



ANDERSON ABNER DE CARVALHO

**SCRUMMPS 2.0 - EVOLUÇÃO DE UMA
FERRAMENTA INTERATIVA PARA SUPORTE
AO SCRUM E MPS.BR**

**LAVRAS - MG
2013**

ANDERSON ABNER DE CARVALHO

**SCRUMMPS 2.0 - EVOLUÇÃO DE UMA FERRAMENTA INTERATIVA
PARA SUPORTE AO SCRUM E MPS.BR**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração:
Engenharia de Software

Coorientador:
Msc. Geovane Nogueira Lima

Orientador:
Prof. Dr. Heitor Augustus Xavier Costa

**LAVRAS - MG
2013**

**Ficha Catalográfica preparada pela Divisão de Processo Técnico da
Biblioteca Central da UFLA**

Carvalho, Anderson Abner de

Evolução de uma Ferramenta Interativa para Suporte ao Scrum E
MPS.BR / Anderson Abner de Carvalho. Lavras - Minas Gerais, 2013. XXX p.

Monografia de Graduação - Universidade Federal de Lavras.
Departamento de Ciência da Computação.

1. Scrum. 2. Mps.Br. 3. Gerência de Projetos. I. CARVALHO, A. A. de.
II. Universidade Federal de Lavras. III. Evolução de uma Ferramenta Interativa
para Suporte ao Scrum e Mps.Br.

ANDERSON ABNER DE CARVALHO

**SCRUMMPS 2.0 - EVOLUÇÃO DE UMA FERRAMENTA INTERATIVA
PARA SUPORTE AO SCRUM E MPS.BR**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

APROVADA em 22 de abril de 2013.

Prof. Msc. Antônio Elizeu da Rocha Neto UFLA

Esp. Ramon Simões Abílio UFLA

Msc. Geovane Nogueira Lima
Coorientador



Prof. Dr. Heitor Augustus Xavier Costa
Orientador

**LAVRAS - MG
2013**

*“Para meus pais, Wilson e Elane, e meus irmãos,
Fabrissa e Luiz André, por sempre acreditarem e
me incentivarem no alcance dos meus objetivos”.*

AGRADECIMENTOS

Agradeço primeiramente a Deus.

À Universidade Federal de Lavras (UFLA) e ao Departamento de Ciência da Computação (DCC), pela oportunidade de realização desse curso e por me guiarem na busca por conhecimento.

Ao meu orientador, Prof. Dr. Heitor A. X.Costa, por sempre me orientar da melhor forma possível, pela paciência e pelo grande aprendizado que tive durante a realização deste trabalho.

Ao meu amigo e coorientador Geovane Nogueira Lima, pelo acompanhamento do trabalho, principalmente pelas críticas, comentários e sugestões visando sempre contribuir.

Ao meu colega Thiago Cavalcanti Reis, por todas as vezes que me ajudou a resolver questões técnicas da ferramenta ScrumMps.

A todos os entrevistados, pelo tempo dispensado e pela rica contribuição.

A todos os meus amigos que me ajudaram de alguma forma a realizar este trabalho.

A minha família. Agradeço a todos vocês, que são os responsáveis por me fazer chegar aqui.

RESUMO

A gerência de projetos é importante para as atividades serem executadas seguindo um planejamento. Ferramentas têm sido propostas para apoiar essa gerência, seja ágil ou tradicional. Scrum é uma metodologia ágil para gerência de projetos baseada em pequenas iterações chamadas *Sprint*. Ainda que metodologias ágeis primam por tornar os processos mais ágeis, a qualidade não é esquecida, pois a qualidade de um produto conquista clientes e aumenta a participação da empresa em um mercado altamente competitivo. Essa qualidade pode ser obtida por meio de melhoria do processo que desenvolve o produto, por exemplo, o MPS.BR, modelo brasileiro de melhoria de processo de software. Neste trabalho, o foco é em práticas documentadas em alguns processos dos Níveis de maturidade F e G do MPS.BR e o objetivo é apresentar a evolução da ferramenta computacional “ScrumMps 2.0” para auxiliar a gerência de projetos de software que utilizam Scrum. Nessa ferramenta, é apresentada a integração de Scrum e MPS.Br, permitindo, por exemplo, acompanhar tarefas e atividades do projeto, gerenciar usuários e histórias de usuários, interagir com equipe de desenvolvimento e abordar conceitos de MPS.Br. A ScrumMps 2.0 foi avaliada por profissionais da área de Computação e discentes dos cursos de graduação em Ciência da Computação e Sistemas de Informação que atuam no mercado. Essa avaliação foi realizada fornecendo um roteiro aos participantes para serem guiados pela ferramenta e um questionário a ser respondido por eles. No resultado obtido, pôde-se perceber que a ScrumMps 2.0 atingiu o objetivo: contribui com a aplicação conjunta de processos do MPS.BR e Scrum.

Palavras-chave: Gerência de Projetos, Scrum, MPS.BR

ABSTRACT

Project management is important to carry out activities following a planning. Some tools were proposed for supporting this management, agile or traditional one. Scrum is an agile methodology for project management based on small iterations (Sprints). Although agile methodologies focuses on more agile processes, quality is not forgotten because products with quality will conquest clients and will augment the participation of the organizations in a highly competitive market. This quality can be reached through process improvement that developing the product, for example, MPS.BR. In this monograph, the objective is to present improvements of a computational tool - ScrumMps 2.0 - to aid Scrum software project management. In ScrumMps 2.0, integration between Scrum and MPS.BR is shown through new functions, such as, to supervise project tasks and activities, to manage user and user storycards, to interact with development team, and to deal for MPS.BR concepts. ScrumMps 2.0 was evaluated by professional of Computation area, and Computer Science and Information System students. This evaluation was carried out for following a script by participants and a questionnaire. The result indicated that ScrumMps 2.0 reached the objective: To contribute for combining MPS.BR and Scrum.

Keywords: Project Management, Scrum, MPS.BR

SUMÁRIO

1.	INTRODUÇÃO.....	11
1.1.	Motivação.....	12
1.2.	Objetivo.....	13
1.3.	Metodologia de Desenvolvimento.....	14
1.3.1.	Tipos de Pesquisa.....	14
1.3.2.	Procedimentos Metodológicos.....	15
1.4.	Estrutura do Trabalho.....	16
2.	METODOLOGIAS ÁGEIS.....	18
2.1.	Considerações Iniciais.....	18
2.2.	Origem dos Métodos Ágeis.....	18
2.3.	Scrum.....	21
2.3.1.	Elementos de Apoio.....	23
2.3.2.	Time Scrum.....	24
2.3.3.	Fluxo do Scrum.....	26
2.3.3.1.	Vision.....	26
2.3.3.2.	Sprint.....	27
2.3.3.3.	Daily Scrum.....	28
2.3.3.4.	Revisão da Sprint.....	28
2.3.3.5.	Retrospectiva.....	29
2.3.4.	Práticas Ágeis Utilizadas com Scrum.....	30
2.3.4.1.	Planning Poker.....	30
2.3.4.2.	User Stories.....	31
2.3.4.3.	Gráficos Burndown.....	32
2.3.4.4.	Kanban.....	33
2.4.	Considerações Finais.....	34
3.	QUALIDADE DE SOFTWARE.....	35
3.1.	Considerações Iniciais.....	35
3.2.	Conceitos.....	35
3.3.	Qualidade de Processo.....	37
3.4.	MPS.BR.....	39
3.4.1.	Níveis de Maturidade.....	41
3.4.2.	Níveis Iniciais do MPS.BR (Níveis G e F).....	42
3.4.2.1.	Nível G.....	42
3.4.2.2.	Nível F.....	46
3.5.	Considerações Finais.....	49
4.	FERRAMENTAS EXISTENTES.....	50
4.1.	Considerações Iniciais.....	50
4.2.	PangoScrum.....	50
4.3.	IceScrum.....	54
4.4.	MeuScrum.....	58

4.5.	FireScrum	60
4.6.	Quadro Comparativo	64
4.7.	Considerações Finais	65
5.	SCRUMMPS 1.0	66
5.1.	Considerações Iniciais	66
5.2.	Funcionalidade	66
5.3.	Principais Funções	68
5.4.	Utilização do MPS.BR	70
5.5.	Considerações Finais	70
6.	SCRUMMPS 2.0	72
6.1.	Considerações Iniciais	72
6.2.	Modelagem e Arquitetura	72
6.2.1.	Diagrama de Casos de Uso	72
6.2.2.	Modelagem de Dados	73
6.2.3.	Diagrama Arquitetural	75
6.3.	Projeto Detalhado e Implementação	76
6.3.1.	Gerência de Portfólio de Projetos	76
6.3.2.	Bug Tracker	78
6.3.3.	Gerência de Riscos	78
6.3.4.	Gráfico Burndown	79
6.3.5.	Gerência de Reuniões	80
6.3.6.	Outras Funções	81
6.4.	Utilização do MPS.BR	82
6.5.	Quadro Comparativo entre ScrumMps 2.0 e Ferramentas Analisadas 84	
6.6.	Considerações Finais	85
7.	AVALIAÇÃO	86
7.1.	Considerações Iniciais	86
7.2.	Caracterização dos Respondentes	87
7.3.	Apoio aos Processos do MPS.BR	89
7.4.	Apoio às Práticas do Scrum	90
7.5.	Usabilidade	92
7.6.	Objetivo	93
7.7.	Considerações Finais	94
8.	CONSIDERAÇÕES FINAIS	96
8.1.	Conclusões	96
8.2.	Contribuições	97
8.3.	Limitações da ScrumMps 2.0	98
8.4.	Trabalhos Futuros	99
	REFERÊNCIAS BIBLIOGRÁFICAS	101

1. INTRODUÇÃO

As empresas necessitam cada vez mais gerenciar projetos de software flexíveis para obter maior agilidade na entrega de produtos que atendam a seus clientes. Assim, algumas empresas adotam a filosofia ágil no gerenciamento de projetos, visando à entrega de sistemas de software ao cliente com rapidez, eficácia e eficiência [Oliveira; Lima, 2011]. Um exemplo dessa filosofia é o Scrum, utilizado para gerenciar o desenvolvimento de produtos complexos [Schwaber; Sutherland, 2011]. Um de seus focos é gerenciar projetos sem determinar como a equipe executa as tarefas de programação. Visando à qualidade dos sistemas de software, as organizações têm investido na melhoria dos seus processos de desenvolvimento para o seu setor de software ser competitivo, nacional e internacional, conforme padrões de qualidade.

A necessidade de introduzir qualidade no processo de produção de software levou ao estabelecimento de normas e modelos de qualidade para aumentar, medir e garantir a qualidade de software [Silva *et al*, 2009]. Em 2003, o modelo de melhoria de processos de desenvolvimento (MPS.BR) foi criado no Brasil, cujo objetivo é estabelecer um caminho economicamente viável para organizações, incluindo pequenas e médias empresas, alcançarem os benefícios da melhoria de processos e da utilização de boas práticas da engenharia de software em um intervalo de tempo razoável [Travassos; Kalinowski, 2012]. Existem ferramentas que auxiliam as empresas a obter um melhor gerenciamento dos seus processos de software.

Em 2011, a ferramenta ScrumMps [Reis, 2011] foi criada (ScrumMps 1.0), cujo objetivo foi unir conceitos da metodologia ágil Scrum e processos do MPS.BR. A ideia de unir esses conceitos foi criar uma ferramenta para aumentar a qualidade do software brasileiro alinhado às tendências do mercado. As

principais funções desenvolvidas foram: i) Manutenção do *Product Backlog*; ii) Planejamento das *Sprints*; e iii) Gerenciamento das tarefas. Essas funções apoiam principalmente o processo Gerência de Projetos no Nível G do MPS.BR. Diante disso, este trabalho apresenta a evolução da ferramenta ScrumMps para a Versão 2.0 (ScrumMps 2.0) desenvolvendo novas funções para auxiliar as práticas de Scrum e processos do MPS.BR não contemplados na primeira versão.

1.1. Motivação

O software tem estado cada vez mais presente no cotidiano das pessoas, porém a maioria dos projetos de software não atende aos objetivos traçados [Reed, 2000 *apud* Rincon, 2009]. Isso decorre da falta de processos adequados nas organizações em que eles são desenvolvidos [Rincon, 2009]. Para entregar sistemas de software com alta qualidade, as empresas têm adotado modelos como o MPS.BR que visam a melhoria da qualidade dos processos de desenvolvimento. Juntamente com a melhoria de processos a partir do ano 2000 surgiu uma tendência para o desenvolvimento ágil de sistemas de software devido a um ritmo acelerado de mudanças e inovações na tecnologia da informação, em organizações e no ambiente de negócios. Implementações bem sucedidas de métodos ágeis têm sido relatadas na literatura [Mahnic, 2011]. Na pesquisa *Agile Adoption Rate* [Ambler, 2008] realizada pelo Dr. Dobbs *Journal*, equipes ágeis relatam melhorias significativas na produtividade, na qualidade e na satisfação dos *stakeholders* e melhorias razoáveis de custo. Em outra pesquisa [VersionOne, 2010], é sinalizado que Scrum é a metodologia ágil mais difundida na indústria, sendo utilizada por 58% das organizações entrevistadas (Figura 1).

Algumas empresas utilizam a metodologia ágil Scrum como base para conseguir implementar o modelo MPS.BR, melhorando a qualidade de seus processos e produtos e obtendo mais produtividade das suas equipes [Catunda *et*

al., 2011; Szimanski *et al.*, 2009; Oliveira *et al.*, 2007]. Porém, no mercado, percebe-se falta de ferramentas computacionais gratuitas para gerenciar projetos que apoiam as abordagens Scrum e MPS.BR em conjunto.

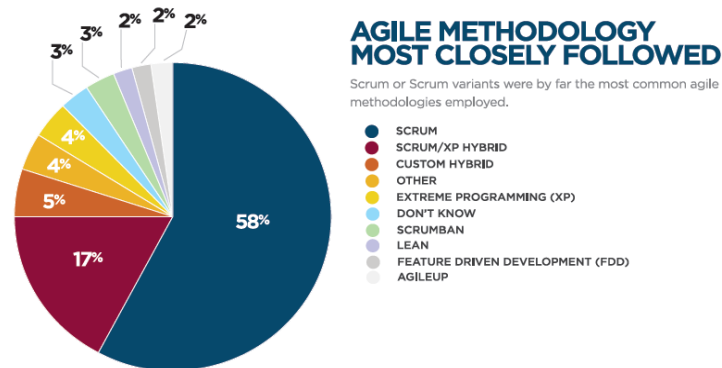


Figura 1 - Metodologias Ágeis Mais Utilizadas (Fonte: [VersionOne, 2010])

1.2. Objetivo

O objetivo deste trabalho é realizar a reengenharia e evoluir a ferramenta interativa para gestão de projetos ScrumMps (chamada a partir desse momento de ScrumMps 1.0). Na nova versão, ScrumMps 2.0, serão abordados conceitos, atributos de processos e resultados de atributos de processos sugeridos nos níveis G e F do MPS.BR sob a luz da metodologia ágil Scrum. Os objetivos específicos são:

- Aprofundar conhecimento na metodologia ágil Scrum;
- Aprofundar conhecimento no MPS.BR;
- Entender o funcionamento do ScrumMps 1.0;
- Realizar a engenharia reversa do ScrumMps 1.0;
- Identificar correções na ScrumMps 1.0 e melhorias para a ScrumMps 2.0.

1.3. Metodologia de Desenvolvimento

A metodologia de pesquisa é um conjunto de métodos, técnicas e procedimentos cuja finalidade é viabilizar a execução da pesquisa que tem como resultado um novo produto, processo ou conhecimento [Jung, 2009].

1.3.1. Tipos de Pesquisa

Uma pesquisa pode ser classificada em (Figura 2) [Jung, 2009]:

- Quanto a Natureza: i) Pesquisa Básica (gerar conhecimento sem finalidades de aplicação) e ii) Pesquisa Aplicada (gerar conhecimento com finalidades de aplicação);
- Quanto aos Objetivos: i) Exploratória (descobrir/innovar); ii) descritiva (como?); e iii) explicativa (por que?);
- Quanto as Abordagens: i) Quantitativa e ii) Qualitativa;
- Quanto aos Procedimentos: i) *Survey*; ii) Pesquisa-Ação; iii) Estudo de Caso Único ou Múltiplos; iv) Operacional; e v) Experimental.

Além disso, os métodos para a coleta dos dados podem ser por meio de (i) observação do participante, (ii) grupos focados, (iii) entrevistas, (iv) questionário, (v) experimentação e (vi) observação.

Quanto a natureza, este trabalho classifica-se como **pesquisa aplicada**, pois são empregados conhecimentos de gerência de projeto e qualidade de software, cujo resultado tem como finalidade ser utilizado em organizações do cenário brasileiro. Quanto aos seus objetivos, o trabalho caracteriza-se como **pesquisa exploratória**, pois visa o desenvolvimento de uma ferramenta computacional para apoiar atividades de gerenciamento de projeto nas organizações. Quanto à sua abordagem, este trabalho é uma **pesquisa quantitativa e qualitativa**. A pesquisa é quantitativa, pois as respostas das

"perguntas fechadas" de um questionário foram compiladas em números para viabilizar a análise dos dados. A pesquisa é qualitativa, pois a aplicação de um questionário teve caráter exploratório, uma vez que estimulou o respondente a pensar e a expressar-se livremente sobre a ferramenta computacional desenvolvida por meio de "perguntas abertas"; na análise da resposta dessas perguntas, foram considerados aspectos como as opiniões e os comentários dos respondentes. Quanto aos procedimentos, este trabalho é caracterizado como uma **pesquisa survey**, pois foram obtidas opiniões de um determinado grupo de pessoas, por meio de um questionário. O método utilizado para coleta de dados foi **experimentação** do software desenvolvido.

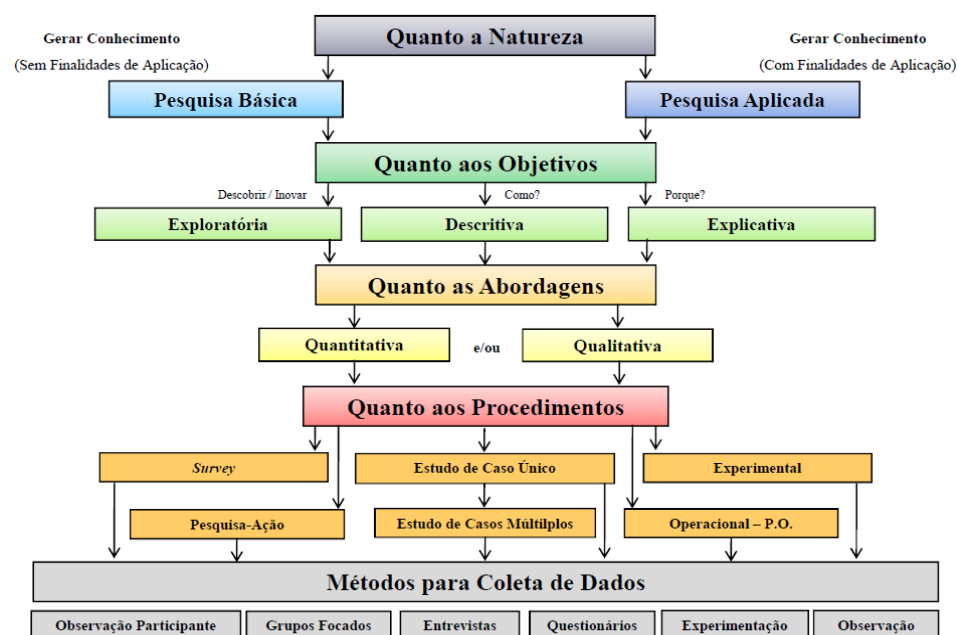


Figura 2 - Tipos de Pesquisa Científica [Jung, 2009]

1.3.2. Procedimentos Metodológicos

Os estudos foram iniciados realizando um levantamento bibliográfico por meio de pesquisa na Internet em bibliotecas digitais e impressas de artigos científicos relacionados ao tema do trabalho. Paralelamente, foi feito um estudo

da ScrumMps 1.0 realizando a engenharia reversa e a reengenharia da ferramenta para identificar correções e melhorias para a ScrumMps 2.0.

A etapa seguinte consistiu da análise dos principais conceitos e técnicas da metodologia ágil Scrum e dos níveis G e F do modelo MPS.BR. Foi feito um estudo de ferramentas de acesso livre existentes no mercado para gerência de projetos utilizando a metodologia ágil Scrum para identificar requisitos para serem contemplados na ScrumMps 2.0.

Posteriormente, foram estudadas as linguagens de programação Adobe Flex e Java para serem utilizadas no desenvolvimento de novos recursos da ScrumMps 2.0, tendo em vista que a sua primeira versão as utilizou. Como instrumento de validação dos trabalhos, um questionário foi elaborado e aplicado a profissionais da área de Engenharia de Software com o intuito de avaliar os novos recursos desenvolvidos na ScrumMps 2.0.

