- Sprint Planning Meeting: checklist para ser utilizado no planejamento e execução da Planning Meeting
- Sprint Review Meeting: checklist para ser utilizado no planejamento e execução da Review Meeting
- Planning Phase: check list para ser utilizado durante a fase de planejamento do projeto
- Staging Phase: check list para ser utilizado durante a fase de preparação do projeto

Para criar um novo checklist, basta clicar no botão  e escolher o template desejado. O formulário com o checklist abrirá e poderá ser utilizado.

---



**Figura 36: Exemplo de Checklist**

Este formulário pode ser salvo clicando no botão . Uma caixa de diálogo irá aparecer para que o seja possível entrar com um nome para o novo formulário. Após completar o *checklist* e salvar, basta clicar no botão  para fechar o formulário e voltar para a página de *checklists*.

Para editar um *checklist* basta clicar no nome do formulário na lista e ele irá abrir para ser editado. Para salvar, basta clicar no botão  e após, clicar no botão  para fechar o formulário e voltar para a página de *checklists*.

## 2.5. PP2.1.2 - PERMITIR A ALOCAÇÃO DE *USER STORIES* EM UMA DETERMINADA *SPRINT*

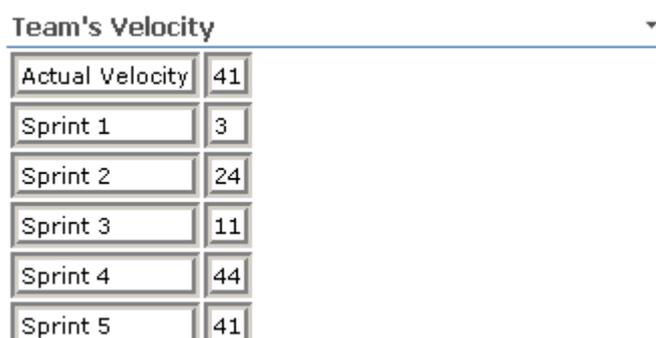
É possível alocar uma User Story à uma determinada Sprint ao se criar ou editar uma User Story. A funcionalidade PP1.1.1, na seção 2.2, mostra como criar e editar User Stories.

## 2.6. PMC1.7.2 - POSSUIR UMA FUNCIONALIDADE PARA COLOCAR AS *USER STORIES* QUE FORAM COMPLETADAS NA *SPRINT* COMO COMPLETAS NO *PRODUCT BACKLOG*, PARA QUE AS ESTIMATIVAS DO PROJETO SEJAM REVISTAS

É possível dar uma User Story como pronta ao se criar ou editar a mesma. A funcionalidade PP1.1.1, na seção 2.2, mostra como criar e editar User Stories.

## 2.7. PP1.4.1 - GERAR AUTOMATICAMENTE A BASE HISTÓRICA, COM A VELOCIDADE DO TIME EM PONTOS POR *SPRINT* A PARTIR DOS DADOS DAS *SPRINTS* JÁ REALIZADAS, A FIM DE PROPORCIONAR DADOS CONSISTENTES COM A REALIDADE PARA USO NAS ESTIMATIVAS DA PRÓXIMA *SPRINT*.

A Web Part “Team’s Velocity”, que pode ser visualizada na aba [Home](#), informa a velocidade histórica do time. A figura 37 ilustra esta funcionalidade:



Team's Velocity	
Actual Velocity	41
Sprint 1	3
Sprint 2	24
Sprint 3	11
Sprint 4	44
Sprint 5	41

Figura 37: *Team's Velocity Web Part*

A velocidade histórica do time é dada em pontos e seus dados são retirados das User Stories finalizadas das Sprints que já foram completadas. A velocidade atual é a mesma da velocidade da última Sprint.

## 2.8. PP2.1.1 - PERMITIR A CRIAÇÃO DE TAREFAS PARA UMA DETERMINADA USER STORY, COM PRIORIDADE, ESTIMATIVA, RECURSOS ALOCADOS E DEPENDÊNCIA ENTRE ESTA E OUTRAS TAREFAS

Ao acessar a opção “User Story Tasks” do menu “Lists”, o sistema irá exibir a lista de tarefas das User Stories do projeto, como pode ser visto na figura 38:

Edit	Task	Priority	Estimated hours	Allocated resource	Dependency	User Story	Worked Hours	Remaining Hours	Status	Sprint
<b>Sum = 117</b>										
						<b>Sum = 24</b>				
<b>User Story : Bug Fixing (2)</b>										
						<b>Sum = 5</b>				
						<b>Sum = 0</b>				
	Fix the Team's velocity web part issue		2	JOKER\nanda		Bug Fixing	2	0	Completed	Sprint 5
	Fix the workflow issue		3	JOKER\Leonardo		Bug Fixing	3	0	Completed	Sprint 5
<b>User Story : PMC1.1.1 A (1)</b>										
						<b>Sum = 1</b>				
						<b>Sum = 0</b>				
	Update the Tasks list with the new fields		1	JOKER\Leonardo		PMC1.1.1 A	1	0	Completed	Sprint 4
<b>User Story : PMC1.1.1 B (8)</b>										
						<b>Sum = 30</b>				
						<b>Sum = 4</b>				
	Add a button to update the chart		3	JOKER\nanda		PMC1.1.1 B	1	0	Completed	Sprint 5
	Investigate asp net charts.	0	4	JOKER\nanda		PMC1.1.1 B	4	0	Completed	Sprint 5
	Create an iteration burndown chart mock using ASP	0	8	JOKER\nanda		PMC1.1.1 B	4	0	Completed	Sprint 5
	Create the real line for estimated hours	1	4	JOKER\nanda		PMC1.1.1 B	4	4	Completed	Sprint 5
	Create the real line for remainin hours per day	2	4	JOKER\nanda		PMC1.1.1 B	4	0	Completed	Sprint 5
	Update the chart to work with different sprints	3	4	JOKER\nanda		PMC1.1.1 B	2	0	Completed	Sprint 5
	Insert the chart on sharepoint	4	1	JOKER\nanda		PMC1.1.1 B	1	0	Completed	Sprint 5
	Create a trigger to get the remaining hours everyday	5	2	JOKER\nanda		PMC1.1.1 B	2	0	Completed	Sprint 5

**Figura 38: User Stories Tasks**

O sistema proporciona funcionalidades para criar, editar, visualizar e remover uma tarefa.

Para criar uma nova tarefa, é necessário clicar no botão **New**. Neste momento o sistema irá exibir o formulário ilustrado na figura 39:

Union > User Story Tasks > New Item

## User Story Tasks: New Item

OK Cancel

Attach File | Spelling... \* indicates a required field

<b>Task *</b>	<input type="text"/>	Name of the task.
<b>Priority</b>	<input type="text"/>	
<b>Estimated hours</b>	<input type="text" value="0"/>	Insert the estimated hours (usually between 4 and 16 hours).
<b>Allocated resource</b>	<input type="text"/>	  
<b>Dependency</b>	<input type="text" value=""/>	Dependency between that and other task
<b>User Story</b>	<input type="text" value=""/>	
<b>Worked Hours</b>	<input type="text" value="0"/>	The number of hours worked on the task.
<b>Remaining Hours</b>	<input type="text"/>	The number of hours that is remaining to complete the task
<b>Status</b>	<input type="text" value="New"/>	

OK Cancel

**Figura 39: Formulário de Inserção de Nova Tarefa**

O processo de criação de uma tarefa consiste em:

- Atribuir um nome para a tarefa. Este é o único campo obrigatório no formulário.
- Informar a prioridade da tarefa.
- Informar as horas estimadas da tarefa. O Scrum recomenda a criação de tarefas entre 4 a 16 horas.
- Colocar o recurso alocado a esta tarefa. Este recurso vem da lista de recursos do projeto.
- Colocar outra tarefa da qual esta é dependente (se houver). Ao clicar na seta ao lado de Dependency, é possível selecionar a tarefa da qual esta depende.

- Informar a User Story a qual esta tarefa pertence. Ao clicar na seta ao lado de User Story, é possível selecionar a User Story a qual esta pertence.
- Número de horas trabalhadas. Por padrão, a criação, uma tarefa vem com 0 horas trabalhadas.
- Informar a quantidade de horas restantes.
- Informar o status da tarefa, que pode ser New, On going e Completed.

Para editar uma tarefa existente, é necessário clicar no botão  ao lado do nome da tarefa que se deseja editar. Neste momento o sistema irá exibir o formulário ilustrado na figura 40, que permitirá a edição desta tarefa.

Union > User Story Tasks > Insert documents > Edit Item

## User Story Tasks: Insert documents

 Attach File | 
  Delete Item | 
  Spelling...
\* indicates a required field

<b>Task *</b>	<input type="text" value="Insert documents"/> <small>Name of the task.</small>
<b>Priority</b>	<input type="text" value="2"/>
<b>Estimated hours</b>	<input type="text" value="1"/> <small>Insert the estimated hours (usually between 4 and 16 hours).</small>
<b>Allocated resource</b>	<input type="text" value="JOKER\Leonardo"/>  
<b>Dependency</b>	<input type="text" value="Create the repository"/>  <small>Dependency between that and other task</small>
<b>User Story</b>	<input type="text" value="PP2.4.1"/> 
<b>Worked Hours</b>	<input type="text" value="1"/> <small>The number of hours worked on the task.</small>
<b>Remaining Hours</b>	<input type="text" value="0"/> <small>The number of hours that is remaining to complete the task</small>
<b>Status</b>	<input type="text" value="Completed"/> 

Version: 3.0  
Created at 5/1/2009 3:50 PM by System Account  
Last modified at 5/20/2009 11:06 PM by System Account

**Figura 40: Formulário de Edição de uma Tarefa**

O processo de edição de uma tarefa consiste no mesmo processo que o da criação.