Durante o desenvolvimento deste trabalho, houve reuniões semanais com o orientador e coorientador para nortear os passos seguintes e corrigir desvios na realização dos passos anteriores.

1.4. Estrutura do Trabalho

Este trabalho está organizado da seguinte forma.

A origem dos métodos ágeis e breve abordagem da metodologia ágil de gerência de projetos Scrum são apresentadas no Capítulo 2.

Definições de qualidade de software e de qualidade de processos e sucinta descrição das normas ISO/IEC 12207:2008 e ISO/IEC 15504 e dos modelos CMMI-DEV e MPS.BR são destacadas no Capítulo 3.

Algumas ferramentas de acesso livre existentes no mercado para gerência de projetos utilizando a metodologia ágil Scrum são analisadas no Capítulo 4 visando identificar requisitos para a ScrumMps 2.0.

Estudo da primeira versão da ScrumMps é mostrado no Capítulo 5.

Recursos desenvolvidos na ScrumMps 2.0, apresentando o Diagrama de Casos de Uso, o Modelo Arquitetural e as novas funções, são descritos no Capítulo 6.

Avaliação da ScrumMps 2.0, utilizada por acadêmicos e profissionais da área de Engenharia de Software é discutida no Capítulo 7.

Conclusões, contribuições, limitações, sugestões de trabalhos futuros são apresentadas no Capítulo 8.

2. METODOLOGIAS ÁGEIS

2.1. Considerações Iniciais

A expressão "Metodologia Ágil" tornou-se conhecida em 2001, quando alguns *experts* da indústria de sistemas de software reuniram-se para encontrar valores e princípios a fim de melhorar o desenvolvimento desses sistemas. A partir dessa união, foi criado o Manifesto Ágil [Agile Manifesto, 2001], cujos valores e princípios estabelecidos nesse Manifesto são fazer com que as equipes de desenvolvimento pudessem responder mais rápido às mudanças nas especificações e que o projeto fosse desenvolvido mais rapidamente.

Breve introdução sobre a origem dos métodos ágeis é apresentada na Seção 2.2. A metodologia ágil de gerência Scrum é abordada na Seção 2.3.

2.2. Origem dos Métodos Ágeis

Por muitos anos, a indústria baseou-se nos métodos tradicionais de desenvolvimento e de gerência de software, os quais definiram padrões para criação de sistemas de software nos meios acadêmico e empresarial. Porém, percebendo que a indústria apresentava muitos casos de fracasso, alguns líderes experientes adotaram modos de trabalho que se opunham aos principais conceitos das metodologias tradicionais. Esses modos foram aprimorados e, em alguns casos, chegaram a se transformar em novas metodologias de desenvolvimento e de gerência de sistemas chamadas de metodologias ágeis [Bassi Filho, 2008].

A abordagem ágil aplicada ao desenvolvimento de projetos ficou mais clara e melhor definida a partir de 2001, quando um grupo de 17 autores e representantes das mais variadas técnicas e metodologias ágeis, *eXtreme Programming* (XP), Scrum, *Dynamic System Development Method* (DSDM),

Adaptive Software Development (ASD), *Crystal*, *Feature Driven Development (FDD)* e *Lean Development*, reuniram-se para discutir e identificar o padrão de desenvolvimento de projetos dentre as técnicas e metodologias existentes [Pereira *et al.*, 2007]. O resultado desse encontro foi a criação do Manifesto para Desenvolvimento Ágil de Software [Agile Manifesto, 2001], que estabeleceu um *framework* comum para processos ágeis valorizando os seguintes itens:

- **Indivíduos e interação entre eles** mais que processos e ferramentas;
- **Software em funcionamento** mais que documentação abrangente;
- **Colaboração com o cliente** mais que negociação de contratos;
- **Responder a mudanças** mais que seguir um plano.

Ou seja, mesmo havendo valor nos itens a direita (sem negrito), deve-se valorizar mais os itens a esquerda (em negrito). O Manifesto Ágil estabelece 12 princípios para serem seguidos [Agile Manifesto, 2001]:

1. A maior prioridade é satisfazer o cliente, realizando entregas adiantadas e contínuas de software de valor;
2. Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adéquam a mudanças, para que o cliente possa tirar vantagens competitivas;
3. Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos;
4. Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto e diariamente durante o curso do projeto;
5. Construir projetos ao redor de indivíduos motivados. Proporcionar ao time de desenvolvimento um ambiente e suporte necessário para que façam seu trabalho;
6. O método mais eficiente e eficaz de transmitir informações para uma equipe de desenvolvimento é por meio de uma conversa cara a cara;

7. Software funcional é a medida primária de progresso;
8. Processos ágeis promovem um ambiente sustentável. Os patrocinadores, os desenvolvedores e os usuários devem ser capazes de manter indefinidamente passos constantes;
9. Contínua atenção a excelência técnica e bom *design* aumentam a agilidade;
10. Simplicidade, entregar apenas o que tem valor para o cliente;
11. As melhores arquiteturas, requisitos e *designs* emergem de times auto-organizáveis;
12. Em intervalos regulares, a equipe de desenvolvimento reflete em como ficar mais efetiva e se ajustam e otimizam seu comportamento.

Foi ressaltado no Manifesto Ágil o que mais tem valor para as metodologias ágeis. Processos, contratos, documentação e planejamento têm valor para o desenvolvimento de sistemas de software, porém são menos importantes do que saber como lidar com pessoas, do que ter o cliente colaborando para encontrar a melhor solução, do que entregar o sistema com qualidade e do que se adaptar às mudanças [Kalermo; Rissanen, 2002 *apud* Bassi Filho, 2008]. Algumas vantagens em aplicar os valores e os princípios ágeis são [Pereira *et al.*, 2007]:

- Nas organizações, criar um ambiente mais propício, para definir mudanças de requisitos e inovação durante o ciclo de desenvolvimento do produto, e colaborativo e produtivo entre desenvolvedores e cliente, resultando em entregas mais rápidas de produto melhor adaptados à realidade do cliente e com a qualidade esperada;
- Facilitar a gerência de projetos, uma vez que existem integração e comprometimento da equipe do projeto que, conseqüentemente, sente-se mais motivada;

- Reforçar o planejamento constante do projeto, o que minimiza os riscos, considerando que o planejamento é mais importante do que o plano. Não se deve parar de planejar até ter encontrado a satisfação do cliente com a entrega do produto final;
- Valorizar a satisfação do cliente.

2.3. Scrum

Em um jogo de *rugby*, "scrum" é uma jogada na qual o time recoloca a bola em jogo. Nessa jogada, é importante realizar um trabalho de equipe, pois, se um dos jogadores na formação falhar, a jogada é comprometida. O termo Scrum surgiu em 1986 [Nonaka; Takeuchi, 1986], porém a metodologia foi desenvolvida e formalizada por Ken Schwaber e Jeff Sutherland que fizeram a primeira coapresentação da metodologia Scrum na conferência OOPSLA (*Object-Oriented Programming, Systems, Languages & Applications*) de 1995 [Schwaber; Sutherland, 2011]. Na apresentação, Schwaber e Sutherland documentaram o aprendizado que tiveram ao longo dos anos anteriores na aplicação do Scrum.

Em Scrum, podem ser empregados vários processos/técnicas, deixando clara a eficácia relativa das práticas de gerência e de desenvolvimento de produtos, de modo que os membros do time possam melhorá-las [Schwaber; Sutherland, 2011]. Por um lado, o Scrum é simples. O processo, as suas práticas, os seus artefatos e suas regras são poucas, simples e fáceis de aprender. Por outro lado, a simplicidade do Scrum pode enganar, pois não descreve o que fazer em cada circunstância da vida de um projeto. Scrum é utilizado para os trabalhos complexos em que não é trivial prever tudo o que vai ocorrer [Schwaber, 2004]. Assim, o Scrum oferece uma estrutura e um conjunto de práticas que permite as equipes de desenvolvimento que o utilizam saber exatamente o que está

acontecendo e fazer ajustes para manter o projeto caminhando para os objetivos desejados.

O principal foco de Scrum é na gerência do projeto, sem determinar como a equipe executará as tarefas de programação. A abordagem de auto-organização da equipe permite a integração com outras metodologias ágeis, cujo foco é nas práticas de programação, por exemplo, XP (*eXtreme Programming*) [Bassi Filho, 2008]. Diversos relatos descrevem a combinação de Scrum e de XP com sucesso [Kniberg, 2007; Pikkarainen; Passoja, 2005 *apud* Bassi Filho, 2008; Danait, 2005 *apud* Bassi Filho, 2008]. As práticas do Scrum estão sustentadas em um processo iterativo e incremental apresentado na Figura 3. O ciclo inferior representa uma iteração de atividades de desenvolvimento que ocorre uma após a outra. A saída de cada iteração é um incremento do produto. O ciclo superior representa a inspeção diária que ocorre durante a iteração, em que os membros do time reúnem-se para inspecionar as atividades dos outros e fazer as adaptações apropriadas. Esse processo se repete até o projeto gerar o produto esperado ou não ser mais financiado [Schwaber, 2004].



Figura 3 - Processo Scrum (Fonte: [Schwaber, 2004])

O principal ponto do Scrum reside na iteração. A equipe analisa os requisitos, considera a tecnologia disponível e avalia suas próprias habilidades e

capacidades. Em seguida, determina coletivamente como construir a funcionalidade, alterando a sua abordagem diariamente à medida que encontra novas complexidades, dificuldades e surpresas. A equipe descobre o que precisa ser feito e escolhe a melhor maneira de fazê-lo. Esse processo iterativo é o responsável pela alta produtividade das equipes que utilizam o Scrum [Schwaber, 2004].

2.3.1. Elementos de Apoio

Os elementos que a equipe produz para seguir as práticas do Scrum são "cartões" com as funções do sistema e os gráficos de acompanhamento. Esses gráficos são atualizados frequentemente e devem refletir o estado do projeto para fornecer *feedback* à equipe. Os elementos gerados pela equipe no decorrer das atividades do Scrum são [Bassi Filho, 2007]:

- **Product Backlog.** Lista dos "cartões" com as funções que o sistema deve possuir e ainda não foram desenvolvidas;
- **Selected Product Backlog.** Um subconjunto de funções que o cliente escolheu a partir do *Product Backlog* para ser implementado na *Sprint* atual e que não pode ser modificado durante a *Sprint*;
- **Sprint Backlog.** Lista priorizada obtida a partir da fragmentação dos itens do *Selected Product Backlog*;
- **Backlog de Impedimentos.** Lista dos obstáculos identificados pela equipe que não pertencem ao contexto do desenvolvimento;
- **Gráficos de Acompanhamento.** A quantidade de trabalho restante é medido nesses gráficos, os mais utilizados em Scrum são gráficos *Burndown*. É recomendado fazê-los para várias esferas do projeto: para o produto, para a *release* e *Sprint*.

2.3.2. Time Scrum

O Time Scrum é formado por três papéis que dividem as responsabilidades de gerência em um projeto Scrum: i) o *Product Owner*; ii) a Equipe de Desenvolvimento; e iii) o *Scrum Master*. Times Scrum são auto-organizáveis, escolhem qual é a melhor forma para completarem seu trabalho ao invés de serem dirigidos por outros de fora da equipe, e multifuncionais, possuem competências necessárias para completar o trabalho sem depender de outros que não fazem parte da equipe. O modelo de equipe no Scrum é projetado para aperfeiçoar a flexibilidade, a criatividade e a produtividade [Schwaber; Sutherland, 2011].

O *Product Owner* é responsável por representar os interesses dos envolvidos no projeto e o seu sistema resultante. Deve possuir a visão do produto a longo prazo com o gerenciamento do *backlog* e a curto prazo com a definição do objetivo da *Sprint* selecionada. Quando existem vários interessados/envolvidos com o produto, o *Product Owner* deve ser representado por uma única pessoa que entenda as necessidades e seja capaz de priorizá-las [Bassi Filho, 2008]. As principais responsabilidades do *Product Owner* são [Schwaber, 2004]: i) Conseguir financiamento inicial e permanente para o projeto; ii) Criar o projeto inicial de requisitos do projeto (*Product Backlog*); iii) Garantir o retorno sobre o investimento (*return on investment*, ROI) do produto; iv) Definir os planos de lançamento; v) Gerenciar o *Product Backlog*; e vi) Garantir que as funções mais valiosas sejam produzidas primeiro. Para o *Product Owner* ter sucesso, a organização deve respeitar as suas decisões. Essas decisões são visíveis no conteúdo e na priorização do *Product Backlog* [Schwaber; Sutherland, 2011].

A Equipe de Desenvolvimento deve ser multifuncional e autossuficiente, o que significa que seus membros devem reunir as habilidades necessárias para

atingir o objetivo, assumindo compromissos e responsabilidades em relação ao projeto. Seus membros devem trabalhar em equipe para atingir o objetivo. Uma equipe de desenvolvimento Scrum deve ser auto-organizada: cada membro deve saber que seu trabalho é igual e, essencialmente, importante para o sucesso do projeto [Bassi Filho, 2008]. As principais responsabilidades da Equipe de Desenvolvimento são [Schwaber, 2004]: i) Desenvolver funções; ii) Descobrir como transformar o *Product Backlog* em um incremento de funções; iii) Gerenciar seu próprio trabalho; e iv) Garantir o sucesso de cada iteração e do projeto como um todo.

O *Scrum Master* deve possuir conhecimento do processo para garantir que ele seja seguido e deve manter uma ampla visão sobre o projeto [Bassi Filho, 2008]. Ele é um líder para o Time Scrum e suas responsabilidades podem ser divididas entre o trabalho com o *Product Owner*, a Equipe de Desenvolvimento e a Organização (Tabela 1).

Tabela 1 - Responsabilidades do Scrum Master (Fonte: [Schwaber; Sutherland, 2011])

Com o <i>Product Owner</i>	Com a Equipe de Desenvolvimento	Com a Organização
Encontrar técnicas para o gerenciamento efetivo do <i>Product Backlog</i> .	Treinar a Equipe de Desenvolvimento em autogerenciamento e interdisciplinaridade.	Liderar e treinar a organização na adoção do Scrum.
Claramente comunicar a visão (<i>vision</i>), o objetivo e os itens do <i>Product Backlog</i> para a Equipe de Desenvolvimento.	Ensinar e liderar a Equipe de Desenvolvimento na criação de produtos de alto valor.	Planejar implementações Scrum dentro da organização.
Ensinar o Time Scrum a criar itens de <i>Product Backlog</i> de forma clara e concisa.	Remover impedimentos para o progresso da Equipe de Desenvolvimento.	Ajudar funcionários e partes interessadas a compreender e tornar aplicável o Scrum e o desenvolvimento de produto empírico.

Tabela 1 - Responsabilidades do Scrum Master (Fonte: [Schwaber; Sutherland, 2011]) (cont.)

Com o <i>Product Owner</i>	Com a Equipe de Desenvolvimento	Com a Organização
Compreender a longo-prazo o planejamento do Produto no ambiente empírico.	Facilitar os eventos Scrum conforme exigidos ou necessários.	Causar mudanças que aumentam a produtividade do Time Scrum.
Compreender e praticar a agilidade.	Treinar a Equipe de Desenvolvimento em ambientes organizacionais nos quais o Scrum não é totalmente adotado e compreendido.	Trabalhar com outro <i>Scrum Master</i> para aumentar a eficácia da aplicação do Scrum nas organizações.
Facilitar os eventos Scrum conforme exigidos ou necessários.		

2.3.3. Fluxo do Scrum

Nas próximas seções, é apresentado o fluxo das atividades do Scrum apresentado na Figura 4.

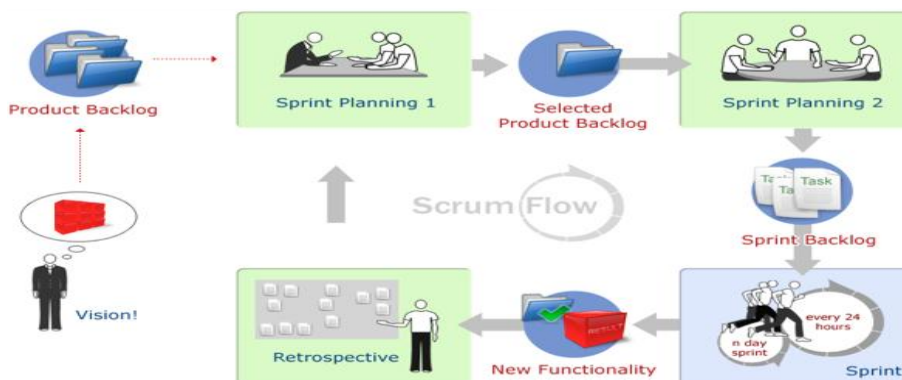


Figura 4 - Fluxo das Atividades do Scrum (Fonte: [Bassi Filho, 2008])

2.3.3.1. Vision

Um projeto Scrum começa com uma visão do sistema (*vision*) a ser desenvolvido. A *vision* pode ser vaga a princípio, talvez descrita em termos de

mercado ao invés de termos de sistema, mas ela tornar-se-á clara com o decorrer do projeto. Posteriormente, o *Product Owner* formula o *Product Backlog* (Figura 4) [Schwaber, 2004]. O primeiro planejamento de um projeto Scrum pode durar desde dias até algumas semanas, o tempo necessário para a equipe conhecer satisfatoriamente as características do produto a ser desenvolvido e escolher as tecnologias e as ferramentas de trabalho [Bassi Filho, 2008].

2.3.3.2. *Sprint*

Sprint é um evento *time-box* de um mês ou menos, durante o qual uma versão incremental potencialmente utilizável do sistema é criada. *Sprints* têm durações coerentes no esforço de desenvolvimento. Uma nova *Sprint* é iniciada imediatamente após a conclusão da *Sprint* anterior [Schwaber; Sutherland, 2011]. Depois da elaboração do *Product Backlog*, os planejamentos seguintes têm duração de aproximadamente dois dias, nos quais três reuniões são realizadas: uma para estimar os itens do *Product Backlog*, outra para definir a meta da *Sprint* e a terceira para definir as tarefas necessárias para atingir a meta. A cada *Sprint*, o *Product Owner* baseia-se nas prioridades de negócio para escolher uma parte do *Product Backlog* para compor o *Sprint Backlog*. A equipe pode concordar com a seleção ou negociar os itens até comprometa-se a produzi-los durante a *Sprint* [Bassi Filho, 2008].

Na reunião de estimativa, a equipe se reúne para estimar a quantidade de trabalho dos itens mais prioritários do *Product Backlog* com a presença do *Scrum Master* e do *Product Owner* para esclarecer dúvidas referentes às regras de negócio. Depois, na primeira reunião de planejamento da *Sprint* (Figura 4, *Sprint Planning 1*), o time Scrum reúne-se para definir a meta da *Sprint* e criar o *Sprint Backlog* de acordo com as prioridades de negócio estabelecidas pelo *Product Owner*. Em seguida, na segunda reunião de planejamento da *Sprint* (Figura 4, *Sprint Planning 2*), a equipe reúne-se para dividir os itens do *Backlog*

selecionado em tarefas de implementação de menor granularidade [Bassi Filho, 2008]. Durante a *Sprint*, o *Scrum Master* assegura que os itens do *Backlog* selecionado não mudarão e elimina os itens do *Backlog* de impedimentos para garantir que a equipe mantenha-se focada em seu objetivo. O *Product Owner* acompanha o desenvolvimento e esclarece eventuais dúvidas, sem poder mudar o escopo. Porém, caso realmente precise mudar os itens do *Sprint Backlog*, ele tem a opção de cancelar a *Sprint*. Nesse caso, a iteração volta ao início, para a *Sprint Planning 1* (Figura 4).

2.3.3.3. Daily Scrum

Diariamente, durante a *Sprint*, os membros da equipe reúnem-se no mesmo horário para fazer a *Daily Scrum* (Reunião Diária do Scrum), uma rápida reunião para sincronizar o trabalho. A *Daily Scrum* é um evento *time-boxed* de aproximadamente 15 minutos para a Equipe de Desenvolvimento sincronizar as atividades e criar um plano para as próximas 24 horas. Essa reunião é feita para inspecionar o trabalho desde a última Reunião Diária e prever o trabalho a ser feito antes da próxima Reunião Diária [Schwaber; Sutherland, 2011]. Durante a reunião, cada integrante da Equipe de Desenvolvimento esclarece: i) O que foi completado desde a última reunião; ii) O que será feito até a próxima reunião; e iii) Quais os obstáculos que estão no caminho. Reuniões Diárias melhoram a comunicação, eliminam outras reuniões, identificam e removem impedimentos para o desenvolvimento, destacam e promovem rápidas tomadas de decisão e melhoram o nível de conhecimento da Equipe de Desenvolvimento [Schwaber; Sutherland, 2011].

2.3.3.4. Revisão da Sprint

Quando a *Sprint* termina, a equipe apresenta o trabalho na reunião de revisão da *Sprint* ao *Product Owner* por meio de uma demonstração. O *Product*

Owner faz testes para verificar se cada item atende às suas expectativas e determinar se a meta foi atingida. A meta "é atingida" ou "não é atingida". Não existem opções intermediárias. Dessa forma, a equipe deve se preocupar com a qualidade de tudo que entrega para não colocar em risco o trabalho da *Sprint* inteira [Bassi Filho, 2008]. A Revisão da *Sprint* é uma reunião *time-boxed* de aproximadamente 4 horas de duração para uma *Sprint* de um mês. Proporcionalmente, um tempo menor é alocado para *Sprints* menores [Schwaber; Sutherland, 2011]. Por exemplo, uma *Sprint* de duas semanas tem Reuniões de Revisão de duas horas.

2.3.3.5. Retrospectiva

Após cada entrega, a equipe realiza uma retrospectiva (Figura 4, *Retrospective*) para avaliar o seu trabalho e identificar as oportunidades para melhorar seu desempenho nos próximas *Sprints*. O *Scrum Master* e os membros da equipe repensam a última *Sprint* e refinam o processo de desenvolvimento com uma reflexão sobre o passado [Bassi Filho, 2008]. O propósito da retrospectiva da *Sprint* é [Schwaber; Sutherland, 2011]: i) Inspeccionar como foi o andamento da última *Sprint* em relação as pessoas, os processos e as ferramentas; ii) Identificar e ordenar os principais pontos de sucesso da *Sprint* e as potenciais melhorias; e iii) Criar um plano para implementar melhorias no modo que o Time Scrum faz seu trabalho. Ao final, o Time Scrum deverá ter identificado melhorias a serem implementadas na próxima *Sprint*. A implementação dessas melhorias é a forma de adaptação à inspeção e melhoria que o Time Scrum faz a si próprio. A Retrospectiva fornece um evento dedicado e focado na inspeção e na adaptação.

2.3.4. Práticas Ágeis Utilizadas com Scrum

Algumas práticas ágeis vêm sendo utilizadas junto com Scrum, entre elas: i) *Planning Poker*; ii) *User Stories*; iii) Kanban; e iv) Gráfico *Burndown*. Nas próximas seções, são apresentadas essas prática resumidamente.

2.3.4.1. *Planning Poker*

O *Planning Poker* [Cohn, 2005] é uma abordagem ágil utilizada com Scrum para estimar os itens do *Backlog* na reunião de planejamento das *Sprints*. A estimativa pode ser feita em horas ou tamanho. No *Planning Poker*, os membros do Time, inclusive o *Product Owner*, participam de forma democrática para chegar a um consenso de estimativa para cada item do *Backlog* [Pereira *et al.*, 2007]. O primeiro passo dessa abordagem é fornecer, para cada membro do Time, um conjunto de cartas com sequência numérica (Figura 5), os valores das cartas mais utilizados são (0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40 e 100), porém outros valores podem ser utilizados para representar a mesma ideia. Os valores das cartas representam a estimativa dos itens do *Backlog* [Schwaber; Sutherland, 2011]. Algumas das cartas do *Planning Poker* têm significado especial [Kniberg, 2007]:

- **0 (zero)**. Representa uma tarefa concluída ou com um tempo curto para conclusão que não vale a pena ser mensurado, por exemplo, alguns minutos;
- **100 (cem)**. Representa uma tarefa grande. O ideal é ela ser quebrada em mais tarefas, pois o risco de estimar errado torna-se alto em tarefas grandes;
- **? (interrogação)**. Representa uma tarefa indefinida. Além de não ser possível entender o seu tamanho, não se consegue dizer se a tarefa é pequena ou grande;
- **"símbolo da xícara de café"**. Representa uma sugestão de pausa para o café, água ou descanso por um integrante do Time, por causa do tempo da reunião estar longo ou a reunião estar cansativa.

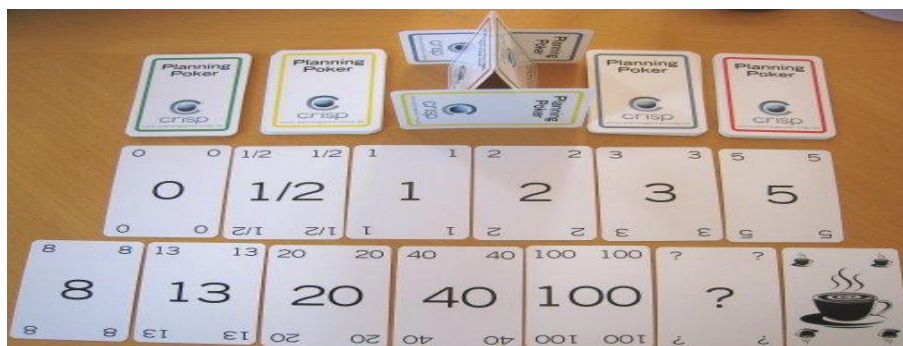


Figura 5 - Cartas do *Planning Poker* (Fonte: [Kniberg, 2007])

Após as cartas serem distribuídas aos participantes, o *Product Owner* ou o cliente apresenta um item do *Backlog* e cada estimador seleciona uma carta para representar sua estimativa, sem mostrar aos demais. Após a seleção, os estimadores revelam as cartas ao mesmo tempo. Em caso de divergência, os participantes devem explicar os motivos de sua escolha, para que os outros membros do Time possam refletir e validá-la. Finalizada a discussão, realiza-se nova rodada para que os estimadores tenham a oportunidade de reavaliar seus julgamentos. É importante existir um moderador para as discussões serem produtivas e não polemizadas. Esse ciclo é seguido até os itens do *Backlog* serem estimados [Pereira *et al.*, 2007; Balle, 2011].

2.3.4.2. *User Stories*

User Stories são pequenas descrições sobre as funções do sistema, cujos detalhes devem ser negociados entre o cliente e a equipe desenvolvedora [Cohn, 2004]. Após serem elaboradas pelo cliente ou *Product Owner*, as *User Stories* podem ser divididas em tarefas de menor granularidade pelos membros do Time Scrum. *User Stories* têm a mesma finalidade dos Casos de Uso da UML (*Unified Modeling Language*). Usualmente, cada *User Story* é um conjunto de frases curtas, em média três sentenças, escritas pelo cliente para definir o que o sistema deve fazer para ele. Elas são utilizadas para criar estimativas de tempo nas

reuniões de planejamento e, mais importante, analisar se o que foi desenvolvido atende as necessidades especificadas na *User Story* [Souza Neto, 2004]. Um exemplo de *User Story* é apresentada na Figura 6 em que dados como data, número e descrição são escritos.

Customer Story and Task Card		BIW Development / COLA	
DATE: 3/19/98	TYPE OF ACTIVITY: NEW: <input checked="" type="checkbox"/> FIX: <input type="checkbox"/> ENHANCE: <input type="checkbox"/> FUNC. TEST: <input type="checkbox"/>		
STORY NUMBER: 1275 / 1275	PRIORITY: USER: _____ TECH: _____		
PRIOR REFERENCE: _____	RISK: _____ TECH ESTIMATE: _____		
TASK DESCRIPTION: SPLIT COLA: When the COLA rate chgs. in the middle of the BIW Pay Period, we will want to pay the 1 ST week of the pay period at the OLD COLA rate and the 2 ND week of the Pay Period at the NEW COLA rate. Should occur automatically based on system design.			
NOTES: on system design. For the OT, we will run a m/frame program that will pay or calc the COLA on the 2 ND week of OT. The plant currently retransmits the hours data for the 2 ND week exclusively so that we can calc COLA. This will come into the Model as a "2144" COLA			
TASK TRACKING: Gross Pay Adjustment. Create RM Boundary and Place in DE Ent Express COLA			
Date	Status	To Do	Comments

Figura 6 - *User Stories* (Fonte: [Beck, 2004])

2.3.4.3. Gráficos *Burndown*

O gráfico *Burndown* é uma representação gráfica do trabalho a ser feito em relação ao tempo. No Scrum, o *Burndown* é utilizado na maioria das vezes para medir o trabalho de uma *Sprint*, mas pode ser utilizado para medir o trabalho do projeto como um todo. É exibido no gráfico o progresso diário do Time em função do total de dias/horas estabelecidas pela soma de horas das tarefas dos itens do *Product Backlog* selecionados [Pereira *et al.*, 2007]. Nesse gráfico (Figura 7), o trabalho restante é visto no eixo horizontal e os períodos de tempo (dias de uma *Sprint*) são vistos no eixo vertical. A quantidade de trabalho concluído em um ponto do tempo é apresentada na intersecção de uma linha de tendência do trabalho restante e o eixo horizontal [Cohn, 2002 apud Balle, 2011].

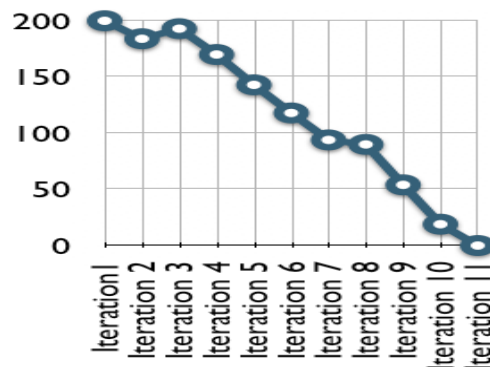


Figura 7 - Exemplo de Gráfico *Burndown* (Fonte: [Cohn, 2002 apud Balle, 2011])

2.3.4.4. *Kanban*

Kanban é uma palavra japonesa que significa placa visível. O seu uso teve origem na indústria de produção [Sugimori *et al.*, 1977]. O *Kanban* não é uma prática que faz parte do Scrum, mas é normalmente utilizado em conjunto com essa metodologia. Ele é uma ferramenta de processo que se baseia em visualizar o fluxo de trabalho, dividindo-o em partes. Por isso, a interação com *User Stories* é tão direta [Balle, 2011]. O *Kanban* possibilita a visualização do fluxo de trabalho, pois cada tarefa da *Sprint* é associada a um cartão "colado" em uma das colunas do fluxo de trabalho (Figura 8). Por exemplo, a tarefa H está "A Fazer" e a tarefa A está "Feito!". Ele ajuda os integrantes do Time Scrum a acompanhar o fluxo das tarefas e a otimizar o processo para tornar menor e mais previsível o tempo de execução dessas tarefas [Kniberg; Skarin, 2009].



Figura 8 - Exemplo de *Kanban* (Fonte: [Kniberg; Skarin, 2009])

2.4. Considerações Finais

Neste capítulo, foram apresentados a origem das metodologias ágeis e os conceitos de Scrum. No Scrum, o objetivo é ajudar as equipes de desenvolvimento de sistemas de software a ganhar mais agilidade gerenciando o desenvolvimento de produtos complexos. Para isso, ele proporciona processos e técnicas que evidenciam a eficácia relativa das práticas de gerenciamento e de desenvolvimento de produtos, de modo que a equipe possa adaptar de acordo com sua necessidade conseguindo melhores resultados.

3. QUALIDADE DE SOFTWARE

3.1. Considerações Iniciais

As empresas querem oferecer produtos e serviços melhores, mais rápidos e mais baratos. Ao mesmo tempo, no ambiente de alta-tecnologia do século XXI, as organizações desenvolvem produtos cada vez mais complexos [SEI, 2010a]. Uma forma de garantir a qualidade de sistemas de software é focar na melhoria dos processos, garantindo a qualidade desde o início do desenvolvimento desses sistemas até a sua entrega e manutenção, controlando e medindo passo a passo a qualidade. É importante realizar testes de verificação e de validação ao final de cada etapa do desenvolvimento, evitando que erros sejam passados de uma fase para a outra, o que minimiza as falhas finais e evita prejudicar o cliente por um mau comportamento do produto.

Conceituação do termo qualidade e uma definição de qualidade de software são apresentadas na Seção 3.2. Um estudo sobre qualidade de processo, descrevendo resumidamente normas e modelos utilizados para a melhoria dos processos de software, é destacado na Seção 3.2. O MPS.BR - Melhoria de Processo do Software Brasileiro é brevemente abordado na Seção 3.4.

3.2. Conceitos

A definição de qualidade de software tem evoluído ao longo do tempo. Inicialmente, foi definida como a conformidade com um padrão ou uma especificação. Posteriormente, a definição foi modificada para adaptar-se aos ambientes de negócios altamente dinâmicos [Gomes, 2004]. Em 1991, a *International Organization for Standardization* (ISO) adotou a norma ISO/IEC 9126 como o padrão para avaliar a qualidade de sistemas de software [Agrawal; Chari, 2007]. A qualidade foi definida nessa norma como a totalidade de funções

e de características de um produto ou serviço que tem a capacidade de satisfazer determinadas necessidades [ISO/IEC 9126, 2001]. A qualidade não é sinônimo de perfeição, trata-se de algo factível, relativo, substancialmente dinâmico e evolutivo, moldado a partir da granularidade dos objetivos a serem atingidos. Considerá-la absoluta e definitiva, transformaria a qualidade em algo inatingível gerando entraves a qualquer esforço de obtê-la [Belchior, 1997].

Provavelmente, o pesquisador da qualidade mais famoso e precursor do movimento de qualidade mundialmente é W. Edwards Deming. Ele definiu qualidade como a conformidade de um produto com as especificações técnicas que lhe foram atribuídas. Joseph M. Juran definiu qualidade em termos da adequação de um produto à sua utilização pretendida. Essa definição aproximou o conceito de qualidade à perspectiva do cliente, diferentemente da definição de Deming cujo foco é na qualidade técnica do produto [Gomes, 2004; Godfrey; Kenett 2007]. O conceito de qualidade tem oito dimensões que correspondem aos fatores que os clientes consideram importantes quando compram um produto para atender as suas necessidades [Garvin, 1987]:

- **Performance.** Desempenho do produto em relação às principais funções;
- **Funcionalidade.** Conjunto de funções que complementam a oferta do produto;
- **Confiabilidade.** Probabilidade do produto continuar a funcionar de forma adequada;
- **Conformidade.** Grau de adequação do produto às especificações. Reflete a perspectiva de Deming e Juran e serve para o controle estatístico;
- **Durabilidade.** Tempo de vida do produto em termos técnicos ou até ao momento em que a reparação deixa de ser eficiente do ponto de vista econômico;

- **Serviço.** Inclui rapidez, cortesia, competência e facilidade em reparar o produto;
- **Aparência.** Refere-se à estética ou apelo sensorial do produto;
- **Imagem.** Refere-se a uma percepção subjetiva de qualidade associada à marca.

No contexto de desenvolvimento de sistemas de software, qualidade pode ser entendida como um conjunto de características a serem satisfeitas em um determinado grau, de modo que o sistema atenda às necessidades explícitas e implícitas de seus usuários. Entretanto, não se obtém qualidade de forma espontânea. Ela tem de ser construída. Assim, a qualidade de sistemas de software depende fortemente da qualidade de seu processo de desenvolvimento [Duarte; Falbo, 2000].

3.3. Qualidade de Processo

As organizações têm investido cada vez mais em pesquisa e em aplicação da melhoria dos processos de software. Isso se justifica em função de estudos que comprovaram que a maioria dos projetos de software não atende aos objetivos traçados [Rincon, 2009]. Isso decorre da falta de processos adequados nas organizações em que eles são desenvolvidos. Diversos modelos e normas internacionais de qualidade de processos de sistemas software foram definidos para atender as necessidades das empresas em melhoria de processos, entre elas:

- **ISO/IEC 15504.** A norma ISO/IEC 15504 é um modelo, cujo foco é a melhoria dos processos de desenvolvimento de sistemas de software e a determinação da capacidade de processos de uma organização [Felipe, 2006]. Essa norma está dividida em sete partes [ISO/IEC 15504, 2003]:
 - ✓ **ISO/IEC 15504-1.** Fornece visão geral e glossário dos conceitos utilizados na avaliação de processos;

- ✓ **ISO/IEC 15504-2.** Estabelece requisitos mínimos para execução de uma avaliação. É a única parte normativa da ISO/IEC 15504;
- ✓ **ISO/IEC 15504-3.** Provê um guia para interpretar os requisitos para uma avaliação;
- ✓ **ISO/IEC 15504-4.** Contém um guia para a utilização dos resultados de uma avaliação de processos, para melhoria ou determinação de capacidade;
- ✓ **ISO/IEC 15504-5.** Exemplifica um modelo de avaliação de processo de software baseado na norma ISO/IEC 12207 [ISO/IEC 12207, 2008];
- ✓ **ISO/IEC TR 15504-6.** Provê um exemplo de modelo de avaliação de processo para processos de ciclo de vida de sistemas, em conformidade com os requisitos da parte 2 da ISO/IEC 15504;
- ✓ **ISO/IEC TR 15504-7.** Define condições para avaliação da maturidade organizacional, baseada nos perfis de capacidade derivados dos processos de avaliação e definem quais são as condições para uma avaliação ser considerada válida.

A Norma ISO/IEC 15504, informalmente, está dividida em duas [Salviano, 2006]. Uma é a ISO/IEC 15504 utilizada como *framework* para modelos e métodos de avaliação e melhoria de processo. A outra é o modelo ISO/IEC 15504-5, um modelo de capacidade de processo (definido segundo o *framework* da ISO/IEC 15504) para a engenharia de software, baseado nos processos definidos na ISO/IEC 12207;

- **CMMI-DEV.** O CMMI-DEV (*Capability Maturity Model Integration for Development*) [SEI, 2010a] é um modelo de maturidade e capacidade de processos de software criado pelo SEI (*Software Engineering Institute*) e que consiste de boas práticas de engenharia de software para o desenvolvimento e a manutenção de produtos e de serviços. O modelo

oferece estrutura e elementos chave para um processo de software eficaz, abrangendo o ciclo de produção, desde a concepção até a entrega e manutenção do software [Mello, 2011]. O CMMI-DEV possui as representações contínua e por estágios [SEI, 2010a]. Na representação contínua, a organização pode selecionar uma área de processo específica e melhorá-la. Na representação por estágios, um conjunto de áreas de processo estabelecido para definir um caminho de melhoria para a unidade organizacional, descrito em termos de níveis de maturidade [SEI, 2010a].

- **MPS.BR.** O Programa Brasileiro de Melhoria de Processos de Software (MPS.BR) [SOFTEX, 2012] é uma iniciativa sob a coordenação da Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), com apoio do governo (MCT e FINEP), SEBRAE/PROIMPE e BID (Banco Interamericano de Desenvolvimento), da indústria de software brasileira e de instituições de pesquisa. O MPS.BR visa ao estabelecimento de um caminho economicamente viável para organizações, incluindo as micro, pequenas e médias empresas (mPME), alcançarem os benefícios da melhoria de processos e da utilização de boas práticas da engenharia de software em um intervalo de tempo razoável [Travassos; Kalinowski, 2012].

3.4. MPS.BR

O MPS.BR foi criado em 2003 e, atualmente, quatro componentes são apresentados [SOFTEX, 2012]: i) MR-MPS-SW (Modelo de Referência MPS para Software); ii) MR-MPS-SE (Modelo de Referência MPS para serviço); iii) MA-MPS (Método de Avaliação MPS); e iv) MN-MPS (Modelo de Negócio MPS). Nos dois primeiros, são definidos requisitos a serem atendidos pelas organizações em seus processos; no terceiro, é estabelecido um processo/método de avaliação de processos, no qual é fornecido sustentação e garantido que o modelo MPS.BR seja empregado de forma coerente com as suas definições; e,

no último, é apresentado o modelo de negócio do MPS para apoiar a sua adoção pelas empresas desenvolvedoras de software e prestadores de serviços.

Tomaram como base para a construção do MPS.BR, as normas ISO/IEC 12207 [ISO/IEC 12207, 2008], ISO/IEC 20000 [ISO/IEC, 2011] e ISO/IEC 15504-2 [ISO/IEC 15504, 2003], o que permite considerar que o modelo está em conformidade com estas normas. O MPS.BR garante o conteúdo do CMMI-SE e CMMI-SVC (SEI, 2010b) por causa da inclusão de processos e de resultados. Os componentes do MSP.BR são exibidos na Figura 9 e são descritos pela SOFTEX por meio de cinco guias [SOFTEX, 2012]: i) Guia Geral MPS de Software; ii) Guia de Aquisição; iii) Guias de Implementação; iv) Guia Geral MPS de Serviços; e v) Guia de Avaliação.