Para visualizar uma tarefa, basta clicar no identificador da mesma na lista de tarefas. A figura 41 lustra a visualização de uma tarefa:

Union > User Story Tasks > Insert documents

## User Story Tasks: Insert documents

Close

New Item | Edit Item | Delete Item | Manage Permissions | Alert Me | Version History

<b>Task</b>	Insert documents
<b>Priority</b>	2
<b>Estimated hours</b>	1
<b>Allocated resource</b>	JOKER\Leonardo
<b>Dependency</b>	Create the repository
<b>User Story</b>	PP2.4.1
<b>Worked Hours</b>	1
<b>Remaining Hours</b>	0
<b>Status</b>	Completed

Version: 3.0  
Created at 5/1/2009 3:50 PM by System Account  
Last modified at 5/20/2009 11:06 PM by System Account

Close

**Figura 41: Formulário de Visualização de Tarefas**

Não é possível editar a tarefa que está sendo visualizada nesta mesma tela, para editá-la, é necessário clicar no botão [Edit Item](#).

Para remover uma tarefa existente, é necessário acessar o formulário de edição ou visualização de tarefas. A partir de um destes formulários, para remover a tarefa, é necessário clicar no botão [Delete Item](#) e confirmar a remoção.

2.9. PP2.2.1 - POSSUIR UM RISK *BACKLOG*, PARA CONTER OS RISCOS, QUE PODEM SER CRIADOS COM PRIORIDADE, CATEGORIA, DESCRIÇÃO, IMPACTO, PROBABILIDADE DE OCORRÊNCIA, ESTRATÉGIA, AÇÃO CORRETIVA E USUÁRIO RESPONSÁVEL PELO MESMO

Ao acessar a opção “Risk Backlog” do menu “Lists”, o sistema irá exibir a lista dos riscos do projeto, como pode ser visto na figura 42:

Union > Risk Backlog

Risk Backlog

Backlog containing all the project risks.

New Actions Settings View: All Items

Edit	Risk	Priority	Category	Description	Impact	Probability	Strategy	Action	User responsible	Closed	Creation date
	Unfinished stories	0	None	The team might not be able to complete all planned user stories with the time frame of 2 sprints	Medium	Very high	Accept	The user stories that can't be finished will be developed in the next product's version	JOKER\scrum.master	Yes	5/18/2009
	Wrong user story points	1		The user story points can be wrongly estimated.	High	Medium	Mitigate/Improve	Review the planned User Story points for the next Sprint on its Planning Meeting and adjust the Product Backlog and Release Plan with the right estimations.	JOKER\nanda; JOKER\Leonardo	Yes	12/1/2008
	User Story dependency	2		Dependency from a User Story with high priority to another one with less priority.	Very low	Low	Mitigate/Improve	Review if the User Stories for the next Sprint have a dependency to another Story with less priority and adjust the Product Backlog and Release Plan with the new prioritization.	JOKER\nanda; JOKER\Leonardo	Yes	12/1/2008
	Less team's velocity	3		Team can have less velocity than the estimated.	Very high	High	Mitigate/Improve	Review team's velocity in the end of each Sprint and update the Product Backlog and Release Plan to reflect the new reality.	JOKER\nanda; JOKER\Leonardo	Yes	12/1/2008
	More team's velocity	4		Team can have more velocity than the estimated.	Medium	Very low	Accept	Review the Product Backlog items, trying to insert more User Stories.	JOKER\nanda; JOKER\Leonardo	Yes	12/1/2008
	User Story and technology	5		User Story not contemplate with the technology used.	Very high	Low	Mitigate/Improve	Modify the User Story to be made with the technology.	JOKER\nanda; JOKER\Leonardo	Yes	3/22/2009
	Sprint time	6		Identify that 2 weeks Sprints are inadequate.	Low	Very low	Mitigate/Improve	Change the next Sprints time, updating the impacted artifacts.	JOKER\nanda; JOKER\Leonardo	Yes	12/1/2008

Figura 42: Risk Backlog

O sistema proporciona funcionalidades para criar, editar, visualizar e remover um risco.

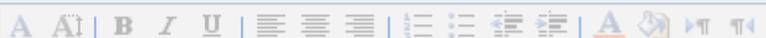
Para criar um novo risco, é necessário clicar no botão **New**. Neste momento o sistema irá exibir o formulário ilustrado na figura 43.

Union > Risk Backlog > New Item

## Risk Backlog: New Item

OK Cancel

Attach File | Spelling... \* indicates a required field

<b>Risk *</b>	<input type="text"/> Risk's name.
<b>Priority</b>	<input type="text"/>
<b>Category</b>	<input type="text"/> Type of the risk according the project's reality, when necessary.
<b>Description</b>	<div style="border: 1px solid gray; padding: 5px;">     </div>
<b>Impact</b>	<input type="text" value="Medium"/>
<b>Probability</b>	<input type="text" value="Medium"/>
<b>Strategy</b>	<input type="text"/> Answer's strategy.
<b>Action</b>	<div style="border: 1px solid gray; padding: 5px;">     </div>
<b>User responsible</b>	<input type="text"/> Enter users separated with semicolons.  
<b>Closed</b>	<input type="checkbox"/> Tells if a risk is closed or not.
<b>Creation date *</b>	<input type="text" value="6/13/2009"/>  Date of the risk's creation.

OK Cancel

Figura 43: Formulário para Criação de Novo Risco

O processo de criação de um risco consiste em:

- Atribuir um nome para o risco. Este campo é obrigatório no formulário.
- Adicionar uma prioridade ao risco.
- Adicionar uma categoria ao risco.
- Adicionar uma descrição ao risco.
- Adicionar um impacto ao risco. Um impacto pode ser: Muito baixo, baixo, médio, alto e muito alto.
- Adicionar uma probabilidade ao risco. Uma probabilidade pode ser: Muito baixa, baixa, média, alta e muito alta.
- Adicionar uma estratégia de resposta ao risco. Existem quatro estratégias de resposta que podem ser selecionadas. Prevenir/Explorar, Transferir/Compartilhar, Mitigar/Melhorar ou Aceitar.
- Informar a ação a ser tomada.
- Informar o usuário responsável por tratar este risco.
- Informar se o risco está aberto ou fechado.
- Informar a data de criação do risco. Por padrão, a data de criação vem com o dia de hoje. A data de criação serve para auxiliar a User Story PMC1.3.1 a descobrir se um risco foi criado antes ou durante o desenvolvimento do projeto.

Para editar um risco existente, é necessário clicar no botão  ao lado do nome do risco que se deseja editar. Neste momento o sistema irá exibir o formulário ilustrado na figura 44, que permitirá a edição deste risco.

Union > Risk Backlog > Unfinished stories > Edit Item

## Risk Backlog: Unfinished stories

OK Cancel

Attach File | 
 Delete Item | 
 Spelling...
 \* indicates a required field

<b>Risk *</b>	<input type="text" value="Unfinished stories"/> Risk's name.
<b>Priority</b>	<input type="text" value="0"/>
<b>Category</b>	<input type="text" value="None"/> Type of the risk according the project's reality, when necessary.
<b>Description</b>	<div style="border: 1px solid gray; padding: 5px;"> <p>The team might not be able to complete all planned user stories with the time frame of 2 sprints</p> </div>
<b>Impact</b>	<input type="text" value="Medium"/>
<b>Probability</b>	<input type="text" value="Very high"/>
<b>Strategy</b>	<input type="text" value="Accept"/> Answer's strategy.
<b>Action</b>	<div style="border: 1px solid gray; padding: 5px;"> <p>The user stories that can't be finished will be developed in the next product's versin</p> </div>
<b>User responsible</b>	<input type="text" value="JOKER\scrum.master"/> Enter users separated with semicolons.
<b>Closed</b>	<input checked="" type="checkbox"/> Tells if a risk is closed or not.
<b>Creation date *</b>	<input type="text" value="5/18/2009"/> Date of the risk's creation.

Version: 2,0  
 Created at 5/18/2009 9:36 PM by [System Account](#)  
 Last modified at 6/7/2009 2:48 PM by [System Account](#)

OK Cancel

Figura 44: Formulário de Edição de um Risco

O processo de edição de um risco consiste no mesmo processo que o da criação.

Para visualizar um risco, basta clicar no identificador do mesmo na lista de riscos. A figura 45 ilustra a visualização de um risco:

Union > Risk Backlog > Unfinished stories

## Risk Backlog: Unfinished stories

Close

New Item | Edit Item | Delete Item | Manage Permissions | Alert Me | Version History

<b>Risk</b>	Unfinished stories
<b>Priority</b>	0
<b>Category</b>	None
<b>Description</b>	The team might not be able to complete all planned user stories with the time frame of 2 sprints
<b>Impact</b>	Medium
<b>Probability</b>	Very high
<b>Strategy</b>	Accept
<b>Action</b>	The user stories that can't be finished will be developed in the next product's versin
<b>User responsible</b>	JOKER\scrum.master
<b>Closed</b>	Yes
<b>Creation date</b>	5/18/2009

Version: 2.0  
Created at 5/18/2009 9:36 PM by System Account  
Last modified at 6/7/2009 2:48 PM by System Account

Close

**Figura 45: Formulário de Visualização de um Risco**

Não é possível editar o risco que está sendo visualizado nesta mesma tela, para editá-lo, é necessário clicar no botão  [Edit Item](#).