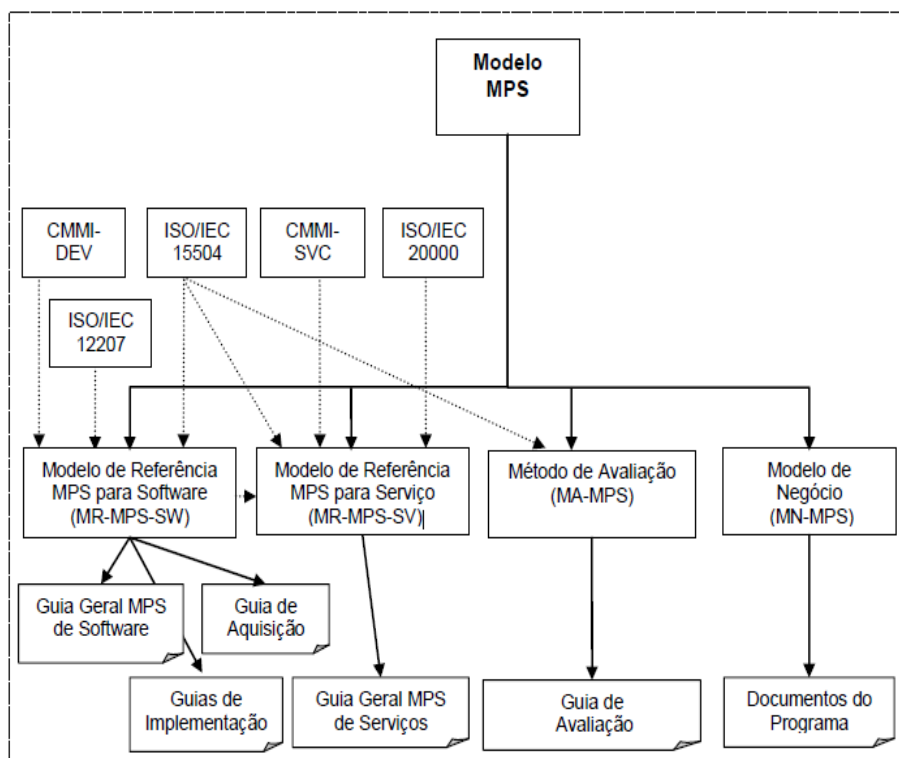


Figura 9 - Componentes do MPS-BR (Fonte: [SOFTEX, 2012])

3.4.1. Níveis de Maturidade

No MR-MPS-SW, são definidos níveis de maturidade que consistem em uma combinação entre processos e sua capacidade. A capacidade do processo é a caracterização da habilidade do processo para alcançar os objetivos de negócio, atuais e futuros, relacionados com o atendimento aos atributos de processo associados aos processos de cada nível de maturidade [SOFTEX, 2012; Mello, 2011]. O progresso e o alcance de um nível de maturidade do MPS-BR se obtêm quando são atendidos os propósitos e os resultados esperados dos respectivos processos e dos atributos de processo estabelecido para aquele nível. A divisão em estágios do modelo MPS tem como referência os níveis de maturidade do CMMI-DEV, mas com dois níveis a mais. O propósito de aumentar a quantidade de níveis é proporcionar implementação e avaliação mais gradual e adequada às pequenas e médias empresas [Colenci Neto, 2008]. No MR-MPS-SW, são definidos sete níveis de maturidade, os níveis e os processos associados a cada um deles é apresentado na Figura 10.

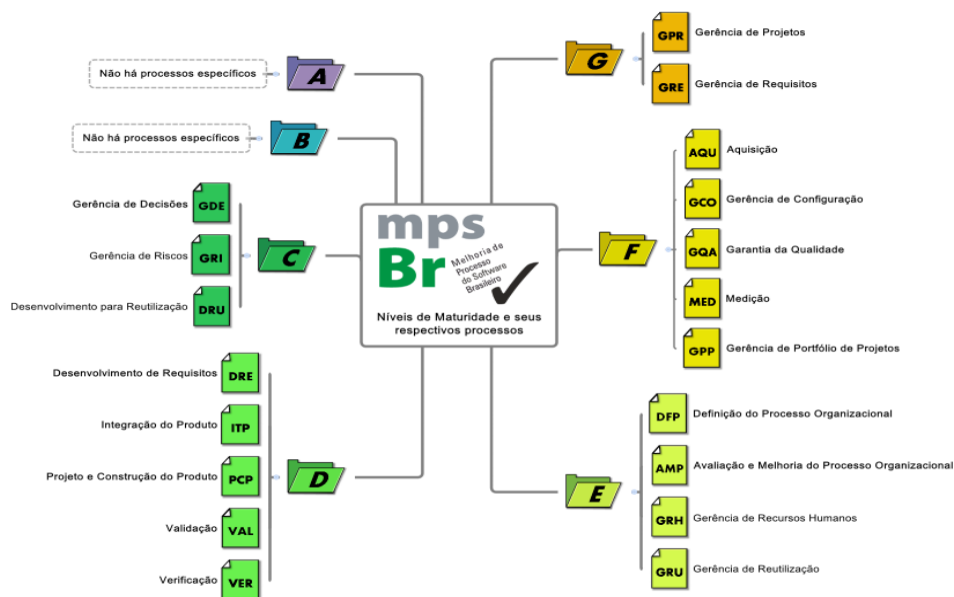


Figura 10 - Níveis e Processos do MPS.BR (Fonte: [SOFTEX, 2012])

Os níveis de maturidade se assemelham a uma escada, nas quais estão os processos, e sua evolução indica o grau em que o processo da organização está. A escada de maturidade se inicia no nível G e progride até o nível A [SOFTEX, 2012]: i) Nível G - Parcialmente Gerenciado; ii) Nível F - Gerenciado; iii) Nível E - Parcialmente Definido; iv) Nível D - Largamente Definido; v) Nível C - Definido; vi) Nível B - Gerenciado Quantitativamente; e vii) Nível A - Otimizado.

3.4.2. Níveis Iniciais do MPS.BR (Níveis G e F)

Nas próximas seções, o Níveis G e o Nível F, utilizados neste trabalho, são detalhados para o desenvolvimento da ferramenta ScrumMps Versão 2.0.

3.4.2.1. Nível G

Esse nível é o primeiro de maturidade do MR-MPS-SW. O objetivo da sua implantação é nortear a organização para que ela seja capaz de gerenciar parcialmente seus projetos de desenvolvimento de software [Pommer, 2011]. Sua implementação deve ser executada com cautela por estabelecer o início dos trabalhos em implantação de melhoria dos processos de software na organização. Ao final da implantação, a organização deve ser capaz de gerenciar parcialmente seus projetos de desenvolvimento de sistemas de software [SOFTEX, 2011a]. No Guia de Implementação, são destacados dois desafios na implementação do Nível G: i) a mudança da cultura organizacional, orientando a definição e melhoria dos processos de desenvolvimento de software; e ii) a definição do conceito acerca do que é "projeto" para a organização.

O Nível G contém os processos Gerência de Projetos e Gerência de Requisitos. Na implementação desses processos, devem ser satisfeitos os atributos de processo AP 1.1 e AP 2.1. O alcance de cada atributo de processo é

avaliado utilizando os respectivos resultados esperados de atributo de processo (RAP) [SOFTEX, 2012]:

- **AP 1.1 O processo é executado.** Este atributo evidencia o quanto o processo atinge o seu propósito. O resultado esperado para esse atributo é:
 - ✓ **RAP 1:** O processo atinge seus resultados definidos.
- **AP 2.1 O processo é gerenciado.** Este atributo evidencia o quanto a execução do processo é gerenciada. Os resultados esperados para esse atributo nos Níveis G e F são:
 - ✓ **RAP 2.** Existe política organizacional estabelecida e mantida para o processo;
 - ✓ **RAP 3.** A execução do processo é planejada;
 - ✓ **RAP 4 (Para o nível G).** A execução do processo é monitorada e ajustes são realizados;
 - ✓ **RAP 4 (Para o nível F).** Medidas são planejadas e coletadas para monitoração da execução do processo e ajustes são realizados;
 - ✓ **RAP 5.** Informações e recursos necessários para a execução do processo são identificados e disponibilizados;
 - ✓ **RAP 6.** Responsabilidades e autoridade para executar o processo são definidas, atribuídas e comunicadas;
 - ✓ **RAP 7.** Pessoas que executam o processo são competentes em termos de formação, treinamento e experiência;
 - ✓ **RAP 8.** Comunicação entre as partes interessadas no processo é planejada e executada de forma a garantir o seu envolvimento;
 - ✓ **RAP 9.** Resultados do processo são revistos com a gerência de alto nível para fornecer visibilidade sobre a sua situação na organização;
 - ✓ **RAP 10 (Para o nível G).** O processo planejado para o projeto é executado;

- ✓ **RAP 10 (Para o nível F).** A aderência dos processos executados às descrições de processo, padrões e procedimentos é avaliada objetivamente e são tratadas as não conformidades.

No processo Gerência de Projetos, o objetivo é identificar, estabelecer, coordenar e monitorar as atividades, as tarefas e os recursos, que um projeto necessita para produzir um produto ou serviço, no contexto dos seus requisitos e restrições do projeto. Além disso, deve-se prover informações sobre o seu andamento, que permitam a realização de correções quando houver desvios significativos no desempenho do projeto. O propósito desse processo evolui à medida que a organização cresce em maturidade. Os resultados esperados para esse processo nos Níveis G e F são [SOFTEX, 2012]:

- **GPR 1.** O escopo do trabalho para o projeto é definido;
- **GPR 2.** As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;
- **GPR 3.** O modelo e as fases do ciclo de vida do projeto são definidos;
- **GPR 4.** O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas;
- **GPR 5.** O orçamento e o cronograma do projeto, incluindo a definição de marcos e de pontos de controle, são estabelecidos e mantidos;
- **GPR 6.** Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;
- **GPR 7.** Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;
- **GPR 8.** Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;

- **GPR 9.** Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, de armazenamento e de distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;
- **GPR 10.** Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos;
- **GPR 11.** A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados;
- **GPR 12.** O Plano do Projeto é revisado com os interessados e o compromisso com ele é obtido e mantido;
- **GPR 13.** O escopo, as tarefas, as estimativas, o orçamento e o cronograma do projeto são monitorados em relação ao planejado;
- **GPR 14.** Os recursos materiais e humanos e os dados relevantes do projeto são monitorados em relação ao planejado;
- **GPR 15.** Os riscos são monitorados em relação ao planejado;
- **GPR 16.** O envolvimento das partes interessadas no projeto é planejado, monitorado e mantido;
- **GPR 17.** Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento;
- **GPR 18.** Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas;
- **GPR 19.** Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão.

No processo Gerência de Requisitos, o objetivo é gerenciar os requisitos do produto e dos componentes do produto do projeto e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto. Os resultados esperados para esse processo nos Níveis G e F são [SOFTEX, 2012]:

- **GRE 1.** O entendimento dos requisitos é obtido junto aos fornecedores de requisitos;
- **GRE 2.** Os requisitos são avaliados com base em critérios objetivos e um comprometimento da equipe técnica com estes requisitos é obtido;
- **GRE 3.** A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
- **GRE 4.** Revisões em planos e produtos de trabalho do projeto são realizadas visando à identificação e à correção de inconsistências em relação aos requisitos;
- **GRE 5.** Mudanças nos requisitos são gerenciadas ao longo do projeto.

3.4.2.2. Nível F

No Nível F, o principal foco é agregar processos de apoio à gestão do projeto no que diz respeito à Garantia da Qualidade (GQA) e Medição (MED), bem como aqueles referentes à organização dos artefatos de trabalho por meio da Gerência de Configuração (GCO). Muitas organizações subcontratam etapas do processo de desenvolvimento ou componentes específicos do produto, essa contratação deve ser controlada com mesmo rigor que as questões internas. Os requisitos úteis para esse controle ser feito de forma adequada é definido no processo Aquisição (AQU). Além disso, implantação do processo Gerência de Portfólio de Projetos (GPP) possibilita às organizações gerência mais efetiva dos recursos disponíveis e investimentos realizados visando ao atendimento dos objetivos estratégicos da organização [SOFTEX, 2011b].

O Nível F é composto pelos processos do nível de maturidade anterior (Nível G) acrescidos dos processos GQA, MED, GCO, AQU e GPP. Nesse nível, a implementação dos processos deve satisfazer os atributos de processo do Nível G AP 1.1, AP 2.1 (relatados anteriormente) e AP 2.2 [SOFTEX, 2012]:

- **AP 2.2 Os produtos de trabalho do processo são gerenciados.** Este atributo evidencia o quanto os produtos de trabalho produzidos pelo processo são gerenciados apropriadamente. Os resultados esperados para esse atributo são:
 - ✓ **RAP 11.** Os requisitos dos produtos de trabalho do processo são identificados;
 - ✓ **RAP 12.** Requisitos para documentação e controle dos produtos de trabalho são estabelecidos;
 - ✓ **RAP 13.** Os produtos de trabalho são colocados em níveis apropriados de controle;
 - ✓ **RAP 14.** Os produtos de trabalho são avaliados objetivamente com relação aos padrões, procedimentos e requisitos aplicáveis e são tratadas as não conformidades.

No processo Portfólio de Projetos, o objetivo é iniciar e manter projetos que sejam necessários, suficientes e sustentáveis, de forma a atender os objetivos estratégicos da organização. Esse processo compromete o investimento e os recursos organizacionais adequados e estabelece a autoridade necessária para executar os projetos selecionados. Ele executa a qualificação contínua de projetos para confirmar que eles justificam a continuidade dos investimentos ou podem ser redirecionados para justificar. Os resultados esperados para esse processo no Nível F são [SOFTEX, 2012]:

- **GPP 1.** As oportunidades de negócio, as necessidades e os investimentos são identificados, qualificados, priorizados e selecionados em relação aos objetivos estratégicos da organização por meio de critérios objetivos;
- **GPP 2.** Os recursos e orçamentos para cada projeto são identificados e alocados;
- **GPP 3.** A responsabilidade e a autoridade pelo gerenciamento dos projetos são estabelecidas;
- **GPP 4.** O portfólio é monitorado em relação aos critérios utilizados para a priorização;
- **GPP 5.** Ações para corrigir desvios no portfólio e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão;
- **GPP 6.** Os conflitos sobre recursos entre projetos são tratados e resolvidos, de acordo com os critérios utilizados para a priorização;
- **GPP 7.** Projetos que atendem aos acordos e aos requisitos que levaram à sua aprovação são mantidos e os que não atendem são redirecionados ou cancelados;
- **GPP 8.** A situação do portfólio de projetos é comunicada para as partes interessadas, com periodicidade definida ou quando o portfólio for alterado.

No processo Garantia da Qualidade, o objetivo é assegurar que os produtos de trabalho e a execução dos processos estejam em conformidade com os planos, procedimentos e padrões estabelecidos. Os resultados esperados para esse processo no Nível F são [SOFTEX, 2012]:

- **GQA 1.** A aderência dos produtos de trabalho aos padrões, procedimentos e requisitos aplicáveis é avaliada objetivamente, antes dos produtos serem entregues e em marcos predefinidos ao longo do ciclo de vida do projeto;

- **GQA 2.** A aderência dos processos executados às descrições de processo, de padrões e de procedimentos é avaliada objetivamente;
- **GQA 3.** Os problemas e as não-conformidades são identificados, registrados e comunicados;
- **GQA 4.** Ações corretivas para as não-conformidades são estabelecidas e acompanhadas até as suas efetivas conclusões. Quando necessário, a escala das ações corretivas para níveis superiores é realizado, de forma a garantir sua solução.

Os processos Gerência de Configuração (GCO), Aquisição (AQU) e Medição (MED) não foram relatados por não serem utilizados no desenvolvimento da ferramenta ScrumMps Versão 2.0.

3.5. Considerações Finais

Neste capítulo, foram apresentados os principais conceitos do termo qualidade para contextualizar qualidade de software. Posteriormente, foi relatado como a melhoria dos processos de desenvolvimento de sistemas de software utilizando normas e modelos como a ISO/IEC 12207, ISO/IEC 550343, CMMI-DEV e MPS.BR podem ajudar as empresas a conseguirem melhorar e manter a qualidade do desenvolvimento de seus sistemas de software. Por fim, foi apresentado o MPS.BR, principalmente o Modelo de Referência MPS para Software (MR-MPS-SW) e os seus níveis G e F.

4. FERRAMENTAS EXISTENTES

4.1. Considerações Iniciais

Neste capítulo, é apresentada uma análise das ferramentas de gerência de projetos que apoiam a metodologia ágil Scrum. O objetivo da análise das ferramentas é levantar requisitos para serem desenvolvidos na ScrumMps 2.0. A escolha das ferramentas foi baseada em uma pesquisa realizada na Internet em fóruns, *blogs* e *sites* sobre metodologias ágeis para saber quais são as ferramentas gratuitas mais utilizadas pela comunidade. Dentre as ferramentas encontradas, quatro tiveram destaque pelos recursos apresentados e foram escolhidas para serem analisadas.

As ferramentas escolhidas foram PangoScrum, IceScrum, MeuScrum e FireScrum apresentadas nas Seções 4.2, 4.3, 4.4 e 4.5, respectivamente. Nessas seções, é apresentado breve estudo dessas ferramenta, relatando principais propriedades e recursos disponibilizados. Além disso, uma análise comparativa, para identificar pontos fortes e fracos, é mostrada na Seção 4.6.

4.2. PangoScrum

A ferramenta PangoScrum é uma aplicação Web proprietária desenvolvida pela empresa PangoScrum.com para gerência de projetos utilizando a metodologia ágil Scrum. A empresa é a responsável pela hospedagem da ferramenta, por disponibilizá-la na internet para seus usuários e garantir privacidade e segurança da informação. O *link* disponibilizado para acesso e utilização da ferramenta é: <http://pangoscrum.com>. PangoScrum está na versão beta e disponível para a comunidade para testes e exploração de recursos. De acordo com o termo de sua utilização, o "PangoScrum Beta" tem período limitado para testes. Após esse período, a sua utilização poderá ser cobrada.

Ainda não foi definida data para o fim do período de disponibilização dessa versão, mas a empresa se comprometeu em informar a comunidade com antecedência sobre o fim desse período.

A tela principal da PangoScrum é dividida em quatro abas principais que permitem o acesso às suas funções (Figura 11): *Dashboard*, *Produtos*, *Sprints* e *Colaboradores*. Na aba *Dashboard*, é apresentada uma visão geral dos *Produtos* e das *Sprints* cadastrados. Na aba *Produtos*, pode-se incluir, consultar, alterar e excluir um produto. Na aba *Sprints*, pode-se incluir, consultar, alterar e excluir uma *Sprint*. Na aba *Colaboradores*, pode-se visualizar os colaboradores com acesso à ferramenta e incluir novos colaboradores.

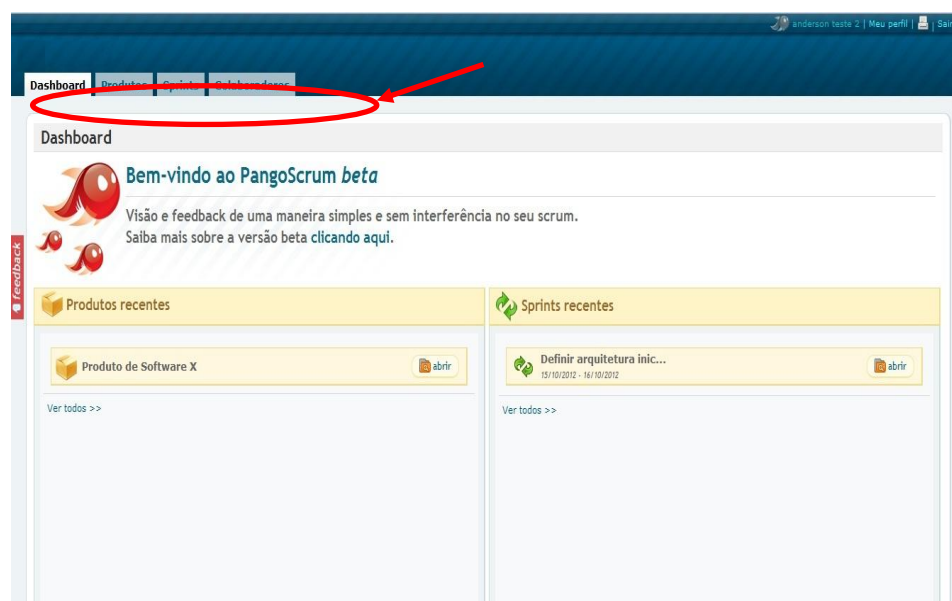


Figura 11 - Tela Dashboard

Após incluir um produto, pode-se gerenciar o *Product Backlog* por criar itens para o produto e acompanhar os itens utilizando três seções (Figura 12): *To-do*, *Em andamento* e *Concluídos*. Posteriormente, esses itens são atribuídos a uma *Sprint*, um valor e uma estimativa são definidos para o item e comentários

podem ser registrados. Duas das desvantagens dessa ferramenta são (i) não utilizar o conceito de *User Stories* e (ii) os itens serem cadastrados com curta descrição (aproximadamente uma linha), tendo que ser utilizados os comentários da tarefa para adicionar mais detalhes.