Para remover um risco existente, é necessário acessar o formulário de edição ou visualização de riscos. A partir de um destes formulários, para remover o risco, é necessário clicar no botão  [Delete Item](#) e confirmar a remoção.

## 2.10. PP2.3.1 - PROPORCIONAR O VERSIONAMENTO DOS ARTEFATOS DO PROJETO

Todos os documento e listas do projeto são versionados. Sempre que um elemento de uma lista ou um documento é modificado, uma nova versão do mesmo é criada. Antigas versões podem ser vistas, restauradas ou deletadas.

Para ver a lista de versões de um elemento de uma lista, deve-se fazer o seguinte:

- Acessar a lista.
- Visualizar o elemento desejado (clcando em cima do elemento).
- Clicar no botão  Version History .
- Exemplo:
  - Lista “Risk Backlog”.
  - Elemento “Wrong user story points”.

Ao visualizar o elemento, é possível encontrar o botão  Version History .

Ao clicar no botão  Version History do elemento “Wrong user story points”, é possível visualizar todo o histórico de versões deste elemento, como ilustrado na figura 46:

Union > Risk Backlog > Wrong user story points > Version History

## Versions saved for Wrong user story points

All versions of this item are listed below with the new value of any changed properties.

[Delete All Versions](#)

No. ↓	Modified	Modified By
3.0	6/7/2009 2:49 PM	System Account
	Closed	Yes
2.0	5/9/2009 5:27 PM	System Account
	Closed	No
	Creation date	12/1/2008
1.0	4/24/2009 8:37 PM	System Account
	Risk	Wrong user story points
	Priority	1
	Description	The user story points can be wrongly estimated.
	Impact	High
	Probability	Medium
	Strategy	Mitigate/Improve
	Action	Review the planned User Story points for the next Sprint on its Planning Meeting and adjust the Product Backlog and Release Plan with the right estimations.
	User responsible	JOKER\nanda JOKER\Leonardo

**Figura 46: Visualização de Versões de um Item**

A figura 47 mostra a criação do risco (versão 1.0), a atualização do risco para que sejam colocados os campos “Closed” e “Creation Date” (versão 2.0) e a atualização do campo “Closed” para “Yes” (versão 3.0).

Para visualizar, restaurar ou deletar uma versão específica, basta arrastar o mouse em cima da data da versão e clicar na seta .

2.0	5/9/2009 5:27 PM			
	Closed	No		<a href="#">View</a>
	Creation date	12/1/2008		<a href="#">Restore</a>
1.0	4/24/2009 8:37 PM			<a href="#">Delete</a>
	Risk	Wrong user story points		

**Figura 47: Visualização dos Elementos de Versionamento**

Para visualizar a versão, clique em [View](#) . Para restaurar a versão, clique em [Restore](#) . Para deletar a versão, clique em [Delete](#) .

Com versões de documentos, a única diferença é como acessar o histórico de versões. Para visualizar o histórico de versões de um documento, é necessário arrastar o mouse em cima do nome do documento e clicar na seta .

Para visualizar o histórico de versões, basta clicar em  [Version History](#).

Para visualizar, restaurar ou deletar versões de documentos, deve-se seguir o mesmo procedimento.

#### 2.11. PP2.4.1 - POSSUIR UM REPOSITÓRIO PARA A GERÊNCIA DE DOCUMENTOS SOBRE EQUIPAMENTOS, INSTALAÇÕES ELÉTRICAS, DEFINIÇÕES DE PROCESSOS, DIAGRAMAS E OUTROS ARTEFATOS NECESSÁRIOS AO PROJETO PARA O GERENCIAMENTO DE RECURSOS

Ao acessar a opção “General Documents repository” do menu “Documents”, o sistema irá exibir o repositório de documentos gerais do projeto. Este repositório serve para que sejam adicionados todos os documentos sobre equipamentos, instalações elétricas, definições de processos, diagramas e outros artefatos necessários ao projeto.

Para efetuar o Upload de um documento, basta clicar no botão .

#### 2.12. TC4.1 – CRIAR UM REPOSITÓRIO PARA CONTER OS DOCUMENTOS NECESSÁRIOS AO SCRUM

Ao acessar a opção “Scrum Documents” do menu “Documents”, o sistema irá exibir o repositório de documentos Scrum do projeto, como pode ser visualizado na figura 48.

Union > Scrum Documents

## Scrum Documents

Repository for the Planning Meeting, review meeting, Retrospective and other necessary documents for Scrum.

New ▾ Upload ▾ Actions ▾ Settings ▾ View: **All Documents** ▾

Type	Name	Modified	Modified By	Sprint	Scrum Documents Approval
<b>[-] Sprint : Sprint 1 (3)</b>					
	Planning Meeting of Sprint 1	5/17/2009 2:58 PM	System Account	Sprint 1	
	Retrospective of Sprint1	5/17/2009 2:58 PM	System Account	Sprint 1	
	Review Meeting of Sprint 1	5/17/2009 2:58 PM	System Account	Sprint 1	
<b>[-] Sprint : Sprint 2 (3)</b>					
	Planning Meeting of Sprint 2	5/5/2009 9:45 PM	System Account	Sprint 2	
	Retrospective of Sprint 2	5/5/2009 9:46 PM	System Account	Sprint 2	
	Review Meeting of Sprint 2	5/5/2009 9:46 PM	System Account	Sprint 2	
<b>[-] Sprint : Sprint 3 (3)</b>					
	Planning Meeting of Sprint 3	5/5/2009 9:46 PM	System Account	Sprint 3	
	Retrospective of Sprint 3	5/5/2009 9:47 PM	System Account	Sprint 3	
	Review Meeting of Sprint 3	5/5/2009 9:47 PM	System Account	Sprint 3	
<b>[-] Sprint : Sprint 4 (3)</b>					
	Planning Meeting of Sprint 4	5/5/2009 9:47 PM	System Account	Sprint 4	
	Retrospective of Sprint 4	5/18/2009 9:10 PM	System Account	Sprint 4	
	Review Meeting of Sprint 4	5/18/2009 9:40 PM	System Account	Sprint 4	
<b>[-] Sprint : Sprint 5 (3)</b>					
	Planning Meeting of Sprint 5	5/19/2009 8:07 PM	System Account	Sprint 5	
	Retrospective of Sprint 5	6/7/2009 2:36 PM	System Account	Sprint 5	
	Review Meeting of Sprint 5	6/13/2009 3:25 PM	System Account	Sprint 5	

**Figura 48: Scrum Documents**

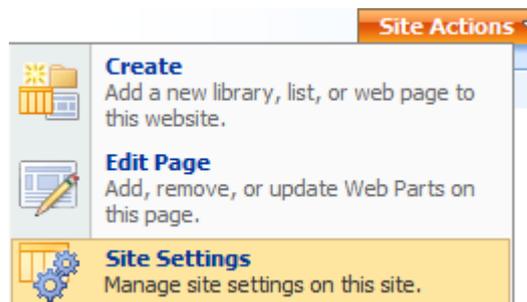
Para visualizar um documento, basta clicar em seu nome.

Existem templates para a criação de documentos do Scrum. Estes templates são mais bem explicados na seção 2.16.

## 2.13. PP2.6.2 - PERMITIR A CRIAÇÃO E ATUALIZAÇÃO DE USUÁRIOS NO SISTEMA QUE REPRESENTEM OS *STAKEHOLDERS*

A ferramenta possui três grupos de usuários que representam os principais stakeholders de um projeto Scrum: Product Owner, Scrum Master e o Time.

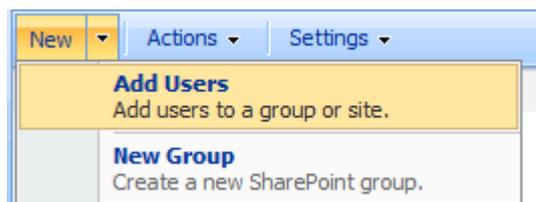
Para gerenciar os usuários e grupos do projeto, é necessário acessar Site Actions -> Site Settings no menu superior direito do site:



**Figura 49: Site Settings**

Na página Site Settings, deve-se acessar a opção People and Groups na seção Users and Permissions.

Para adicionar um novo usuário, basta clicar no botão New -> Add User:



**Figura 50: Adicionar Novo Usuário**

O formulário para adição de um novo usuário no sistema irá aparecer.

Union > Site Settings > Permissions > Add Users

## Add Users: Union

Use this page to give new permissions.

**Add Users**  
 You can enter user names, group names, or e-mail addresses. Separate them with semicolons.  
[Add all authenticated users](#)

**Give Permission**  
 Choose the permissions you want these users to have. You can add users to a SharePoint group (which is already assigned to a permission level), or you can add users individually and assign them to a specific permission level.  
 SharePoint groups are recommended as they allow for ease of permission management across multiple sites.

Users/Groups:

Give Permission

Add users to a SharePoint group

Team [Contribute, Limited Access]

[View permissions this group has on sites, lists, and items...](#)

Give users permission directly

Full Control - Has full control.

Design - Can view, add, update, delete, approve, and customize.

Contribute - Can view, add, update, and delete.

Read - Can view only.

View Only - Members of this group can view pages, list items, and documents. If the document has a server-side file handler available, they can only view the document using the server-side file handler.