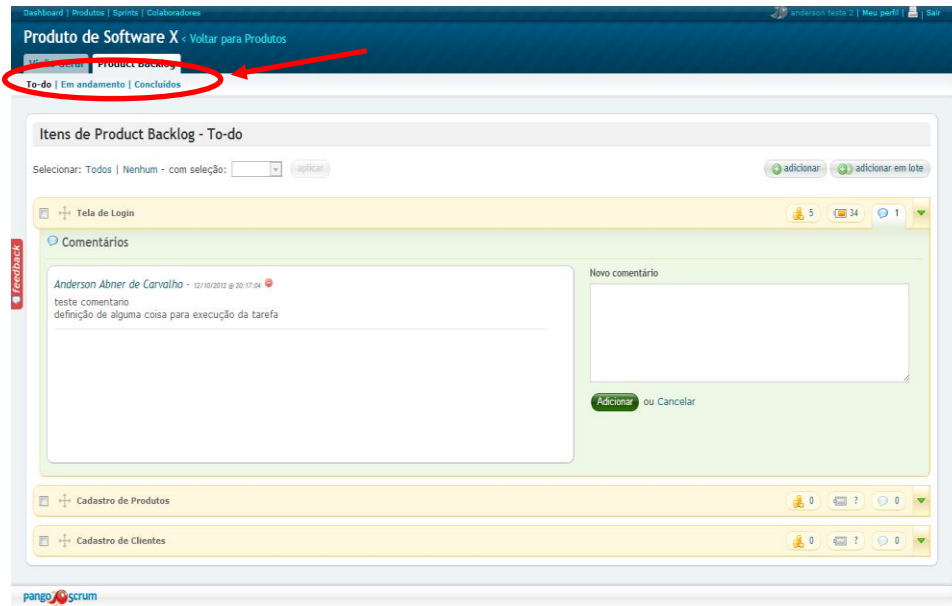


Figura 12 - Tela *Product Backlog* da Ferramenta PangoScrum

Na aba Visão Geral, três telas, uma ao lado da outra, são disponibilizadas no gerenciamento das *Sprints* (Figura 13): i) Resumo; ii) Calendário; e iii) *Burndown*. Na tela Resumo, são apresentadas informações gerais do produto, por exemplo, quantidade de itens, estimativa de pontos dos itens e data de início e de fim da *Sprint*. Na tela Calendário, eventos para *Sprint* podem ser criados. Para isso, deve-se adicionar um dos eventos: *Sprint Planning*, *Sprint Review*, Retrospectiva da *Sprint* e Reunião Diária, em um dia específico. Os eventos *Sprint Planning* e Retrospectiva da *Sprint* são, respectivamente, data de início e de fim da *Sprint*. Porém, não se consegue fazer algum tipo de registro para documentar os eventos. Na tela *Burndown*, o gráfico *Burndown* é apresentado.

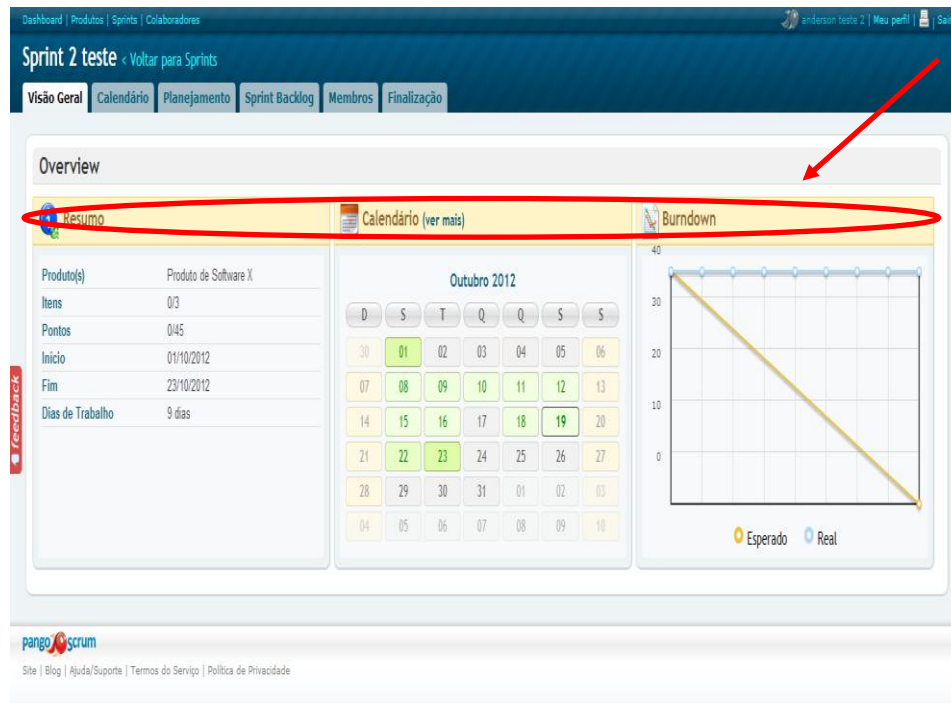


Figura 13 - Tela *Sprint* Visão Geral da Ferramenta PangoScrum

Na aba Planejamento, os itens do *Product Backlog* com o status *To-do* podem ser associados a uma *Sprint*. Duas das desvantagens nessa função são (i) não ser possível vincular um membro da equipe para a tarefa da *Sprint* e (ii) não existir o conceito de *Kanban* ou *Task Board*, o que dificulta visualizar o andamento das tarefas da *Sprint*. Na ferramenta PangoScrum, há os papéis *Product Owner*, *Scrum Master* e Desenvolvedor. O papel *Product Owner* é definido quando o produto é criado, os papéis *Scrum Master* e Desenvolvedor são atribuídos utilizando a aba Membros no gerenciamento de uma *Sprint*. Apenas colaboradores ativos na ferramenta podem ser atribuídos aos papéis. Nessa ferramenta, o cliente não possui relacionamento direto com a equipe de desenvolvimento e não pode interagir com as atividades.

4.3. IceScrum

A ferramenta IceScrum é uma aplicação Web livre, de código aberto e destinada às equipes de desenvolvimento ágil e está disponibilizada para a comunidade sobre as licenças GNU (*Gnu is Not Unix*) Affero GPL (*General Public License*) v3 e LGPL (*Lesser General Public License*) v3. A sua principal patrocinadora é a empresa Kagilum SAS que se situa em Toulouse, no sul da França. A ferramenta foi criada em 2006, conta com 40 colaboradores e possui mais de três mil *downloads*/mês. O *download* da ferramenta e do seu código fonte pode ser realizado utilizando o *link*: <http://www.icescrum.org>.

A ferramenta foi projetada principalmente para a metodologia ágil Scrum, suas principais funções são os pilares do Scrum: *Product Backlog*, *Sprint*, *ScrumMaster* e *Product Owner*. Além dessas funções, a ferramenta possui vários outros recursos baseados em técnicas ágeis como o *Planning Poker*. Para começar a utilizar os seus recursos, o usuário deve incluir um novo projeto. Nessa inclusão, pode-se determinar se o projeto será privado ou público. Caso seja privado, apenas usuários que fazem parte do Time do projeto têm acesso. A escolha dos membros é feita na definição do projeto e pode ser alterada posteriormente. Os papéis que a equipe do projeto pode assumir são: *Product Owner*, *ScrumMaster*, Membro da Equipe e *Stakeholder*. Na definição do projeto, deve-se definir algumas práticas utilizadas no decorrer do projeto, por exemplo:

- Optar por um dos pacotes para estimação: Fibonacci ou Inteiros. Se a opção for Fibonacci, uma sequência de fibonacci é utilizada como peso dos pontos de uma *User Story* ou tarefa. Se a opção for Inteiros, o peso pode ser qualquer inteiro entre 1 e 100, inclusive;
- Definir se os finais de semana nos gráficos serão apresentados;

- Escolher se o sistema automaticamente modificará o *status* de uma *User Story* para "Concluída" quando as tarefas estiverem concluídas;
- Escolher se o sistema criará automaticamente uma tarefa vazia em uma *User Story* quando a iteração é ativada;
- Escolher se o sistema associará automaticamente um membro da equipe ao criar/iniciar tarefa;
- Definir se as tarefas recorrentes serão exibidas;
- Definir se as tarefas urgentes serão exibidas;
- Definir limite de tarefas urgentes em andamento;
- Definir duração das iterações em dias.

Ainda na definição do projeto, deverão ser definidas a data de início do projeto e da primeira iteração (*Sprint*) e a visão do projeto. A tela inicial da ferramenta após criação de um projeto é exibida na Figura 14. Nessa tela, é exibido um painel (*dashboard*) com diversas informações do projeto, tais como, gráfico *Burndown* do projeto, descrição e visão do projeto, descrição da última *Sprint* finalizada e da última retrospectiva e as últimas atividades dos membros do Time do projeto. Após a criação de um projeto, *User Stories* podem ser criadas na seção de Área Livre. Os membros do projeto podem criar *User Stories*, porém apenas o usuário *Product Owner* do projeto pode aceitá-las e movê-las para o *Product Backlog*.

Na tela Planejar Entrega, podem ser escolhidas as datas de entrega das *Sprints*. Depois dessa escolha, um gráfico com a linha do tempo do projeto é exibido (Figura 15). Na tela Planejamento de Iteração, a *Sprint* corrente pode ser gerenciada. Para isso, o usuário *Product Owner* ou *Scrum Master* devem fazer o cadastro de tarefas. As tarefas podem ser de dois tipos: i) tarefas recorrentes; e ii) tarefas urgentes. Dois problemas encontrados na ferramenta são (i) não se pode vincular as tarefas a uma *User Story* do *Product Backlog* e (ii) as tarefas

não podem ser associadas a um membro usuário. Com as tarefas criadas, pode-se gerenciá-las utilizando o Kanban com o recurso de "arrastar e soltar" (Figura 16). A ferramenta permite a geração dos gráficos *Burndown* da *Sprint* corrente e contém uma página na qual a reunião de retrospectiva da *Sprint* pode ser documentada. Uma desvantagem da ferramenta é a instalação de *plugins* que se associam a ferramenta para utilizar alguns recursos, por exemplo, o *Planning Poker*.

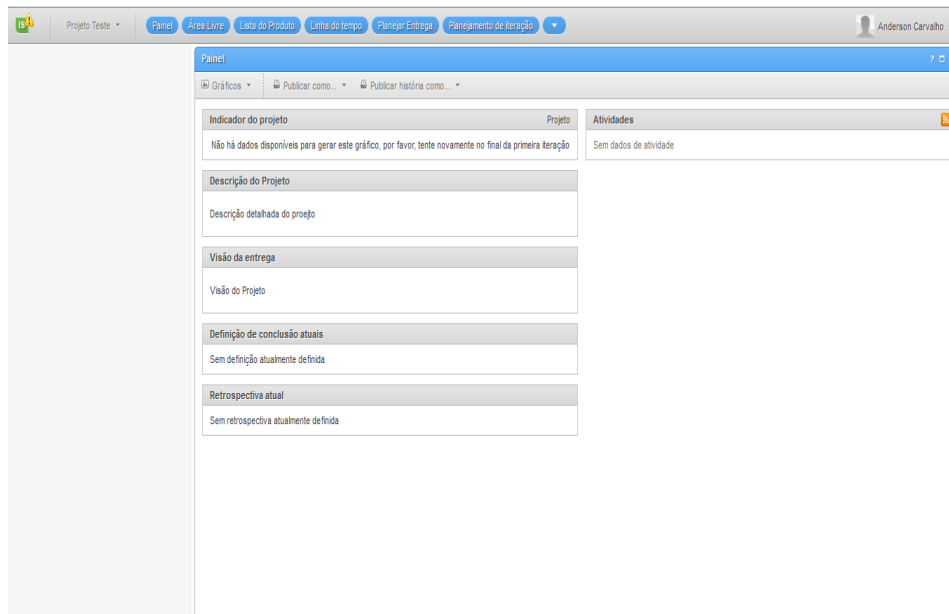


Figura 14 - Tela Inicial da Ferramenta iceScrum

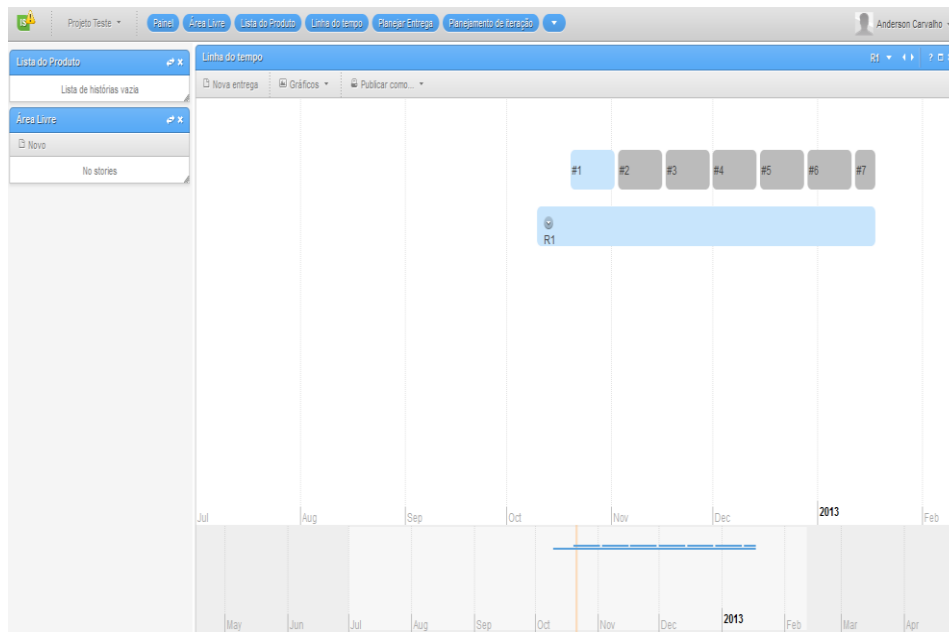


Figura 15 - Tela Linha do Tempo da Ferramenta iceScrum

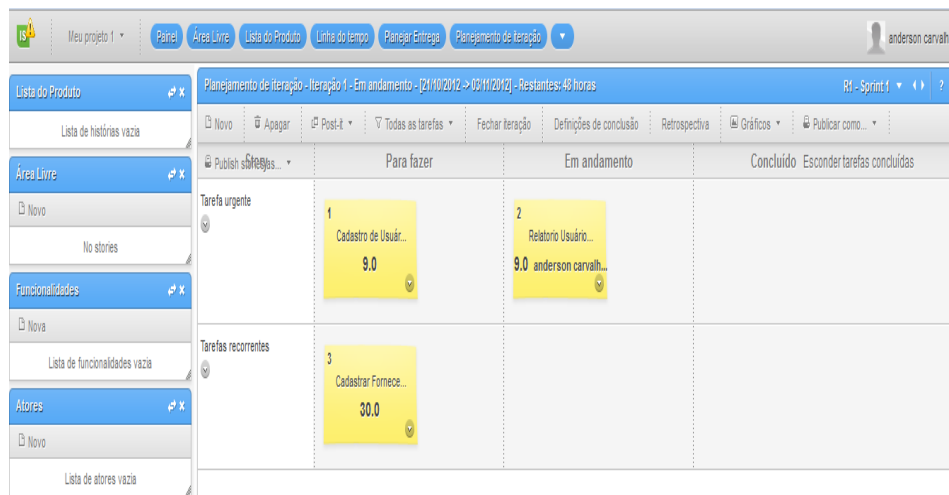


Figura 16 - Tela Kanban da Ferramenta iceScrum

4.4. MeuScrum

A ferramenta MeuScrum é uma aplicação Web proprietária desenvolvida pela empresa Interart Tecnologia para gerência de projetos baseada na metodologia ágil Scrum. Essa empresa é a responsável pela hospedagem da ferramenta, por disponibilizá-la na internet para seus usuários e por garantir a privacidade e segurança da informação. O *link* disponibilizado para acesso e utilização da ferramenta é: <http://www.meuscrum.com>. Atualmente, a utilização dos recursos do MeuScrum é feita de forma gratuita, porém, de acordo com o termo de uso da ferramenta, a Interart Tecnologia poderá cobrar qualquer quantia pelo seu uso e/ou encerrar, a qualquer tempo, a prestação do serviço. Para isso, ela deve comunicar com antecedência mínima de 60 dias ao usuário para ele decidir se deseja ou não continuar a utilizar a ferramenta MeuScrum.

No MeuScrum, são oferecidos recursos para auxiliar na gerência de projetos de equipes que utilizam o Scrum. Na tela inicial (Figura 17), é exibido um *Dashboard* com uma lista dos projetos favoritos para facilitar o acesso e um pequeno tutorial com sete itens informando ao usuário o modo de utilizar a ferramenta. Na tela Listagem de Projetos (Figura 18), são disponibilizados ícones para: i) Editar Projetos; ii) Visualizar *Kanban* de uma *Sprint*; iii) Visualizar Gráfico *Burndown* de uma *Sprint*; iv) Tornar Projeto Favorito; v) Incluir *Sprints* para um projeto; vi) Incluir *User Stories*; vii) Definir Membros do *Time Scrum*; viii) Editar Colunas do *Kanban* de um projeto; e ix) Arquivar Projetos.

Ao acessar o *Kanban* (Figura 19), *User Stories* e tarefas podem ser incluídas para o projeto. Cada tarefa é vinculada a uma *User Stories* e a um membro do *Time Scrum*. O gerenciamento das tarefas é feito utilizando o recurso "arrastar e soltar", movendo as tarefas entre as colunas do *Kanban*. Por

padrão, são disponibilizadas as colunas "A Fazer", "Fazendo", "Feito" e "Impedimento", porém podem ser adicionadas outras colunas.

meuscrum
online kanban tool

Boa Tarde, Anderson Carvalho Dashboard Projetos Minha Conta Ajuda Sair

Dashboard

Sua Área de Trabalho

Instabilidade momentânea 6/9/2012
Prezado usuário, temos verificado alguns problemas eventuais de instabilidade no sistema. Já identificamos a causa do problema e estamos trabalhando para resolvê-lo o mais brevemente possível. Contamos com a sua compreensão.

Projetos Favoritos Todos os projetos

Nome	Descrição do Projeto
Projeto Teste	Teste da ferramenta

Por onde começar?

O MeuScrum é um sistema simples de usar e bastante intuitivo. Para iniciar o seu uso, siga os passos abaixo:

1. Acesse a página de projetos
2. Crie um projeto, clicando no botão "Novo Projeto"
3. Crie as colunas necessárias para o quadro de Kanban (ToDo, Done, etc.), clicando no ícone
4. Crie um ou mais sprints, clicando no ícone
5. Crie as histórias do projeto, clicando no ícone
6. Opcionalmente, defina os usuários que terão acesso ao projeto, clicando no ícone
7. Em seguida, você já poderá acessar o Kanban do projeto (clicando no nome do mesmo na listagem), onde poderá incluir as tarefas para cada usuário participante

Desenvolvido por interQI
versão 2.0

Figura 17 - Tela *Dashboard* da Ferramenta MeuScrum

meuscrum
online kanban tool

Boa Noite, Anderson Carvalho Dashboard Projetos Minha Conta Ajuda Sair

Projetos

Listagem de Projetos

Novo Projeto

Ativos

Nome	Descrição do Projeto	Data Cadastro	Data Arquivo	Ativo	Projetos
Projeto Teste	Teste da ferramenta	27/08/2012 17:56	-		
Projeto tools	Descrição detalhada do projeto.	20/10/2012 17:30	-		

Legenda: Kanban Editar Burndown Favorito Sprints Histórias Equipe Colunas Arquivar

Desenvolvido por interQI
versão 2.0

Figura 18 - Listagem de Projetos da Ferramenta MeuScrum

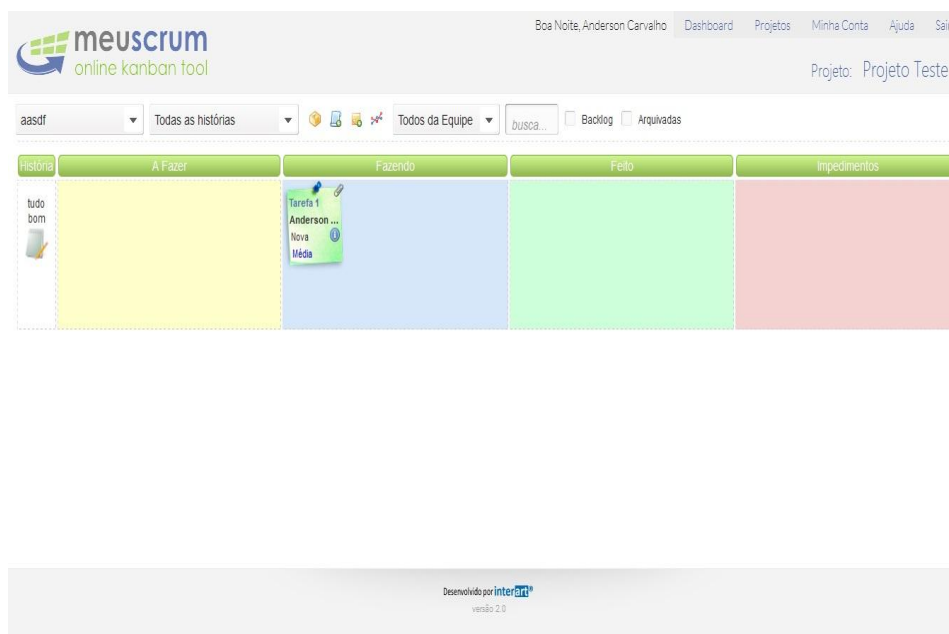


Figura 19 - Tela Kanban da Ferramenta MeuScrum

Os papéis existentes na ferramenta MeuScrum são: i) Administrador; ii) *Scrum Master*; iii) Gerente; iv) *Product Owner*; iv) Colaborador; e v) *Stakeholder*. Nessa ferramenta, o cliente tem permissão para criar histórias e visualizar o andamento das *Sprints* o que o aproxima da equipe de desenvolvimento e garante sua participação no processo de desenvolvimento do projeto. Na ferramenta MeuScrum, não há funções para documentar eventos de uma *Sprint*, tais como, *Daily Scrum* e *Retrospectiva da Sprint*, e para estimar as tarefas, por exemplo, *Planning Poker*.