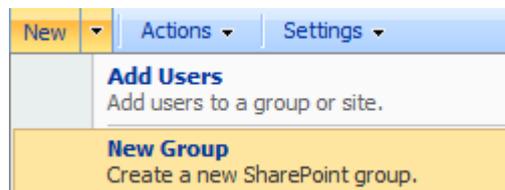
OK Cancel

**Figura 51: Formulário de Adição de um Novo Usuário**

Adicione os usuários desejados na caixa de texto e o grupo o qual este usuário pertence escolhendo uma opção na dropdown list.

Se, em algum projeto, for necessário mapear outros stakeholders que não são Product Owner, Scrum Master ou que não fazem parte do time, é possível criar outros grupos na ferramenta, como por exemplo, marketing, suporte, etc.

Para tanto, deve-se clicar em New -> Add Group:



**Figura 52: Adicionar Novo Grupo**

O formulário para adição de novos grupos irá aparecer:

Union > People and Groups > New Group

## New Group

Use this page to create a group.

<p><b>Name and About Me Description</b> Type a name and description for the group.</p>	<p>Name: <input type="text"/></p> <p>About Me: <input type="text"/></p> <p><a href="#">Click for help about adding HTML formatting.</a></p>
<p><b>Owner</b> The owner can change anything about the group such as adding and removing members or deleting the group. Only one user or group can be the owner.</p>	<p>Group owner: <input type="text" value="SHAREPOINT\system"/>  </p>
<p><b>Group Settings</b> Specify who has permission to see the list of group members and who has permission to add and remove members from the group.</p>	<p>Who can view the membership of the group?  <input checked="" type="radio"/> Group Members      <input type="radio"/> Everyone</p> <p>Who can edit the membership of the group?  <input checked="" type="radio"/> Group Owner      <input type="radio"/> Group Members</p>
<p><b>Membership Requests</b> Specify whether to allow users to request membership in this group and allow users to request to leave the group. All requests will be sent to the e-mail address specified. If auto-accept is enabled, users will automatically be added or removed when they make a request.</p> <p><b>Caution:</b> If you select yes for the Auto-accept requests option, any user requesting access to this group will automatically be added as a member of the group and receive the permission levels associated with the group.</p>	<p>Allow requests to join/leave this group?  <input type="radio"/> Yes      <input checked="" type="radio"/> No</p> <p>Auto-accept requests?  <input type="radio"/> Yes      <input checked="" type="radio"/> No</p> <p>Send membership requests to the following e-mail address:  <input type="text"/></p>
<p><b>Give Group Permission to this Site</b> Specify the permission level that you want members of this SharePoint group to have on this site. If you do not want to give group members access to this site, ensure that all checkboxes are unselected.</p> <p><a href="#">View site permission assignments</a></p>	<p>Choose the permission level group members get on this site: http://joker:25000</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Full Control - Has full control.</li> <li><input type="checkbox"/> Design - Can view, add, update, delete, approve, and customize.</li> <li><input type="checkbox"/> Contribute - Can view, add, update, and delete.</li> <li><input type="checkbox"/> Read - Can view only.</li> <li><input type="checkbox"/> View Only - Members of this group can view pages, list items, and documents. If the document has a server-side file handler available, they can only view the document using the server-side file handler.</li> </ul>

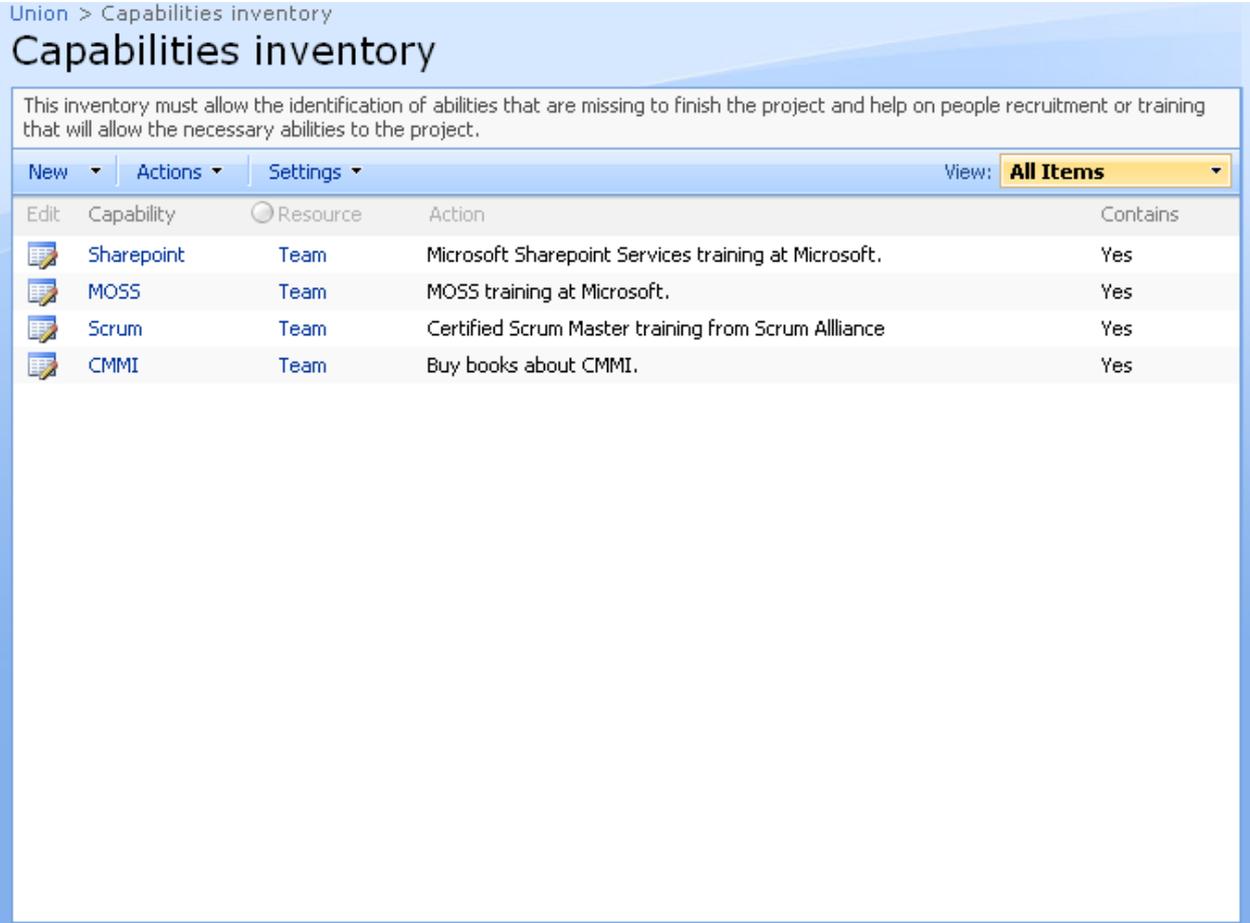
**Figura 53: Formulário de Adição de um Novo Grupo**

Forneça um nome para o grupo e adicione informações bem como as permissões que este grupo terá no sistema.

A partir disto, será possível inserir novos usuários ao novo grupo criado.

2.14. PP2.5.1 - POSSUIR UM INVENTÁRIO DE HABILIDADES NECESSÁRIAS AO PROJETO, PARA IDENTIFICAR AS HABILIDADES QUE ESTÃO FALTANDO PARA COMPLETAR O PROJETO E AUXILIAR NA CONTRATAÇÃO DE PESSOAS OU TREINAMENTOS QUE PROPORCIONARÃO AS HABILIDADES NECESSÁRIAS PARA O PROJETO

Ao acessar a opção “Capabilities inventory” do menu “Lists”, o sistema irá exibir inventário de habilidades do projeto, como pode ser visto na figura 54:



Edit	Capability	Resource	Action	Contains
	Sharepoint	Team	Microsoft Sharepoint Services training at Microsoft.	Yes
	MOSS	Team	MOSS training at Microsoft.	Yes
	Scrum	Team	Certified Scrum Master training from Scrum Alliance	Yes
	CMMI	Team	Buy books about CMMI.	Yes

Figura 54: Inventário de Habilidade

O sistema proporciona funcionalidades para criar, editar, visualizar e remover habilidades.

Para criar uma nova habilidade, é necessário clicar no botão . Neste momento o sistema irá exibir o formulário ilustrado na figura 55:

Union > Capabilities inventory > New Item

## Capabilities inventory: New Item

OK Cancel

Attach File | Spelling... \* indicates a required field

<b>Capability *</b>	<input type="text"/> Necessary capability
<b>Resource *</b>	<input type="text"/> Resource that must know this capability.  
<b>Action</b>	<div style="border: 1px solid gray; padding: 5px;">  </div> Action to be taken to allow the resource to have this capability.
<b>Contains</b>	<input type="checkbox"/> The resource already have this capability?

OK Cancel

**Figura 55: Formulário de Criação de Nova Habilidade**

O processo de criação de uma habilidade consiste em:

- Informar um nome para a habilidade.
- Informar o(s) recurso(s) que precisa(m) possuir esta habilidade.
- Informar as ações que devem ser feitas para possibilitar que o recurso possua esta habilidade.
- Informar se o recurso já contém esta habilidade.

Para editar uma habilidade existente, é necessário clicar no botão  ao lado do nome da habilidade que se deseja editar. Neste momento o sistema irá exibir o formulário ilustrado na figura 56, que permitirá a edição desta habilidade.