4.5. FireScrum

A ferramenta FireScrum é uma aplicação Web livre, de código aberto e disponibilizada sob a licença GPL. O código fonte e um tutorial para instalação e configuração são disponíveis no *site*: <http://sourceforge.net/projects/firescrum/>. No FireScrum, o objetivo é apoiar a gestão de projetos em empresas de software

que utilizam princípios de gestão ágil em seus projetos. FireScrum foi desenvolvido durante o programa de mestrado em engenharia de software da CESAR.EDU e tornou-se um produto da INES (Instituto Nacional de Ciência e Tecnologia para Engenharia de Software). Essa ferramenta contém diversos recursos para a gerência de projetos de equipes Scrum, por exemplo: i) Gerenciar *Product Backlog*; ii) Gerenciar *Sprints*; iii) Gerenciar usuários; iv) Gerar relatórios com gráfico *Burndown* das *Sprints* e do produto; v) *Bug Tracking*; vi) Gerenciar realização de *Daily Meeting*; vii) *Planning Poker*; e viii) *Task Board (Kanban)*.

Para incluir um produto, o usuário deve fornecer nome e descrição do produto e a unidade a ser utilizada nas estimativas das tarefas. Após a inclusão, uma listagem *Product Backlog* é exibida na tela inicial da ferramenta. Nessa listagem, o usuário pode incluir itens (*User Stories*) e tarefas. Na inclusão de tarefas, são definidos o nome e a descrição da tarefa, o membro responsável e a estimativa. Em seguida, as tarefas são vinculadas a um item do *Product Backlog*. Ao incluir uma *Sprint*, deve-se realizar seu planejamento, com a definição de data de início e de fim da *Sprint* e o fornecimento das metas e de uma descrição para a *Sprint*. Após a inclusão, itens do *Product Backlog* podem ser selecionados para serem incluídos no *Sprint Backlog*, utilizando o recurso "arrastar e soltar". As tarefas relacionadas ao item são automaticamente passadas para a *Sprint* selecionada. Um exemplo de movimentação do item "itembacklog1" para a *Sprint* "S1" é apresentado na Figura 20.

Os usuários com perfil Administrador podem adicionar outros usuários no sistema e vinculá-los a um produto. Com esse vínculo, o usuário tem acesso às funções referentes ao produto. No FireScrum, não há distinção entre os papéis do Scrum, ou seja, qualquer membro do Time podem fazer qualquer atividade sem restrições por papel. Os relatórios gerados pela ferramenta são gráficos

Burndown do projeto e da *Sprint*, considerando as datas de lançamento das *Sprints* e as tarefas concluídas. São disponibilizados na ferramenta FireScrum alguns módulos elaborados para equipes de desenvolvimento distribuídas que utilizam o Scrum, por exemplo:

- *Task board*. Nesse módulo, é disponibilizada uma tela (Figura 21) na qual se pode gerenciar tarefas de uma *Sprint*. O *status* das tarefas é modificado utilizando o recurso de "arrastar e soltar". Uma tarefa pode ser alterada nessa tela;
- *Planning Poker*. Nesse módulo, é disponibilizada uma tela (Figura 22) na qual se pode estimar os itens do *Backlog*. O usuário define as cartas a serem utilizadas no *Planning Poker* e cria uma sessão para que os outros usuários possam acessar e participar da estimativa. As discussões sobre as estimativas são realizadas por meio de um *chat* entre os usuários;

The screenshot displays the FireScrum interface with two main panels. The left panel, titled 'Product Backlog', contains a table with columns: 'Uncommitted Backlog Items', 'Backlog Effort', 'Task Hours', and 'Business Value'. It lists 'Itembacklog2' (Effort: 25, Hours: 6) and 'Itembacklog1' (Effort: 50, Hours: 5). The right panel, titled 'Committed Backlog', shows a table with columns: 'Committed Backlog Items', 'Backlog Effort', 'Task Hours', and 'Business Value'. It lists 'Itembacklog1' (Effort: 50, Hours: 5) under the 'S1' sprint. The interface also includes a menu bar (File, User, Reports, Modules) and a toolbar (S1, S2).

Figura 20 - Movimentação de um Item do *Product Backlog* para uma *Sprint* na Ferramenta FireScrum

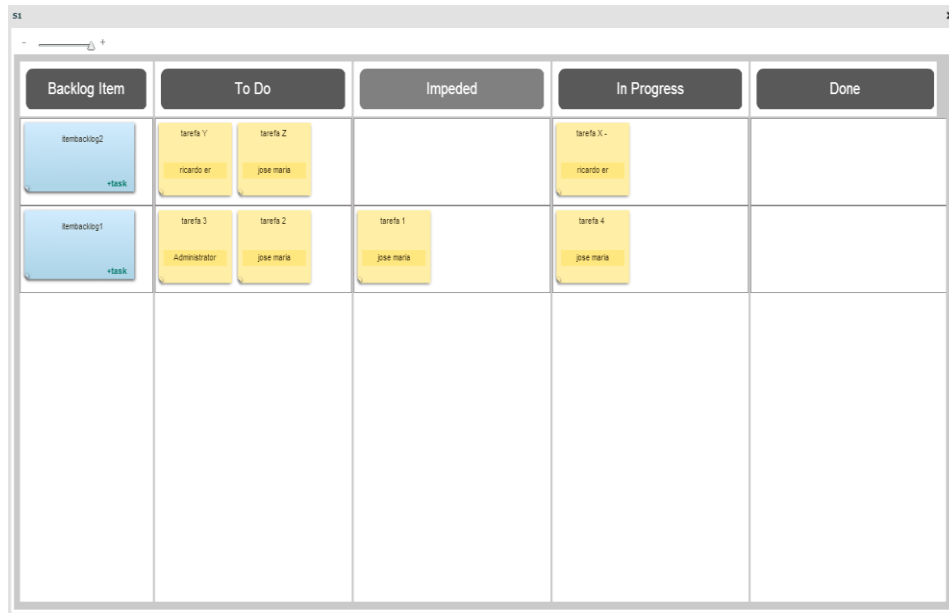


Figura 21 - Módulo *Task Board* da Ferramenta FireScrum

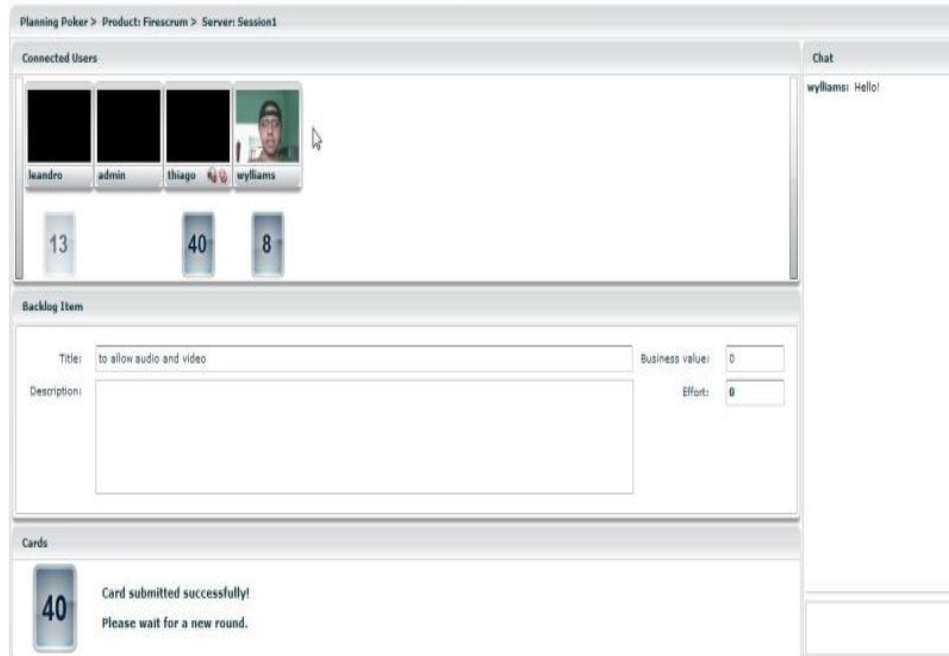


Figura 22 - Módulo *Planning Poker* da Ferramenta FireScrum

- *Daily Meeting*. Nesse módulo, é disponibilizada uma sessão em que os membros do Time acessam e comunicam-se utilizando um *chat* para efetivar as reuniões diárias do Scrum. Posteriormente, eles podem acessar o histórico dessas reuniões;
- *Test Module*. Nesse módulo, o usuário define suítes, planos e casos de testes para os itens do *Product Backlog*;
- *Bug Tracking*. Nesse módulo, há a gerência de *bugs* encontrados durante os testes do projeto.

4.6. Quadro Comparativo

Nesta seção, é apresentada uma análise comparativa das quatro ferramentas apresentadas neste capítulo: PangoScrum, IceScrum, MeuScrum e FireScrum. Um quadro comparativo, contendo os principais recursos e características das ferramentas, é exibido na Tabela 2. Pôde-se constatar que as ferramentas analisadas possuem vários recursos semelhantes para a gerência ágil de projetos utilizando Scrum. As ferramentas FireScrum e IceScrum destacaram-se por apresentar recursos voltados para equipes de desenvolvimento distribuídas e por serem *open source*.

Tabela 2 - Quadro Comparativo das Ferramentas Analisadas

Características/Recursos	PangoScrum	IceScrum	MeuScrum	FireScrum
Aplicação Web	X	X	X	X
<i>Open Source/Free</i>		X		X
SaaS (<i>Software as a Service</i>)	X		X	
<i>Dashboard</i>	X	X	X	
Gerenciamento <i>Product Backlog</i>	X	X		X
Planejamento e gerenciamento das Sprints	X	X	X	X

Tabela 2 - Quadro Comparativo das Ferramentas Analisadas (cont.)

Características/Recursos	PangoScrum	IceScrum	MeuScrum	FireScrum
<i>User Stories</i>		X	X	X
Gráfico <i>Burndown</i> do Projeto		X	X	X
Gráfico <i>Burndown</i> das <i>Sprints</i>	X	X		X
<i>Kanban / Task Board</i>		X	X	X
Papeis do Scrum (<i>Product Owner, Scrum Master</i> e Desenvolvedor)	X	X	X	
Cliente pode interagir com as atividades na ferramenta		X	X	
Associar tarefas a um usuário			X	X
<i>Planning Poker</i>		X		X
Registro das Reuniões Diárias do Scrum				X
Registro das Reuniões de Retrospectiva		X		
Definir Visão do Projeto		X		
Módulo de <i>Bug Tracker</i>				X

4.7. Considerações Finais

Neste capítulo, foi realizado um estudo de quatro ferramentas de apoio à gerência de projetos que utilizam a metodologia ágil Scrum. A análise e a comparação dos recursos das ferramentas foi importante para servir como guia para a identificação e o desenvolvimento de recursos para serem contemplados na ferramenta ScrumMps 2.0. Por exemplo, o gráfico *Burndown*, o registro das reuniões diárias e de reuniões de retrospectiva do Scrum, o módulo *Bug Tracker* e a definição da Visão do projeto.

5. SCRUMMPS 1.0

5.1. Considerações Iniciais

Este capítulo está destinado a apresentar um estudo da ferramenta ScrumMps 1.0 [Reis, 2011], evoluída neste trabalho para ScrumMps 2.0. O motivo para criação da ScrumMps se deve ao fato da ausência de ferramentas que abordem o conhecimento da metodologia ágil Scrum juntamente com o MPS.BR, a ideia de unir esses dois conceitos foi criar uma ferramenta que aumentasse a qualidade do software brasileiro alinhado às tendências do mercado. ScrumMps 1.0 foi desenvolvida para Web utilizando as linguagens de programação Adobe Flex¹ e Java². O intuito de criar uma aplicação Web foi por causa da mobilidade que a internet proporciona e por permitir que o sistema esteja disponível em tempo integral. Na ScrumMps 1.0, foram desenvolvidos recursos para o gerenciamento de projetos, tais como, acompanhamento de tarefas, manutenção de cadastro de usuários e de estórias abordando os conceitos de MPS.BR.

O Diagrama de Casos de Uso da ScrumMps 1.0 é apresentado na Seção 5.2. Um *roadmap* na ScrumMps 1.0, mostrando as principais telas desenvolvidas, é destacado na Seção 5.3. Os resultados esperados pelo MPS.BR no Nível G apoiados por ScrumMps 1.0 como forma de gerenciar o projeto são elencados na Seção 5.4.

5.2. Funcionalidade

O Diagrama de Caso de Uso é apresentado na Figura 23 [Reis, 2011]. O ator Cliente pode realizar os casos de uso: i) Cadastrar História; ii) Acompanhar Projeto. O ator Desenvolvedor relaciona-se com o ator Cliente por herança o que

¹ <http://www.adobe.com/br/products/flex.html>.

² http://www.java.com/pt_BR/about

significa que, além de poder realizar os casos de uso: i) Cadastrar Tarefas; e ii) Acompanhar Tarefas, ele pode realizar os casos de uso do ator Cliente. O ator *Scrum Master* relaciona-se com o ator Desenvolvedor por herança o que significa que, além de poder realizar os casos de uso: i) Cadastrar Projetos; ii) Cadastrar Funcionários; e ii) Planejar *Sprints*, ele pode realizar os casos de uso do ator Desenvolvedor.

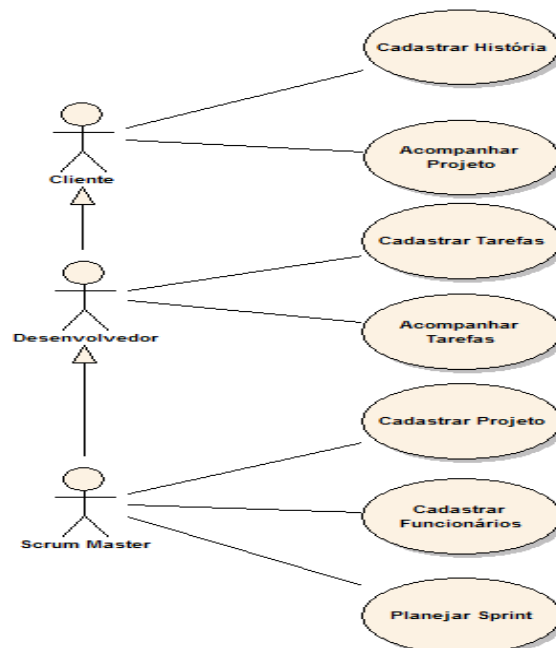


Figura 23 - Diagrama de Casos de Uso - ScrumMps 1.0 (Fonte: [Reis, 2011])

Após autenticar-se no ScrumMps 1.0, o ator Cliente tem a opção de incluir/alterar uma História a ser aprovada, posteriormente, pelo *Product Owner* para entrar no *Product Backlog* e acompanhar o andamento do projeto. O ator Desenvolvedor, após autenticar-se, pode modificar o *status* da tarefa. O *Scrum Master* possui papel importante no sistema, pois possui permissão para (i) incluir/alterar um projeto, (ii) manter o *Product Backlog* juntamente com o *Product Owner*, (iii) fazer o planejamento das *Sprints* e (iv) dividir as histórias em tarefas, destinando-as aos desenvolvedores.

5.3. Principais Funções

As funções foram desenvolvidas considerando a utilização da metodologia ágil Scrum e apoiando os conceitos do Nível G do MPS.BR. As principais funções desenvolvidas foram [Reis, 2011]:

- Manter Cadastro de *Product Backlog*.** Nesta função, pode-se manter cadastro das histórias do projeto utilizadas durante o ciclo de desenvolvimento do produto (Figura 24). Para criar uma história, o usuário deve selecionar o botão localizado no canto superior direito chamado "Nova História". O *Product Backlog* é mantido pela equipe de desenvolvimento e pelos *Product Owners*, encurtando a interação entre ambos. Quando uma história é incluída, ela recebe o *status* "Pendente". Após aprovação do *Product Owner* ou do *Scrum Master*, o seu *status* muda para "Aprovada" e pode ser incluída em uma *Sprint*. O *Product Owner* e o *Scrum Master* podem rejeitar histórias;

Fábrica de Carros					
Lista de Histórias					
Nova História					
Mostrar Rejeitados					
Pendentes (0)					
ID	Nome	%	🚩	★	
Aprovadas (2)					
ID	Nome	%	🚩	★	
3	historia 3	0	4	0	
1	Construir as Rodas	0	3	4	
Rejeitadas (1)					
ID	Nome	%	🚩	★	
2	23423423	24,98	4	3	

Figura 24 - *Product Backlog*

- Realizar Planejamento das *Sprints*.** Nessa função, podem-se realizar a inclusão de *Sprints* e realizar o registro das reuniões de *Scrum Planning 1* e *Scrum Planning 2*. No registro das reuniões, é feita a seleção das histórias do *Product Backlog* a serem desenvolvidas na *Sprint* (Figura 25). Após o

planejamento da *Sprint*, o *Scrum Master* pode incluir as tarefas da *Sprint* e vinculá-las a um desenvolvedor. Cada tarefa tem uma história, apenas histórias do *Sprint Backlog* podem ser relacionadas a uma tarefa;

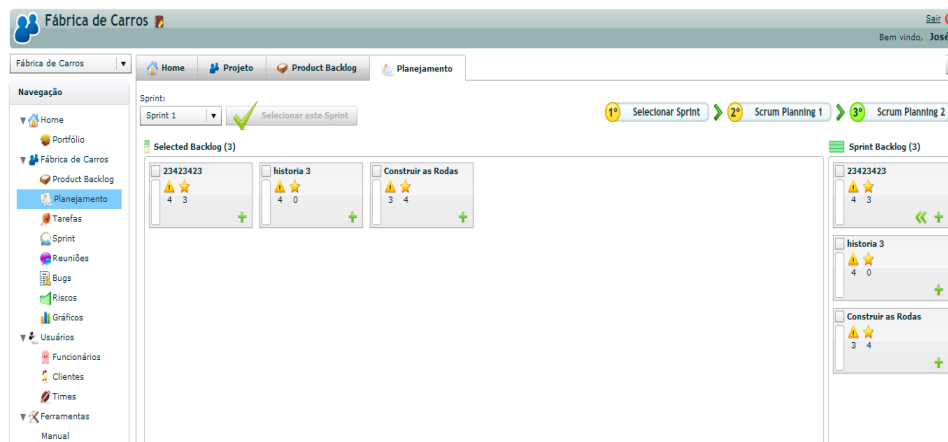


Figura 25 - Planejamento das Sprints

- **Gerenciar Status das Tarefas.** Nessa função, pode-se gerenciar as tarefas de uma *Sprint* e os desenvolvedores podem alterar o *status* de uma tarefa (Figura 26). As colunas do *Kanban* são: i) "Não Iniciado"; ii) "Andamento"; iii) "Verificar"; e iv) "Finalizado". Utilizando o recurso "arrastar e soltar", o usuário seleciona uma tarefa e a solta na coluna desejada.

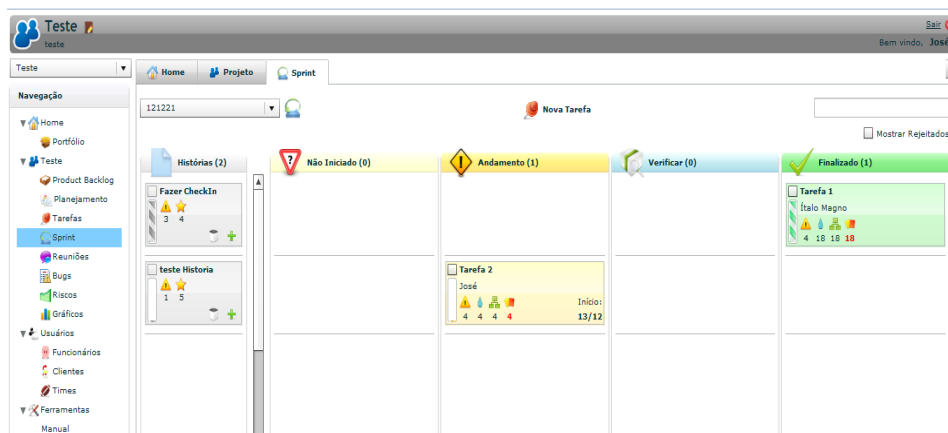


Figura 26 - Kanban

5.4. Utilização do MPS.BR

O MPS.BR é adotado na ferramenta como forma de gerenciar o projeto. De acordo com o estudo das funções desenvolvidas na ScrumMps 1.0, foi identificado o apoio dos resultados esperados pelo MPS.BR no Nível G apresentados na Tabela 3.