Union > Capabilities inventory > Sharepoint > Edit Item

## Capabilities inventory: Sharepoint

OK Cancel

Attach File | Delete Item | Spelling... \* indicates a required field

<b>Capability *</b>	<input type="text" value="Sharepoint"/> Necessary capability
<b>Resource *</b>	<input type="text" value="Team"/> Resource that must know this capability.
<b>Action</b>	<div style="border: 1px solid gray; padding: 5px;"> <p>Microsoft Sharepoint Services training at Microsoft.</p> </div> Action to be taken to allow the resource to have this capability.
<b>Contains</b>	<input checked="" type="checkbox"/> The resource already have this capability?

Version: 1.0  
 Created at 5/9/2009 4:12 PM by System Account  
 Last modified at 5/9/2009 4:12 PM by System Account

OK Cancel

**Figura 56: Formulário de Edição de uma Habilidade**

O processo de edição de uma habilidade consiste no mesmo processo que o da criação.

Para remover uma habilidade existente, é necessário acessar o formulário de edição ou visualização de habilidades. A partir de um destes formulários, para remover a habilidade, é necessário clicar no botão **Delete Item** e confirmar a remoção.

## 2.15. PP2.6.1 - POSSUIR UMA LISTA CONTENDO OS *STAKEHOLDERS* RELEVANTES E SEU ENVOLVIMENTO NO PROJETO

É possível criar um documento para mapeamento dos stakeholders do projeto. Para tanto, basta acessar a seção Management Documents no menu Documents à esquerda do portal.

No site Management Documents, basta acessar o menu New -> Stakeholder Map template and fill the document with the project's stakeholders information.

## 2.16. TC4.4 – CRIAR TEMPLATES PARA DOCUMENTOS SCRUM

A seção 2.12 comenta como acessar o repositório de documento Scrum do projeto. Esta seção aborda os 3 tipos de template criados para documentos Scrum: Planning Meeting, Review Meeting e Retrospective Meeting. Para criar um novo documento a partir de um template, basta clicar na seta ao lado de “New”.

Para salvar o documento na lista de documentos Scrum, basta dar uma “Save” no documento Word.

Abaixo, seguem informações sobre cada um dos tipos de template.

- Planning Meeting: ao clicar em “Planning Meeting”, um novo documento Word é aberto. Neste documento é possível informar:
  - A Sprint da qual a Planning Meeting faz parte.
  - A data da Planning Meeting.
  - A velocidade atual do time.
  - A quantidade em pontos de todas as User Stories , de User Stories finalizadas e de User Stories não finalizadas.
  - Um espaço para colocar as User Stories planejadas para a Sprint.
  - Espaço para comentários.
  
- Review Meeting: ao clicar em “Review Meeting”, um novo documento Word é aberto. Neste documento é possível informar:
  - A Sprint da qual a Review Meeting faz parte.
  - A data da Review Meeting.

- A velocidade do time nesta Sprint.
  - A lista de User Stories fechadas.
  - A lista de User Stories não fechadas.
  - A lista de User Stories adicionadas ao projeto durante a Sprint.
  - Os riscos adicionados.
  - Os riscos fechados.
  - Um espaço para escrever sobre escopo, tempo e custo.
  - Espaço para comentários.
- Retrospective: ao clicar em “Retrospective Meeting”, um novo documento Word é aberto. Neste documento é possível informar:
    - A Sprint da qual a Retrospective Meeting faz parte.
    - A data da Retrospective Meeting.
    - O que o time fez certo durante a Sprint?
    - O que o time deveria melhorar nas próximas Sprints?
    - Espaço para comentários.

#### 2.17. PP3.3.1 - ENVIAR UMA NOTIFICAÇÃO SOLICITANDO A APROVAÇÃO DO PLANO DESENVOLVIDO DURANTE A *PLANING MEETING* DE UMA *SPRINT*

Toda vez que um documento de Planning Meeting é criado dentro do repositório de documentos chamado “Scrum Documents”, um workflow é criado requisitando aprovação do time, do Product Owner e do Scrum Master. A figura 57 mostra que o status do workflow de aprovação para um documento que acaba de ser criado é “In progress”:

Union > Scrum Documents > Planning Meeting teste > Workflows

## Workflows: Planning Meeting teste

Use this page to start a new workflow on the current item or to view the status of a running or completed workflow.

**Start a New Workflow**

Workflows cannot be started on this document because it is currently being edited.

**Workflows**

Select a workflow for more details on the current status or history.

Name	Started	Ended	Status
<b>Running Workflows</b>			
Scrum Documents Approval	6/21/2009 12:54 PM		In Progress
<b>Completed Workflows</b>			
There are no completed workflows on this item.			

**Figura 57: Status do Workflow**

A figura 58 ilustra a lista de tarefas de aprovação criadas quando um documento de Planning Meeting é criado:

Union > Scrum Documents Approval Tasks

## Scrum Documents Approval Tasks

Task list for workflow.

New Actions Settings View: **All Tasks**

@	Title	Assigned To	Status	Priority	Due Date	% Complete	Link	Outcome
	Please approve Planning Meeting teste <span style="color: green;">NEW</span>	JOKER\Leonardo	Not Started	(2) Normal			Planning Meeting teste	
	Please approve Planning Meeting teste <span style="color: green;">NEW</span>	JOKER\nanda	Not Started	(2) Normal			Planning Meeting teste	
	Please approve Planning Meeting teste <span style="color: green;">NEW</span>	System Account	Not Started	(2) Normal			Planning Meeting teste	
	Please approve Planning Meeting teste <span style="color: green;">NEW</span>	JOKER\scrum.master	Not Started	(2) Normal			Planning Meeting teste	
	Please approve Planning Meeting teste <span style="color: green;">NEW</span>	System Account	Not Started	(2) Normal			Planning Meeting teste	
	Please approve Planning Meeting teste <span style="color: green;">NEW</span>	JOKER\product.owner	Not Started	(2) Normal			Planning Meeting teste	
	Please approve Planning Meeting teste <span style="color: green;">NEW</span>	System Account	Not Started	(2) Normal			Planning Meeting teste	

**Figura 58: Tarefas de Aprovação**

A figura 59 mostra a tela de aprovação/rejeição de uma das tarefas criadas:

Union > Scrum Documents Approval Tasks > Please approve Planning Meeting teste

## Scrum Documents Approval Tasks: Please approve Planning Meeting teste

[X Delete Item](#)

This workflow task applies to [Planning Meeting teste](#).

**Approval Requested**

---

From: System Account  
Due by:

Please approve Planning Meeting teste

Type comments to include with your response:

---

**Other options**  
[Reassign task](#)   [Request a change](#)

**Figura 59: Formulário de Aprovação ou Rejeição**

A figura 60 mostra que o status do workflow de aprovação para um documento que foi aprovado por todos é “Approved”:

Union > Scrum Documents > [Planning Meeting teste](#) > Workflows

## Workflows: Planning Meeting teste

Use this page to start a new workflow on the current item or to view the status of a running or completed workflow.

**Start a New Workflow**

Workflows cannot be started on this document because it is currently being edited.

**Workflows**

Select a workflow for more details on the current status or history.

Name	Started	Ended	Status
<b>Running Workflows</b>			
<a href="#">Scrum Documents Approval</a>	6/21/2009 12:54 PM		Approved
<b>Completed Workflows</b>			
There are no completed workflows on this item.			

**Figura 60: Documento Aprovado**

A figura 61 mostra que o status do workflow de aprovação para um documento que foi reprovado por pelo menos uma pessoa é “Rejected”:

The screenshot shows a web interface for managing workflows. At the top, there is a breadcrumb trail: "Union > Scrum Documents > Planning Meeting teste > Workflows". Below this is a yellow header with the title "Workflows: Planning Meeting teste". A message states: "Use this page to start a new workflow on the current item or to view the status of a running or completed workflow." There are two main sections: "Start a New Workflow" and "Workflows". The "Start a New Workflow" section has a message: "Workflows cannot be started on this document because it is currently being edited." The "Workflows" section has a message: "Select a workflow for more details on the current status or history." Below this is a table with columns: "Name", "Started", "Ended", and "Status".

Name	Started	Ended	Status
<b>Running Workflows</b>			
Scrum Documents Approval	6/21/2009 1:25 PM		Rejected
<b>Completed Workflows</b>			
There are no completed workflows on this item.			

**Figura 61: Documento Rejeitado**

## 2.18. PMC1.1.1 A - PROPORCIONAR A INSERÇÃO DE HORAS REALIZADAS E ATUALIZAÇÃO DAS HORAS RESTANTES NAS TAREFAS DE UMA USER STORY

É possível inserir horas realizadas e atualizar horas restantes em uma tarefa quando se edita uma. A funcionalidade PP2.1.1, na seção 2.8, mostra como criar e editar tarefas.