Tabela 3 - Resultados Esperados (Fonte: [Reis, 2011])

Resultado Esperado	Implementação
GPR 3	Os <i>stakeholders</i> tem acesso ao andamento do projeto, tarefas realizadas e histórias criadas pelos clientes utilizando o <i>menu Sprint</i> de ScrumMps 1.0. Os desenvolvedores, gerentes e clientes têm visão das fases do ciclo de vida do projeto.
GPR 4	ScrumMps 1.0 possui suporte para armazenar projetos criados, gerando uma base de dados histórica para calcular esforço e custo de projetos futuros.
GPR 9	Os usuários podem acessar informações relevantes sobre o projeto de forma segura, associados ao seu perfil de acesso, e manipular informações de forma independente.
GPR 12	Com ScrumMps 1.0, pode-se apresentar o andamento do projeto aos integrantes com o intuito revisar o planejamento e minimizar riscos.
GPR 13	Em ScrumMps 1.0, são fornecidas informações, tais como, tarefas, estimativas, orçamento e cronograma do projeto, que possibilitam ao gerente de projetos monitoração dos itens a fim de detectar problemas e corrigi-los.

5.5. Considerações Finais

Neste capítulo, foram apresentadas as funções desenvolvidas na ScrumMps 1.0, bem como as suas principais telas. Essas funções foram mostradas utilizando o Diagrama de Casos de Uso. Além disso, foi mostrado como o MPS.BR foi utilizado na conceituação de algumas dessas funções. Após o estudo realizado na ScrumMps 1.0, foram identificados pontos que poderiam ser melhorados considerando o objetivo de construir uma aplicação *Web* para

gerenciar projetos unindo práticas do Scrum e processos do MPS.BR. No próximo capítulo, são apresentadas as funções desenvolvidas na ScrumMps 2.0.

6. SCRUMMPS 2.0

6.1. Considerações Iniciais

Neste capítulo, são abordadas as decisões tomadas para a evolução da ferramenta ScrumMps 2.0. Funções foram incluídas na ferramenta referentes à metodologia ágil Scrum e que apoiam os processos Gerência de Projeto e Gerência de Portfólio de Projetos definidos nos níveis G e F do MPS.BR. Durante a etapa de desenvolvimento, foram identificados alguns erros na ScrumMps 1.0. Esses erros foram corrigidos e são relatados.

Questões técnicas da implementação das funções desenvolvidas são abordadas na Seção 6.2. A funcionalidade e as principais telas desenvolvidas na ScrumMps 2.0 são apresentadas na Seção 6.3. Detecção dos resultados esperados de alguns processos dos Níveis G e F do MPS.BR identificados na ferramenta são relatados na Seção 6.4.

6.2. Modelagem e Arquitetura

Nesta seção, são apresentados o Diagrama de Casos de Uso, o Modelo Relacional do Banco de Dados, o Diagrama Arquitetural e a Implementação da ScrumMps 2.0.

6.2.1. Diagrama de Casos de Uso

Nesta seção, são apresentados os casos de uso da ScrumMps 2.0 (Figura 27). O ator Cliente pode inserir e acompanhar o andamento dos *Bugs* que ele relatou e visualizar o gráfico *Burndown* das *Sprints* para acompanhar o seu andamento. O ator Desenvolvedor tem acesso às funções do ator Cliente e pode inserir/acompanhar os riscos do projeto e registrar as reuniões das *Sprints*. Os atores *Scrum Master* e *Product Owner* têm acesso a todas as funções disponíveis

na ScrumMps 2.0. Além dos casos de uso realizados pelo ator Desenvolvedor, esses dois atores podem avaliar e analisar o portfólio de projetos.

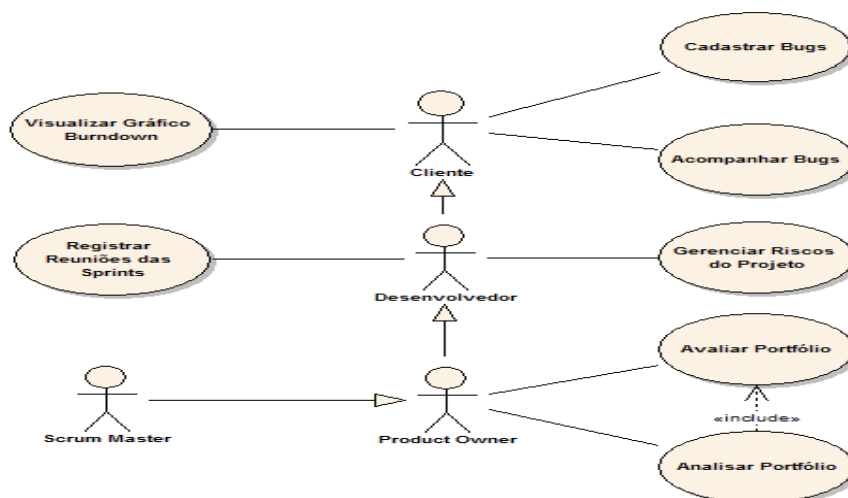


Figura 27 - Diagrama de Casos de Uso

6.2.2. Modelagem de Dados

Na ScrumMps 2.0, foi mantida a escolha do sistema gerenciador de banco de dados MySQL³ por facilitar o desenvolvimento das novas funções, incluindo as novas tabelas e os novos relacionamentos. Algumas tabelas da ScrumMps 1.0 foram modificadas para se adequarem aos novos requisitos. O Modelo Relacional é apresentado na Figura 28. Para facilitar o entendimento e a visualização, algumas tabelas e alguns relacionamentos da ScrumMps 1.0 foram ocultados por não ter ligação com as funções desenvolvidas na nova versão. Por exemplo, a tabela "Bug" tem uma relação 1xN com a tabela "Projeto", o que significa que cada *bug* cadastrado deve ser de um determinado projeto. A relação entre as entidades "Projeto" e "Critério Avaliação" é NxM, pois um projeto pode ter vários critérios para ser avaliado e um critério de avaliação pode ser utilizado para avaliar vários projetos.

³ <http://www.mysql.com/>

Por isso, houve a necessidade de criar uma tabela intermediária ("proj11_projeto_avaliacao"). Nessa tabela, foi adicionado o campo "Valor" em que é armazenada a avaliação feita pelo usuário para um projeto e critério(s) específicos. Uma questão que merece destaque é as entidades terem relação direta ou indireta com a entidade "Empresa" pelo campo "IDTPES_EMP". Isso acontece porque a arquitetura ScrumMps 2.0 foi construída para suportar o uso da ferramenta por mais de uma empresa ao mesmo tempo utilizando apenas uma base de dados.

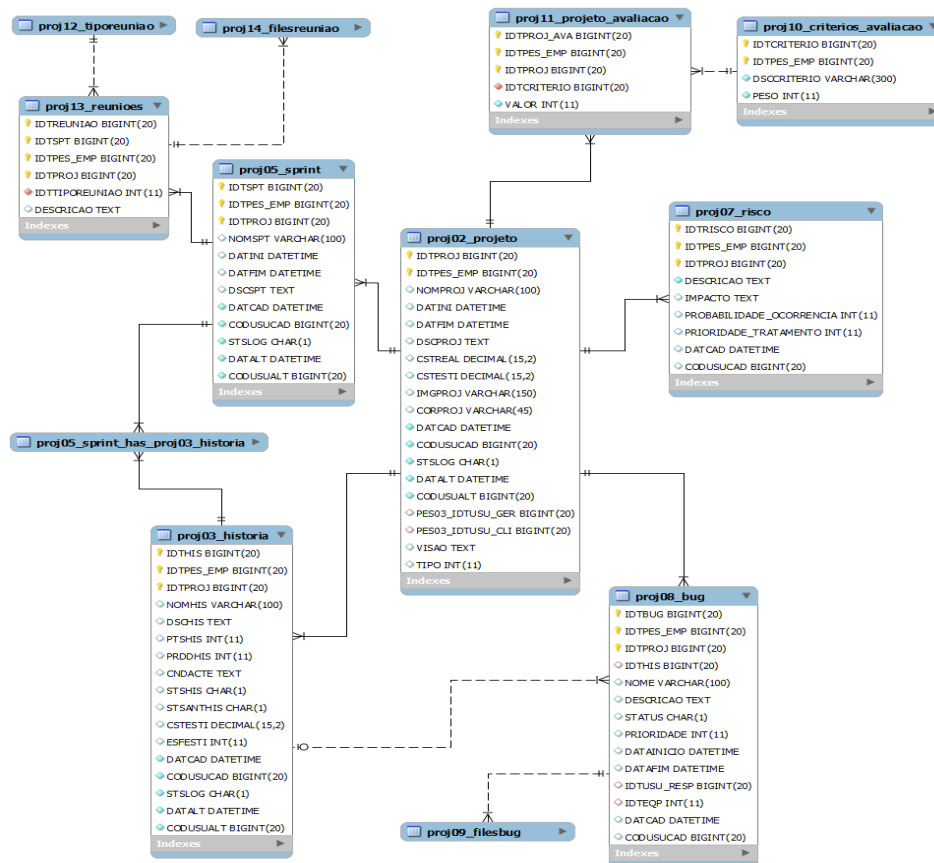


Figura 28 - Modelo Relacional

6.2.3. Diagrama Arquitetural

As tecnologias utilizadas para o desenvolvimento da ScrumMps 2.0 foram Java (*Back-end*) e Adobe Flex (*Front-end*) por permitir a construção de uma interface Web rica (*Rich Internet Application - RIA*) para o usuário final. Além disso, foram utilizados os *frameworks* Sprint Flex⁴ e o BlazeDS⁵ por facilitar a comunicação entre Java e Adobe Flex. O Diagrama Arquitetural simplificado da ScrumMps 2.0 é apresentado na Figura 29. O pacote "Cliente" possui o componente ".swf" com os subcomponentes "view" e "entidades" e o pacote "Servidor" possui os componentes "java" com os subcomponentes "serviços" e "entidades", "persistência" e "jdbc".

O usuário utiliza um *web browser* para acessar a ScrumMps 2.0. Após a requisição do usuário, o servidor Web da aplicação retorna um arquivo .swf para o *web browser* que utiliza o *plugin* Adobe Flash Player⁶ para exibir o arquivo. Caso o *web browser* utilizado não possua o *plugin*, são exibidos uma mensagem da necessidade da sua instalação e um *link* para a página de *download*. O acesso ao banco de dados é feito utilizando classes .java para realizar as operações de manutenção (consultar, alterar, incluir e excluir os dados).

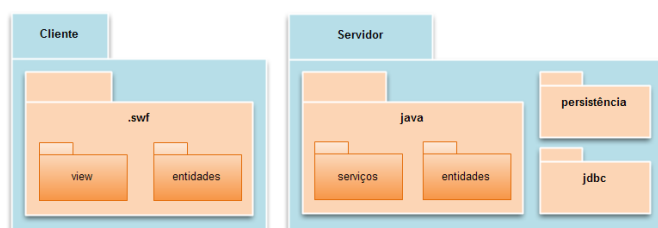


Figura 29 - Diagrama Arquitetural (Fonte: [Reis, 2011])

⁴ <http://www.springsource.org/spring-flex>

⁵ <http://sourceforge.net/adobe/blazeds>

⁶ <http://get.adobe.com/br/flashplayer/>

6.3. Projeto Detalhado e Implementação

Nesta seção, é apresentada a funcionalidade da ScrumMps 2.0 a fim de mostrar o funcionamento de itens importantes para a metodologia ágil Scrum e os processos do modelo MPS.BR.

6.3.1. Gerência de Portfólio de Projetos

No processo Gerência de Portfólio de Projetos, o propósito é iniciar e manter projetos necessários, suficientes e sustentáveis, de forma a atender os objetivos estratégicos da organização. A gerência de portfólio envolve atividades relacionadas à gerência de projetos de uma organização, cujas atividades estão relacionadas com a gerência da carteira de projetos. Isto engloba as atividades de seleção dos projetos que comporão a carteira, bem como análise, ao longo de sua execução, para determinar se continuam viáveis e adequados em relação aos motivos pelos quais foram aprovados [SOFTEX, 2011b]. Foram desenvolvidas duas funções na ScrumMps 2.0 relacionadas ao processo Gerência de Portfólio de Projetos presente no Nível F do MPS.BR, as quais apenas os atores *Product Owner* e *Scrum Master* têm acesso:

- **Avaliação do portfólio.** Os projetos ou oportunidades de negócio são avaliados com base em critérios definidos pela organização (Figura 30). Em geral, esses critérios estão relacionados a, por exemplo [SOFTEX, 2011b]: i) retorno sobre o investimento; ii) alinhamento com os planos estratégicos e táticos; iii) balanceamento da carteira; iv) uso mais efetivo de recursos; v) probabilidade de sucesso (prazo, custo e escopo); vi) oportunidade de mercado; e vii) risco. Cada critério definido possui descrição e peso entre 1 e 10, inclusive. O usuário avalia os projetos definindo um valor para cada critério cadastrado: i) "Alto"; ii) "Médio"; e iii) "Baixo". Para modificar os critérios de avaliação, o usuário deve selecionar o botão "Modificar Critérios";

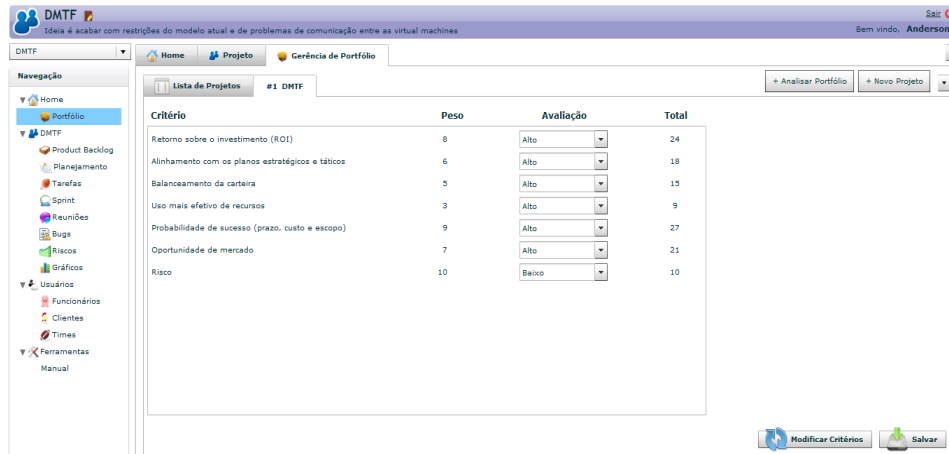


Figura 30 - Avaliação do Projeto

- Analisar Portfólio.** A análise do portfólio é feita utilizando um gráfico de bolhas gerado dinamicamente com base na escolha do usuário para os eixos x, y e raio das bolhas e para os quais são definidos critérios de análise. O usuário define quais projetos ou oportunidades de negócio são exibidos no gráfico. Após gerar o gráfico, o usuário pode analisar qual projeto ou oportunidade de negócio tem mais valor para a organização e definir, por exemplo, qual merece mais recursos ou qual projeto precisa ser cancelado. A análise do portfólio para dois projetos é apresentada na Figura 31.

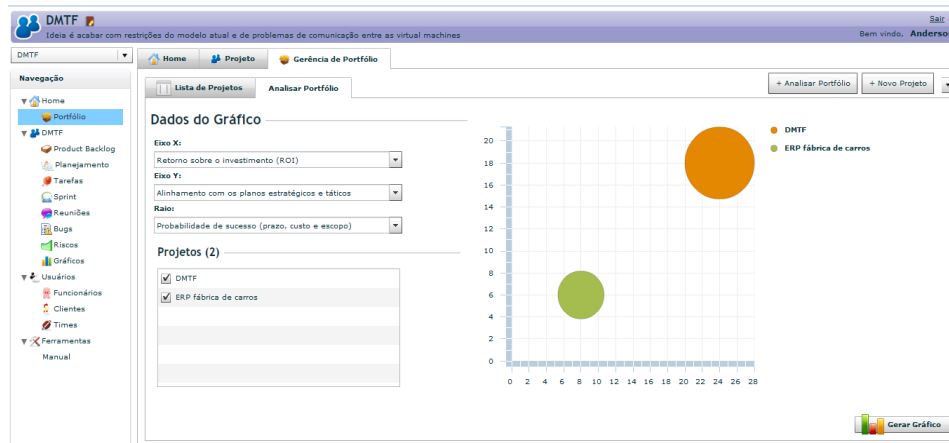


Figura 31 - Análise de Portfólio

6.3.2. Bug Tracker

No *Bug Tracker*, pode-se realizar a inserção e o acompanhamento de *bugs* do projeto (Figura 32). O ator Cliente tem permissão para relatar e acompanhar o andamento dos *bugs*, o que possibilita maior interação entre esse ator e o Time Scrum. O ator Desenvolvedor pode alterar os *bugs* associados a ele e os atores *Scrum Master* e *Product Owner* podem realizar qualquer tipo de alteração nos *bugs* existentes no sistema. Os *bugs* relatados podem ser associados a um membro do Time Scrum e ter *status* "Não Iniciado", "Andamento", "Verificar", "Finalizado" e "Rejeitado". O responsável pela inclusão pode definir uma prioridade para o *bug*, relacionar com uma história do *Product Backlog* do projeto e fazer o *upload* de um arquivo para o servidor da aplicação para disponibilizá-lo aos demais membros do Time Scrum e para o cliente do projeto.

Figura 32 - Tela Inclusão de *Bugs*

6.3.3. Gerência de Riscos

No ScrumMps 2.0, o Time Scrum pode documentar os riscos do projeto, os seus impactos, as probabilidades de ocorrência e as prioridades de tratamento

(Figura 33), porém o ator Cliente não tem permissão para cadastrar/monitorar os riscos do projeto. A probabilidade de ocorrência dos riscos varia entre "Alta", "Média", "Baixa" e "Desconhecida". A prioridade de tratamento varia entre "Crítica", "Emergência", "Urgente", "Normal" e "Baixa". Após incluir um risco, ele pode ser acompanhado pelos usuários que possuem permissão utilizando uma listagem que exhibe os dados do risco ordenados pela Prioridade de Tratamento.

The screenshot displays the 'Manutenção de Cadastro de Riscos' interface. On the left is a navigation menu with options like Home, Portfólio, DMTF, Product Backlog, Planejamento, Tarefas, Sprint, Reuniões, Bugs, Riscos (highlighted), Gráficos, Usuários, Funcionários, Clientes, Times, Ferramentas, and Manual. The main area shows a form for editing a risk record with the following fields:

- Id:** 1
- Data de Cadastro:** 23/02/2013
- Probabilidade de Ocorrência:** Alta
- Descrição do Risco:** Não conhecimento da parte tecnica.
- Prioridade de Tratamento:** Crítica
- Impacto do Risco:** Impacto do risco

At the bottom right of the form are three buttons: 'Excluir' (with a red X icon), 'Limpar Campos' (with a trash icon), and 'Salvar' (with a green checkmark icon).

Figura 33 - Manutenção de Cadastro de Riscos

6.3.4. Gráfico *Burndown*

No Gráfico *Burndown*, o Time Scrum e o cliente do projeto podem visualizar o andamento das *Sprints* por meio de um gráfico. O eixo y desse gráfico é a soma das estimativas das tarefas não finalizadas da *Sprint* e o eixo x corresponde aos dias da *Sprint*. Esse gráfico mostra a quantidade restante de trabalho por dia. Os atores têm a opção de ocultar, no eixo x do gráfico, os dias referentes a fins de semana. Essa prática é utilizada no Scrum para ocultar no gráfico os dias que não têm tarefa a ser finalizada. A cada dia da *Sprint*, o

gráfico gera um "ponto" que, ao ser selecionado, exibe uma legenda contendo a data e a soma exata das estimativas das tarefas não finalizadas (Figura 34).



Figura 34 - Gráfico *Burndown*

6.3.5. Gerência de Reuniões

Os atores *Product Owner*, *Scrum Master* e Desenvolvedor podem registrar reuniões que ocorrem durante as *Sprints* selecionando o link "Reuniões" no *menu* de navegação do ScrumMps 2.0 (Figura 35). Para realizar esse registro, o ator informa a *Sprint* e o tipo da reunião que pode ser "Planning 1", "Planning 2" ou "Retrospectiva". Em seguida, é realizado o relato da reunião utilizando o campo "Descrição" e pode-se utilizar as opções *upload/download* de arquivos. Esse relato pode ser visto por outros integrantes do projeto.