## 2.19. PMC2.1.1 - PROPORCIONAR O CADASTRO DE ITENS DENTRO DO IMPEDIMENTS BACKLOG, ASSOCIANDO O MEMBRO DO TIME BLOQUEADO (QUANDO HOVER), PRIORIDADE E MEMBRO QUE DEVE TOMAR ALGUMA AÇÃO PARA CORRIGIR O MESMO

Ao acessar a opção “Impediments Backlog” do menu “Lists”, o sistema irá exibir a lista de impedimentos do projeto, como pode ser visto na figura 62:

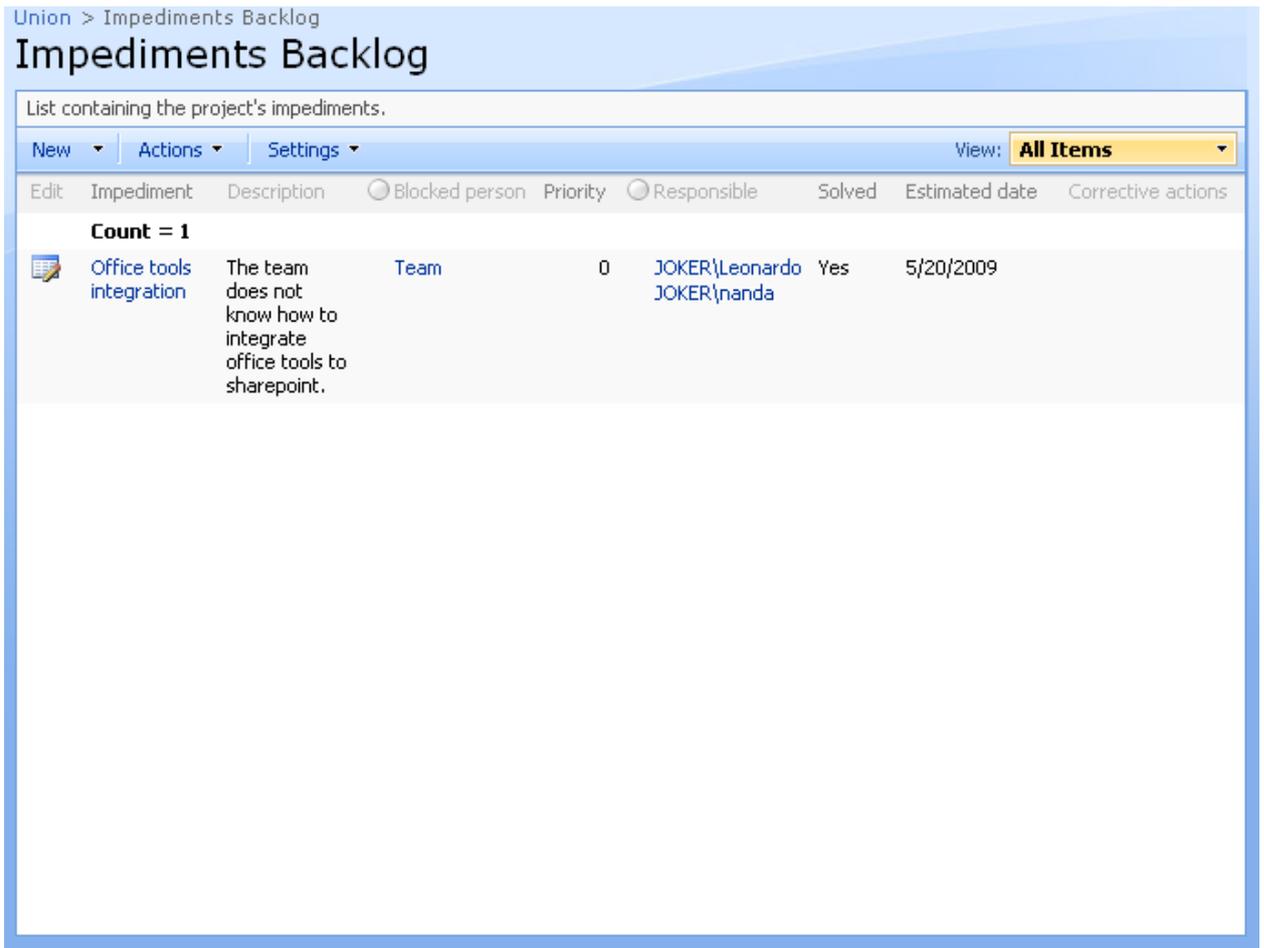


Figura 62: Impediments Backlog

O sistema proporciona funcionalidades para criar, editar, visualizar e remover impedimentos.

Para criar um novo impedimento, é necessário clicar no botão [New](#). Neste momento o sistema irá exibir o formulário ilustrado na figura 63:

Union > Impediments Backlog > New Item

## Impediments Backlog: New Item

OK Cancel

Attach File | Spelling... \* indicates a required field

<b>Impediment *</b>	<input type="text"/> Impediment's name.
<b>Description</b>	<input type="text"/>
<b>Blocked person</b>	<input type="text"/> The person who is blocked by the impediment.  
<b>Priority</b>	<input type="text"/>
<b>Responsible *</b>	<input type="text"/> The team member that must take an action to fix the impediment.  
<b>Corrective actions</b>	 <input type="text"/> Corrective actions to be taken to solve this impediment.
<b>Estimated date</b>	<input type="text" value="6/14/2009"/>  Estimated date to solve the impediment.
<b>Solved</b>	<input type="checkbox"/> Tells if the impediment was solved or not.

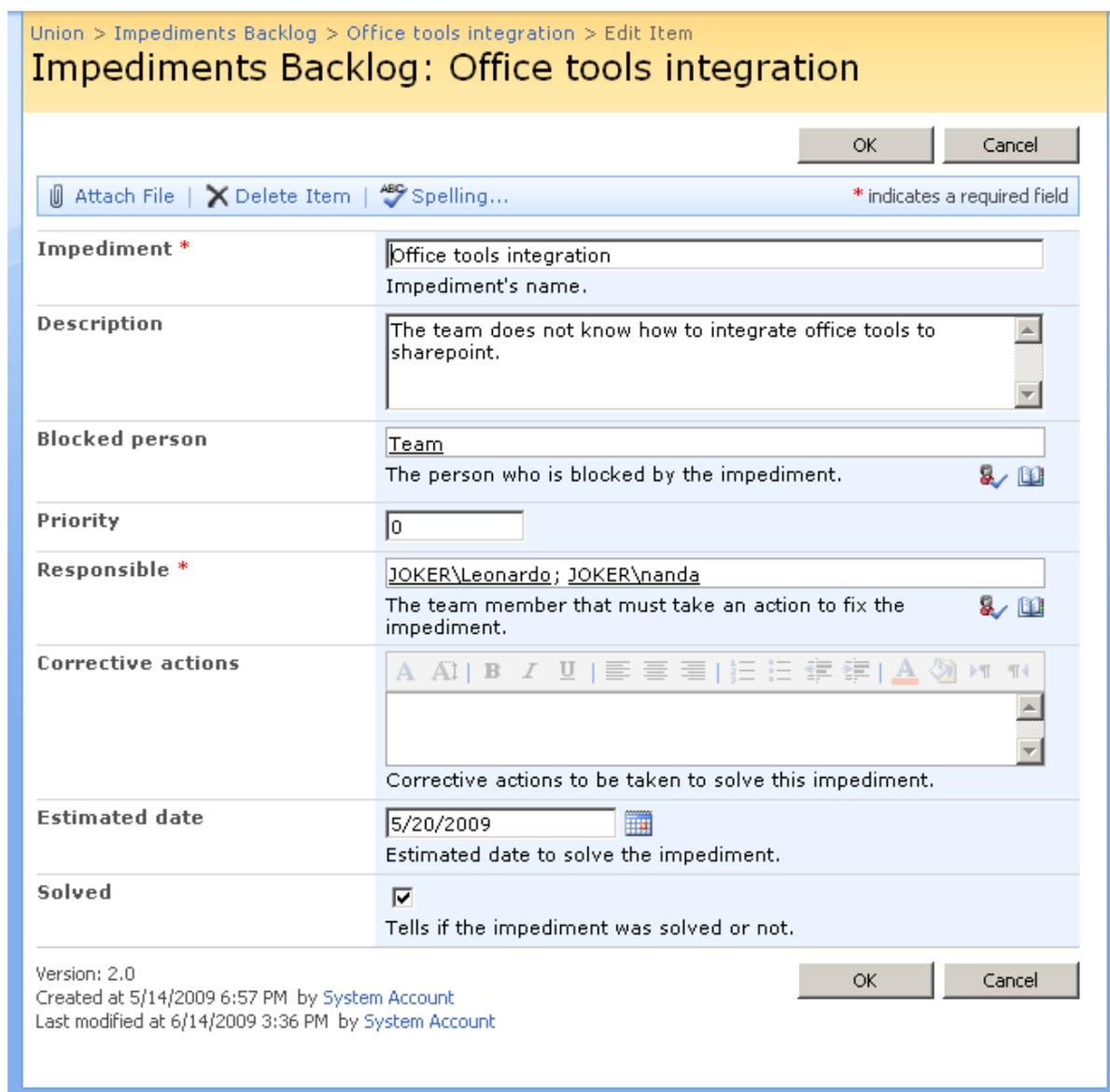
OK Cancel

**Figura 63: Formulário de Criação de um Novo Impedimento**

O processo de criação de um impedimento consiste em:

- Informar o nome do impedimento.
- Informar a descrição do impedimento.
- Informar o membro do time que está bloqueado por este impedimento.
- Informar a prioridade do impedimento.
- Informar o usuário responsável por solucionar este impedimento.
- Informar as ações a serem feitas para solucionar o impedimento.
- Informar a data estimada para a correção do impedimento. Por padrão, a data de hoje é informada.
- Informar se o impedimento já foi solucionado.

Para editar um impedimento existente, é necessário clicar no botão  ao lado do nome do impedimento que se deseja editar. Neste momento o sistema irá exibir o formulário ilustrado na figura 64, que permitirá a edição deste impedimento.



Union > Impediments Backlog > Office tools integration > Edit Item

## Impediments Backlog: Office tools integration

OK Cancel

Attach File | Delete Item | Spelling... \* indicates a required field

<b>Impediment *</b>	Office tools integration Impediment's name.
<b>Description</b>	The team does not know how to integrate office tools to sharepoint.
<b>Blocked person</b>	Team The person who is blocked by the impediment.  
<b>Priority</b>	0
<b>Responsible *</b>	JOKER\Leonardo; JOKER\nanda The team member that must take an action to fix the impediment.  
<b>Corrective actions</b>	 Corrective actions to be taken to solve this impediment.
<b>Estimated date</b>	5/20/2009  Estimated date to solve the impediment.
<b>Solved</b>	<input checked="" type="checkbox"/> Tells if the impediment was solved or not.

Version: 2.0  
Created at 5/14/2009 6:57 PM by System Account  
Last modified at 6/14/2009 3:36 PM by System Account

OK Cancel

**Figura 64: Formulário de Edição de um Impedimento**

O processo de edição de um impedimento consiste no mesmo processo que o da criação.

Para remover um impedimento existente, é necessário acessar o formulário de edição ou visualização de impedimentos. A partir de um destes formulários, para remover o impedimento, é necessário clicar no botão  Delete Item e confirmar a remoção.

2.20. PMC1.2.1 - POSSUIR UMA FUNCIONALIDADE PARA QUE, DURANTE A DAILY SCRUM, O SCRUM MASTER INSIRA, PARA CADA MEMBRO DO TIME, O QUE AQUELE MEMBRO FEZ DESDE A ÚLTIMA DAILY SCRUM, O QUE ELE PLANEJA FAZER ATÉ A PRÓXIMA DAILY SCRUM E QUAIS SÃO OS SEUS IMPEDIMENTOS.

Ao acessar a opção “Daily Scrums” do menu “Lists”, o sistema irá exibir a lista de atividades diárias de cada membro do projeto, como pode ser visto na figura 65:

Edit	Member	Date	Yesterday	Today	Impediments
Date: 6/14/2009 (2)					
	JOKER\Leonardo	6/14/2009	Finished User Story PP1.1.1	Will start the User Story PP1.2.1	<a href="#">Office tools integration</a>
	JOKER\nanda	6/14/2009	Worked on User Story PP2.1.2	Will work on User Story PP2.1.2	
Date: 6/12/2009 (2)					
	JOKER\Leonardo	6/12/2009	Worked on User Story PP1.1.1	Will work on User Story PP1.1.1	<a href="#">Office tools integration</a>
	JOKER\nanda	6/12/2009	Finished User Story PP2.1.3	Will work on User Story PP2.1.2	

Figura 65: Daily Scrums

O sistema ordena as atividades em dias, para mostrar todas as atividades do time em um determinado dia.

O sistema proporciona funcionalidades para criar, editar, visualizar e remover estas atividades.

Para criar uma nova atividade, é necessário clicar no botão . Neste momento o sistema irá exibir o formulário ilustrado na figura 66:

Union > Daily Scrums > New Item

## Daily Scrums: New Item

OK Cancel

Attach File | Spelling... \* indicates a required field

**Date \*** 6/14/2009

**Member \***

**Yesterday**

What that member made since last Daily Scrum.

**Today**

What the member plans to do until the next Daily Scrum.

**Impediments**

Office tools integratio

Add >

< Remove

What are the member's impediments.

**Obs.**

OK Cancel

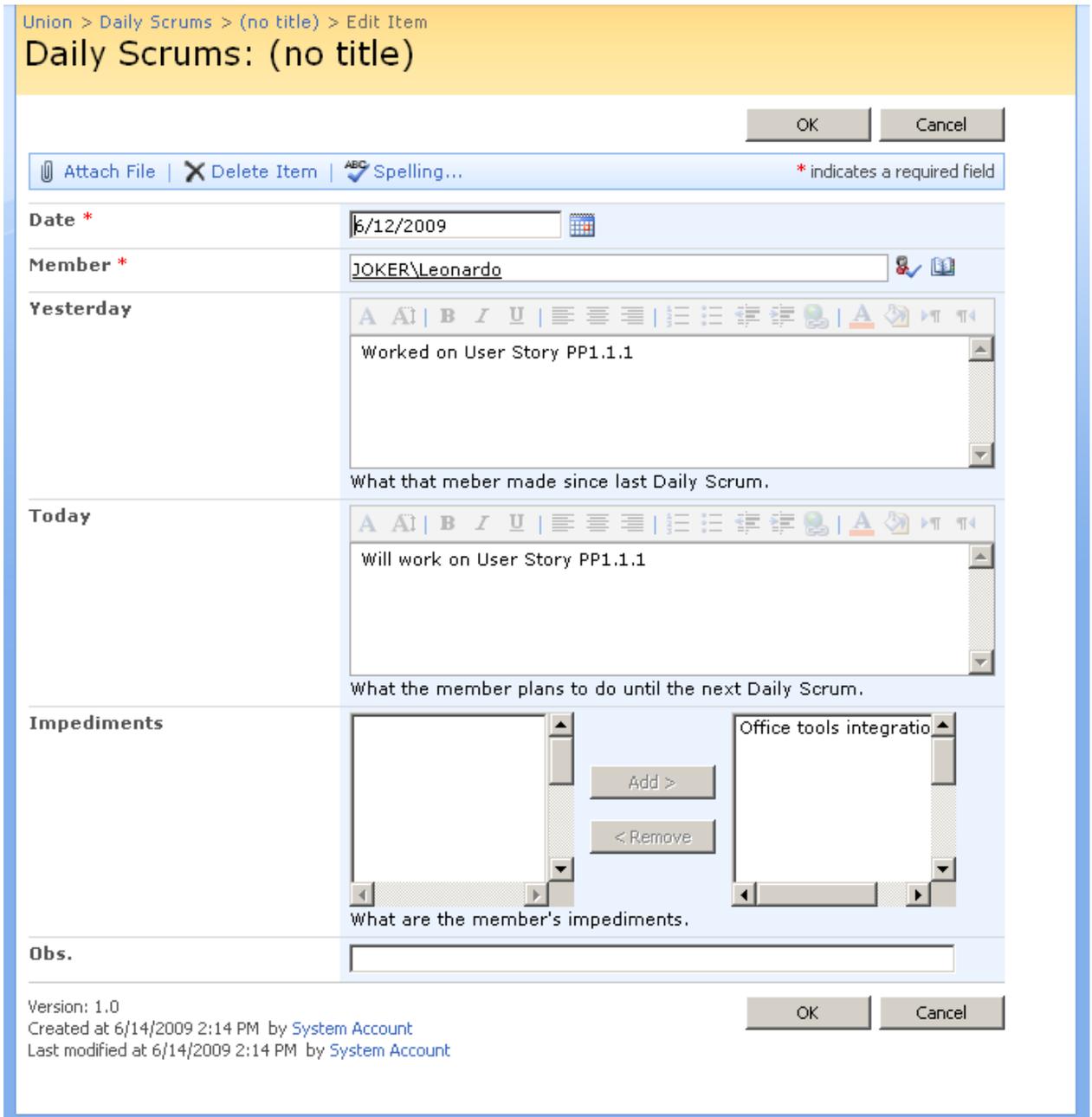
**Figura 66: Formulário de Criação de uma Nova Daily Scrum**

O processo de criação de uma atividade consiste em:

- Informar a data da Daily Scrum. Por padrão, a data é inserida pelo sistema com o dia de hoje. Este campo é obrigatório.
- Informar o membro do time que está tendo as atividades inseridas.
- O que o membro fez desde a última Daily Scrum.
- O que o membro pretende fazer até a próxima Daily Scrum.
- Quais são os impedimentos deste membro do time. Os impedimentos são buscados da lista de impedimentos mostrada na seção 2.8. Não é possível criar um impedimento nesta tela.

- Também é possível colocar uma observação sobre esta atividade.

Para editar uma atividade existente, é necessário clicar no botão  ao lado do nome da atividade que se deseja editar. Neste momento o sistema irá exibir o formulário ilustrado na figura 67, que permitirá a edição desta atividade.



Union > Daily Scrums > (no title) > Edit Item

## Daily Scrums: (no title)

OK Cancel

Attach File | Delete Item | Spelling... \* indicates a required field

**Date \*** 6/12/2009

**Member \*** JOKER\Leonardo

**Yesterday**

Worked on User Story PP1.1.1

What that member made since last Daily Scrum.

**Today**

Will work on User Story PP1.1.1

What the member plans to do until the next Daily Scrum.

**Impediments**

Office tools integratio

Add >

< Remove

What are the member's impediments.

**Obs.**

Version: 1.0  
Created at 6/14/2009 2:14 PM by System Account  
Last modified at 6/14/2009 2:14 PM by System Account

OK Cancel

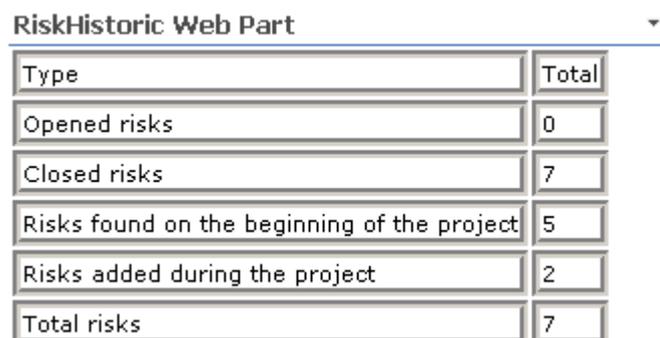
Figura 67: Formulário de Edição de uma Daily Scrum

O processo de edição de uma atividade consiste no mesmo processo que o da criação.

Para remover uma atividade existente, é necessário acessar o formulário de edição ou visualização de atividades. A partir de um destes formulários, para remover a atividade, é necessário clicar no botão  e confirmar a remoção.

## 2.21. PMC1.3.1 - GUARDAR UM HISTÓRICO DE RISCOS PARA MOSTRAR A QUANTIDADE DE RISCOS ENCONTRADOS NO INÍCIO DO PROJETO, A QUANTIDADE DE RISCOS FECHADOS, OS RISCOS QUE FORAM ADICIONADOS DURANTE O PROJETO E A QUANTIDADE DE RISCOS ABERTOS

A Web Part “Risk Historic”, que pode ser visualizada na aba , informa o histórico de riscos do projeto. A figura 68 ilustra esta funcionalidade:



Type	Total
Opened risks	0
Closed risks	7
Risks found on the beginning of the project	5
Risks added during the project	2
Total risks	7

**Figura 68: Risk History Web Part**

A Web Part mostra a quantidade de riscos abertos e fechados, a quantidade de riscos encontrados antes e depois do início do projeto e a quantidade total de riscos.

## 2.22. PMC1.1.1B - ATUALIZAR O ITERATION BURNDOWN CHART AUTOMATICAMENTE APÓS CADA MUDANÇA EM UMA TAREFA

O burndown chart do projeto encontra-se na página principal:



Figura 69: Burndown Chart

O Burndown chart default é da sprint corrente ou da última sprint do projeto. É possível visualizar charts de outras sprints apenas selecionando a sprint desejada no dropdown list. Além disso, é possível realizar um Update do gráfico sempre que desejar, apenas clicando no botão Update Chart.

## 2.23. TC4.3 – INSERIR UM STATUS PARA CADA TAREFA DE UMA USER STORY.

Os estados possíveis são: “New”, “Assigned” e “Closed”.

A funcionalidade PP2.1.1, na seção 2.8, mostra como inserir e editar tarefas.

#### 2.24. PMC1.4.1 – PROPORCIONAR O VERSIONAMENTO DO PLANO DE GERENCIAMENTO DE DADOS COM AS REVISÕES DE ALTERAÇÃO E VISUALIZAÇÃO FEITAS PELO *SCRUM MASTER*

Ao acessar a opção “Management Documents” do menu “Documents”, o sistema irá exibir o plano de gerenciamento de dados do projeto.

Assim como os demais documentos e listas, o plano de gerenciamento de dados está versionado. A seção 2.30 explica como visualizar, editar e remover versões de documentos e listas.

#### 2.25. PMC1.5.1 – MOSTRAR UMA LISTA DAS DAILY SCRUMS FEITAS NO DECORRER DO PROJETO PARA QUE ISTO POSSA SER UTILIZADO COMO EVIDÊNCIA DE REGISTRO DO ENVOLVIMENTO DOS *STAKEHOLDERS*

A funcionalidade PMC1.2.1, na seção 2.20, mostra como visualizar a lista de Daily Scrums.

#### 2.26. PMC1.6.1 – POSSUIR UM REPOSITÓRIO PARA QUE SEJAM COLOCADOS DOCUMENTOS CRIADOS DURANTE A REUNIÃO DE REVIEW, COMO UMA ATA OU UM VÍDEO COM A DEMONSTRAÇÃO DO INCREMENTO FUNCIONAL

As funcionalidades TC4.1 e TC4.4, nas suas respectivas seções 2.12 e 2.16, mostram como criar documentos para a Review Meeting.

#### 2.27. PMC2.2.1 – PERMITIR A INSERÇÃO DE UMA DATA ESTIMADA PARA CORREÇÃO E AÇÕES CORRETIVAS A SEREM TOMADAS DENTRO DE CADA IMPEDIMENTO NO *IMPEDIMENTS BACKLOG*

É possível inserir uma data estimada para correção e ações corretivas a serem tomadas nos impedimentos quando se cria ou edita um. A funcionalidade PMC2.1.1, na seção 2.19, mostra como criar e editar impedimentos no *Impediments Backlog*.

## 2.28. PMC2.3.1 – ARMAZENAR TODOS OS IMPEDIMENTOS JÁ RESOLVIDOS NO IMPEDIMENT *BACKLOG*

É possível marcar um impedimento como resolvido quando se cria ou edita um impedimento. A funcionalidade PMC2.1.1, na seção 2.19, mostra como criar e editar impedimentos no Impediments *Backlog*.

## 2.29. PP2.3.2 - HABILITAR O GERENCIAMENTO DO *PRODUCT BACKLOG* APENAS PELO O *PRODUCT OWNER* E A VISUALIZAÇÃO DO MESMO PARA TODO O TIME

Somente um usuário do grupo “Product Owner” pode criar, editar e remover itens do Product Backlog, demais usuários apenas podem visualizar a lista e seus itens.

A figura 70 ilustra o Product Backlog visualizado pelo Product Owner:

The screenshot shows the 'Product Backlog' interface. The top navigation bar includes 'Union' and 'Welcome JOKER\product.owner'. The main content area displays 'Product Backlog' with a table of items. The table has columns for Edit, Identifier, Description, Priority, Details, Acceptance Tests, Points, Completed, and Sprint. The table is grouped into 'Sprint : Sprint 1 (1)' and 'Sprint : Sprint 2 (6)'. The left sidebar contains navigation options like 'Documents', 'Lists', 'Discussions', and 'Sites'.

Edit	Identifier	Description	Priority	Details	Acceptance Tests	Points	Completed	Sprint
						<b>Sum = 170</b>	<b>Count = 36</b>	
<b>Sprint : Sprint 1 (1)</b>								
	PP2.1.3	Allow the creation of Sprints, with start and end dates.	1	Allow the creation of Sprints, containing its name, a start date and an end date.		3	Yes	Sprint 1
						<b>Sum = 3</b>	<b>Count = 1</b>	
<b>Sprint : Sprint 2 (6)</b>								
	PP1.1.1	Provide a template for creation of User Stories, with a simple and practical way to maintain.	2	<ul style="list-style-type: none"><li>Create a template for User Stories's creation, containing:<ul style="list-style-type: none"><li>Name</li><li>Description</li><li>Priority</li><li>Details</li><li>Acceptance tests.</li></ul></li><li>It must be possible to create, update, remove and visualize the User Stories.</li></ul>	<ul style="list-style-type: none"><li>Only the name must be required.</li><li>No stories with same priority.</li><li>Priority only accept number values.</li></ul>	5	Yes	Sprint 2
						<b>Sum = 24</b>	<b>Count = 6</b>	

Figura 70: Product Backlog Visualizado pelo Product Owner

A figura 71 ilustra o Product Backlog visualizado por demais usuários, onde não é possível criar ou editar User Stories.

The screenshot shows the Union Product Backlog interface. The top navigation bar includes 'Home' and 'Checklists'. The main content area is titled 'Product Backlog' and displays 'All User Stories of the Product'. A table lists user stories with columns for 'Edit', 'Identifier', 'Description', 'Priority', 'Details', 'Acceptance Tests', 'Points', 'Completed', and 'Sprint'. The table is grouped into two sprints: 'Sprint : Sprint 1 (1)' and 'Sprint : Sprint 2 (6)'. The first sprint contains one user story (PP2.1.3) with a priority of 1, 3 points, and is marked as 'Yes' completed. The second sprint contains one visible user story (PP1.1.1) with a priority of 2, 5 points, and is marked as 'Yes' completed. The interface also features a left sidebar with navigation options like 'Documents', 'Lists', 'Discussions', and 'Sites', and a top search bar.

Edit	Identifier	Description	Priority	Details	Acceptance Tests	Points	Completed	Sprint
						<b>Sum = 170</b>	<b>Count = 36</b>	
<b>Sprint : Sprint 1 (1)</b>								
	PP2.1.3	Allow the creation of Sprints, with start and end dates.	1	Allow the creation of Sprints, containing its name, a start date and an end date.		3	Yes	Sprint 1
						<b>Sum = 3</b>	<b>Count = 1</b>	
<b>Sprint : Sprint 2 (6)</b>								
	PP1.1.1	Provide a template for creation of User Stories, with a simple and practical way to maintain.	2	<ul style="list-style-type: none"> <li>Create a template for User Stories's creation, containing: <ul style="list-style-type: none"> <li>Name</li> <li>Description</li> <li>Priority</li> <li>Details</li> <li>Acceptance tests.</li> </ul> </li> <li>It must be possible to create, update, remove and visualize the User Stories.</li> </ul>	<ul style="list-style-type: none"> <li>Only the name must be required.</li> <li>No stories with same priority.</li> <li>Priority only accept number values.</li> </ul>	5	Yes	Sprint 2
						<b>Sum = 24</b>	<b>Count = 6</b>	

Figura 71: Product Backlog Visualizado pelos outros usuários

## 2.30. PP3.1.1 – PERMITIR A VISUALIZAÇÃO DE UMA DETERMINADA VERSÃO DE UM PLANO

Todos os documentos e listas são versionados e suas versões podem ser vistas, assim como todos os planos do projeto. A funcionalidade PP2.3.1, na seção 2.10, mostra como visualizar uma versão de um documento.

2.31. PP1.1.3 – REALIZAR O VERSIONAMENTO DO *PRODUCT BACKLOG* PARA QUE AS INFORMAÇÕES NELE CONTIDAS DESDE O INÍCIO DO PROJETO ESTEJAM DISPONÍVEIS DURANTE TODO SEU CICLO DE VIDA

Ao acessar a opção “Product Backlog” do menu “Lists”, o sistema irá exibir as User Stories do projeto.

Assim como os demais documentos e listas, o Product Backlog está versionado. A seção 2.30 explica como visualizar, editar e remover versões de documentos e listas