Figura 35 - Relatar Reuniões

6.3.6. Outras Funções

Além das funções descritas nas seções anteriores, outras melhorias identificadas durante o estudo da ScrumMps 1.0 foram desenvolvidas na ScrumMps 2.0, bem como a correção de erros encontrados, por exemplo:

- **Tela Inicial dos Projetos.** Na tela inicial de visualização dos projetos, que apresentava apenas uma linha do tempo do projeto, houve modificação para apresentar também dois *widjets* (Figura 36): i) visão do projeto; e ii) gráfico *burndown* da *Sprint* em execução. Essa modificação foi realizada para facilitar a visualização de informações importantes do projeto pelos *stakeholders*;
- **Manter Cadastro de Times.** Foi desenvolvida uma função para realizar a manutenção do cadastro de Times da organização. Na ScrumMps 1.0, os times tinham que ser inseridos diretamente no banco de dados, o que impossibilitava um usuário realizar a manutenção desse cadastro;
- **Manter Cadastro de Funcionários.** A ScrumMps 1.0 apresentava erro ao armazenar novo funcionário. O problema foi corrigido na ScrumMps 2.0;

- **Atualização de Tarefas ao Modificar o Projeto.** Na ScrumMps 1.0, após o usuário selecionar outro projeto, as tarefas do projeto anterior eram exibidas na listagem de tarefas do projeto selecionado. O problema foi corrigido na ScrumMps 2.0;
- **Selecionar Histórias Finalizadas para as Sprints.** Na ScrumMps 1.0, era possível incluir histórias com as tarefas finalizadas em uma *Sprint* (*Sprint Backlog*), gerando vários problemas. O problema foi corrigido na ScrumMps 2.0;
- **Validações em geral.** Foram desenvolvidas validações nas funções de manutenção de cadastros da ScrumMps 1.0 para evitar erros por dados inválidos ou nulos.

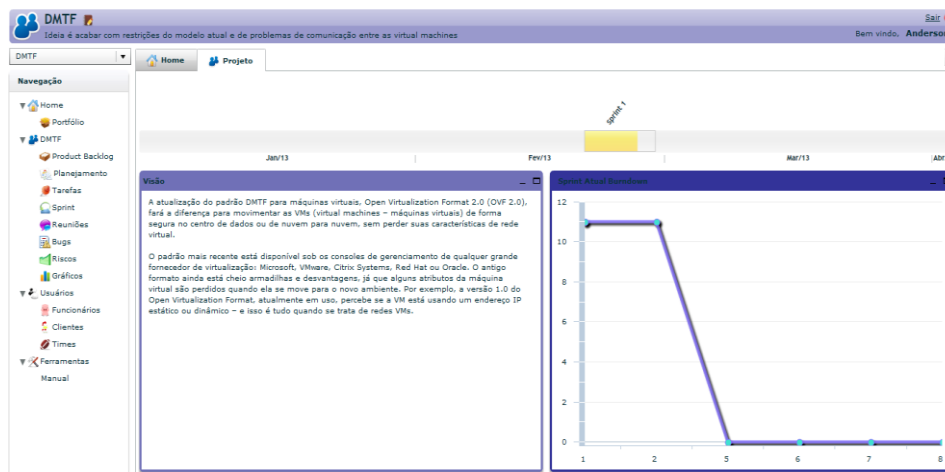


Figura 36 - Tela de Visualização do Projeto

6.4. Utilização do MPS.BR

O MPS.BR é adotado na ScrumMps como forma de gerenciar o projeto em seu ciclo de desenvolvimento, desde o levantamento de requisitos até a entrega do produto. Pode-se coletar informações utilizadas em futuras estimativas e análises para alcançar a qualidade do produto [Reis, 2011]. No desenvolvimento das funções Gerência de Portfólio e Riscos da ScrumMps 2.0,

foram apoiados os seguintes resultados esperados dos processos Gerência de Projetos (GPR) do Nível G e Gerência de Portfólio de Projetos (GPP) do Nível F do MPS.BR:

- **GPR 6.** Utilizando a função Gerência de Riscos, o usuário é capaz de documentar os riscos do projeto, os seus impactos, a probabilidade de ocorrência e a prioridade de tratamento;
- **GPR 15.** Os responsáveis pelo projeto podem utilizar a função Gerência de Riscos para monitorar os riscos documentados em relação ao planejado;
- **GPP 1.** Na função Gerência de Portfólio de Projetos, os usuários podem identificar, qualificar, priorizar e selecionar as oportunidades de negócio e as necessidades em relação aos objetivos estratégicos da organização por meio de critérios;
- **GPP 3.** Ao inserir um projeto, deve-se definir um responsável para gerenciar o projeto (*Scrum Master*);
- **GPP 4.** Os usuários podem monitorar o portfólio de projetos da organização em relação aos critérios utilizados para a priorização;
- **GPP 7.** Caso algum projeto não atenda os acordos e os requisitos que levaram à sua aprovação, o usuário tem a opção de cancelar/excluir o projeto;
- **GPP 8.** Utilizando a função de Gerência de Portfólio de Projetos, pode-se comunicar a situação do portfólio de projetos para as partes interessadas.

Outras funções desenvolvidas na ScrumMps 2.0 têm pequena relação com os processos Medição e Garantia da Qualidade do Nível F do MPS.BR. O gráfico *Burndown* pode ser utilizado para auxiliar os responsáveis do projeto a fazer medições da produtividade da equipe. O *Bug Tracker* pode ser utilizado para documentar uma não-conformidade do produto e para definir um responsável pela correção.

6.5. Quadro Comparativo entre ScrumMps 2.0 e Ferramentas Analisadas

Nesta seção, é apresentada uma análise comparativa da ScrumMps 2.0 e das quatro ferramentas analisadas no Capítulo 4: PangoScrum, IceScrum, MeuScrum e FireScrum. Nessa análise, foram avaliados os principais recursos e as características das ferramentas (Tabela 4).

Pôde-se constatar que a principal vantagem da ScrumMps 2.0 em relação às outras ferramentas é a presença de recursos baseados em processos definidos no MPS.BR, tais como, (i) Avaliação de Portfólio de Projetos, (ii) Análise de Portfólio de Projetos e (iii) Gerência de Riscos. Os recursos não contemplados no desenvolvimento da ScrumMps 2.0, por exemplo, (i) *Planning Poker*, (ii) Gráfico *Burndown* do Projeto e (iii) Registro das Reuniões Diárias do Scrum, são sugeridos como trabalhos futuros na Seção 8.4 deste trabalho.

Tabela 4 – Quadro Comparativo entre a ScrumMps 2.0 e as Ferramentas Analisadas

Características/Recursos	ScrumMps 2.0	PangoScrum	IceScrum	MeuScrum	FireScrum
Aplicação Web	X	X	X	X	X
Open Source/Free	X		X		X
SaaS (<i>Software as a Service</i>)		X		X	
Dashboard	X	X	X	X	
Gerenciamento <i>Product Backlog</i>	X	X	X		X
Planejamento e gerenciamento das Sprints	X	X	X	X	X
<i>User Stories</i>	X		X	X	X
Gráfico <i>Burndown</i> do Projeto			X	X	X
Gráfico <i>Burndown</i> das Sprints	X	X	X		X
<i>Kanban / Task Board</i>	X		X	X	X
Papeis do Scrum (<i>Product Owner, Scrum Master e Desenvolvedor</i>)	X	X	X	X	
Cliente pode interagir com as atividades na ferramenta	X		X	X	
Associar tarefas a um usuário	X			X	X

Tabela 4 – Quadro Comparativo entre a ScrumMps 2.0 e as Ferramentas Analisadas (cont.)

Características/Recursos	ScrumMps 2.0	PangoScrum	IceScrum	MeuScrum	FireScrum
<i>Planning Poker</i>			X		X
Registro das Reuniões Diárias do Scrum					X
Registro das Reuniões de Retrospectiva	X		X		
Definir Visão do Projeto	X		X		
Módulo de <i>Bug Tracker</i>	X				X
Avaliação do Portfólio de Projetos	X				
Análise do Portfólio de Projetos	X				
Gerenciamento de Riscos	X				

6.6. Considerações Finais

A construção das funções relatadas neste capítulo foram baseadas no Scrum e no MPS.BR. Durante a fase de análise e de implementação da ScrumMps 2.0, foi evidenciado a dificuldade em desenvolver uma ferramenta que integre duas abordagens distintas que visam a melhoria dos processos de gerência e de desenvolvimento de software. Assim, é interessante que a ScrumMps 2.0 continue a ser evoluída/melhorada após a conclusão deste trabalho por ser uma ferramenta de código livre. O código da ferramenta foi disponibilizado em: <https://scrummps.googlecode.com/svn/branches/versao2>. Nesse *link*, também foram disponibilizados os *scripts* SQL para criação do banco de dados da ScrumMps 2.0.

Neste capítulo, foram apresentadas as funções desenvolvidas na ScrumMps 2.0 e as suas principais telas. Além disso, questões técnicas envolvendo o desenvolvimento dessas funções e como o MPS.BR foi utilizado na conceituação de algumas dessas funções foram abordadas neste capítulo.

7. AVALIAÇÃO

7.1. Considerações Iniciais

Neste capítulo, é discutida a avaliação da ScrumMps 2.0 realizada por meio de um questionário aplicado a profissionais da área de Engenharia de Software e discentes dos cursos de Ciência da Computação e Sistemas de Informação de uma Instituição Federal de Ensino Superior. Para essa avaliação, os participantes (respondentes) realizaram testes, seguindo um roteiro, na ScrumMps 2.0 disponibilizada na Internet no *link*: <http://www.scrummeps.com> e, posteriormente, responderam um questionário. O questionário aplicado aos respondentes teve os seguintes objetivos:

- Verificar o apoio da ScrumMps 2.0 aos processos do MPS.BR;
- Verificar o apoio da ScrumMps 2.0 às práticas do Scrum;
- Avaliar a usabilidade da ScrumMps 2.0;
- Avaliar a contribuição da ScrumMps 2.0 na aplicação conjunta dos processos do MPS.BR com o Scrum;
- Verificar a aceitação da ScrumMps 2.0 para apoiar empresas que visam implantar o modelo MPS.BR em conjunto com o Scrum.

Questões referentes a caracterização dos respondentes da avaliação são apresentadas na Seção 7.2. A análise da avaliação referente ao apoio da ScrumMps 2.0 aos processos do MPS.BR e ao Scrum é relatada nas Seções 7.3 e 7.4, respectivamente. A análise dos resultados da avaliação da usabilidade da ScrumMps 2.0 é mostrada na Seção 7.5. A análise dos resultados da avaliação referente à contribuição da ScrumMps 2.0 na aplicação conjunta do Scrum e MPS.BR e à aceitação da ScrumMps 2.0 são abordadas na Seção 7.6.

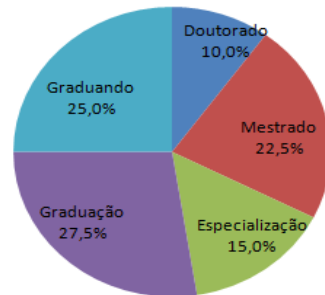
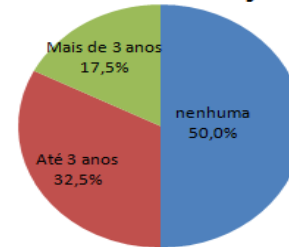
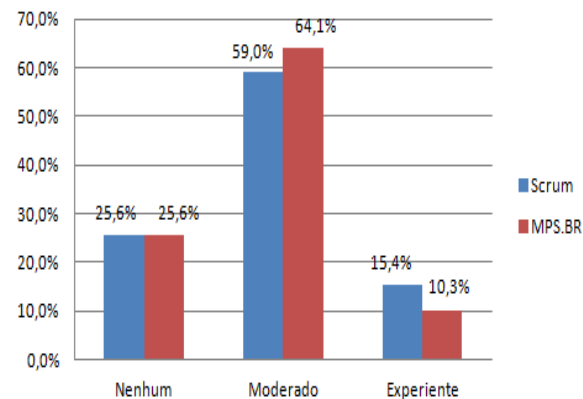
7.2. Caracterização dos Respondentes

Foram convidados 78 participantes para realizar a avaliação da ScrumMps 2.0, sendo que 40 (51,3%) responderam o questionário. Para caracterizar o perfil dos respondentes, foram elaboradas as seguintes perguntas: i) Qual é o seu grau de escolaridade?; ii) Qual é o seu nível de conhecimento em Scrum?; iii) Qual é o seu nível de conhecimento em MPS.BR?; iv) Qual sua experiência com gerência de projetos?; e iv) Outras informações do perfil. Na caracterização, os 40 respondentes forneceram respostas às perguntas apresentadas.

Analisando as respostas, pode-se perceber que 10 respondentes (25%) estão cursando a graduação, 11 respondentes (27,5%) são graduados, 6 respondentes (15%) possuem especialização, 9 respondentes (22,5%) possuem mestrado e 4 respondentes (10%) possuem doutorado (Figura 37). A respeito do conhecimento na área de gerência de projetos (Figura 38), 20 respondentes (50%) possuem experiência na área, sendo 13 (32,5%) com experiência de até 3 anos e 7 (17,5%) com experiência acima de 3 anos, e 20 respondentes (50%) não possuem experiência. Sobre o conhecimento dos temas abordados na ScrumMps 2.0 (Figura 39), a maioria dos respondentes possui nível "Experiente" a "Moderado" de conhecimento no Scrum e no MPS.BR, em ambos 29 respondentes (74,4%), sendo 23 (59,0%) Moderado e 6 (15,4%) Experiente no Scrum e 25 (64,1%) Moderado e 4 (10,3%) Experiente no MPS.BR. Além disso, 10 respondentes (25,6%) não possuem conhecimento em Scrum nem MPS.BR.

Uma das perguntas sobre o perfil dos respondentes foi disponibilizada de forma dissertativa. As respostas que merecem destaques são:

- "Implementador MPS.BR, instrutor do curso de introdução ao MPS.BR, avaliador líder MPS.BR, certificado PMP com mais de 9 anos de experiência em Gerência de Projetos";

Grau de Escolaridade**Figura 37 - Grau de Escolaridade dos Respondentes****Experiência em Gerência de Projetos****Figura 38 - Experiência em Gerência de Projetos dos Respondentes****Conhecimento em Scrum e MPS.BR****Figura 39 - Conhecimento dos Respondentes sobre os Temas Abordados (Scrum e MPS.BR)**

- "Implementador oficial MPS.BR com mais de 14 anos de experiência prática com desenvolvimento de sistemas e projetos de software";
- "Possuo certificação Scrum Master";
- "Trabalhei como desenvolvedor e líder de projetos, sendo que 2 deles empregando a metodologia Scrum";
- "Docente ensino superior. Experiência em gerência de projetos apenas na parte de ensino";

- "Tenho conhecimento em outros frameworks ágeis, no modelo CMMI-DEV e PMBOK".

7.3. Apoio aos Processos do MPS.BR

Os recursos desenvolvidos na ScrumMps 2.0 apoiam os processos Gerência de Projetos (GPR), Gerência de Requisitos (GRE) e Gerência de Portfólio de Projetos (GPP) do MPS.BR. Para verificar esse apoio, foram elaboradas 4 afirmações. Para cada afirmação, os respondentes escolheram uma das seguintes alternativas: i) "Discordo Plenamente"; ii) "Discordo"; iii) "Concordo"; e iv) "Concordo Plenamente". O resultado é apresentado na Figura 40.

Analisando os resultados, os quesitos de (i) acompanhar o portfólio, (ii) identificar e priorizar os projetos, (iii) gerenciar os requisitos e (iv) gerenciar os riscos até o Nível G estão presentes na ScrumMps 2.0 (respondentes que concordam ou concordam plenamente) para 32 respondentes (76,5%), 31 respondentes (84,2%), 34 respondentes (87,1%) e 36 respondentes (92,3%), respectivamente. Por outro lado, 5 respondentes (13,5%), 6 respondentes (15,8%), 5 respondentes (12,9%) e 3 respondentes (7,7%) discordam ou discordam plenamente dos quesitos. Cabe ressaltar que há diferença em alguns percentuais, pois há variação na quantidade de respondentes para cada quesito, sendo 37 respondentes, 38 respondentes, 39 respondentes e 39 respondentes, respectivamente.

Quanto à interpretação dos respondentes que discordam do apoio da ScrumMps 2.0 ao gerenciamento adequado dos riscos (5 respondentes - 13,5%), pode ter ocorrido certo viés em sua resposta por considerarem a definição de gerenciamento de riscos no Nível C do MPS.BR, no qual são abordadas ações de gerenciamento qualitativo dos riscos do projeto, por exemplo, a mitigação dos

riscos. Entretanto, é definido no Nível G do MPS.BR que os riscos precisam ser gerenciados quantitativamente, ou seja, realizando a identificação e a documentação dos riscos. A ScrumMps 2.0 permite a documentação dos riscos armazenando seu impacto, sua probabilidade de ocorrência e sua prioridade de tratamento como definido no Nível G.

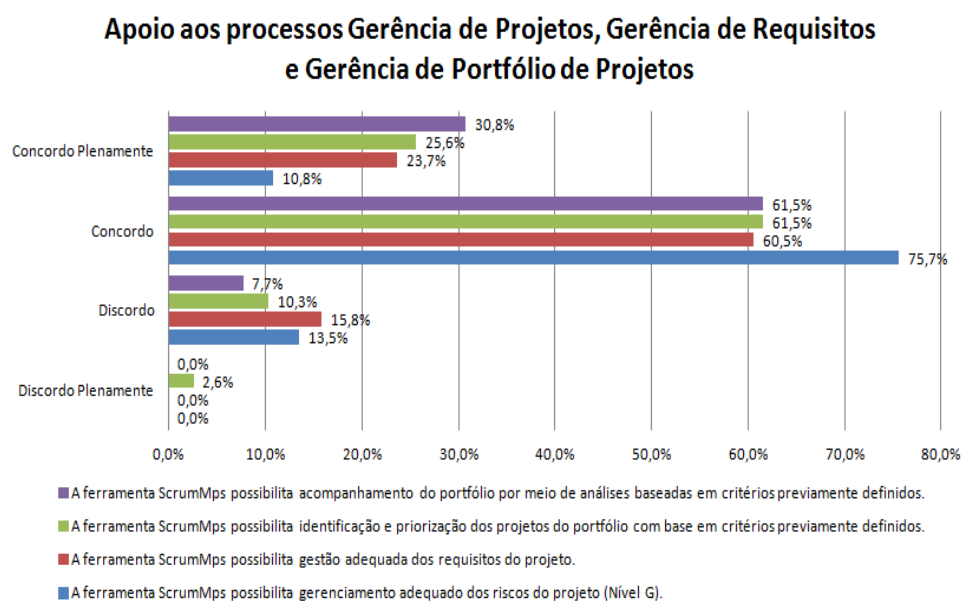


Figura 40 - Avaliação do Apoio da ScrumMps 2.0 aos Processos do MPS.BR

7.4. Apoio às Práticas do Scrum

Para verificar o apoio da ScrumMps 2.0 às práticas do Scrum, foram elaboradas 6 afirmações, para as quais os respondentes deveriam escolher uma das seguintes alternativas: i) "Discordo Plenamente"; ii) "Discordo"; iii) "Concordo"; e iv) "Concordo Plenamente". As afirmações sintetizam os recursos da ScrumMps 2.0 ao apoio das práticas do Scrum. O resultado é apresentado na Figura 41.

Analisando os resultados, pode-se concluir que a ScrumMps 2.0 atende adequadamente as práticas do Scrum, pois 38 respondentes (97,4%), 39

respondentes (100%), 29 respondentes (78,4%), 38 respondentes (97,5%), 33 respondentes (89,9%) e 37 respondentes (97,3%) concordam ou concordam plenamente com as afirmações, respectivamente. Os resultados indicam que 5 respondentes (13,2%) não consideram o gráfico *Burndown* adequado. Essa quantidade pode ser explicada pelo fato desse gráfico não ter sido gerado adequadamente durante a realização do teste, seguindo o roteiro, na ScrumMps 2.0. Os resultados ainda indicam que 8 respondentes (21,6%) discordam que a ScrumMps 2.0 possibilita a gestão de impedimentos e o papel do *Scrum Master*. Essa quantidade pode ser explicada pelo fato da ScrumMps 2.0 não conter uma função para documentação e para acompanhamento dos impedimentos do projeto. Cabe ressaltar que há diferença em alguns percentuais, pois há variação na quantidade de respondentes para cada afirmação, sendo 39 respondentes, 39 respondentes, 37 respondentes, 39 respondentes, 38 respondentes e 38 respondentes, respectivamente.

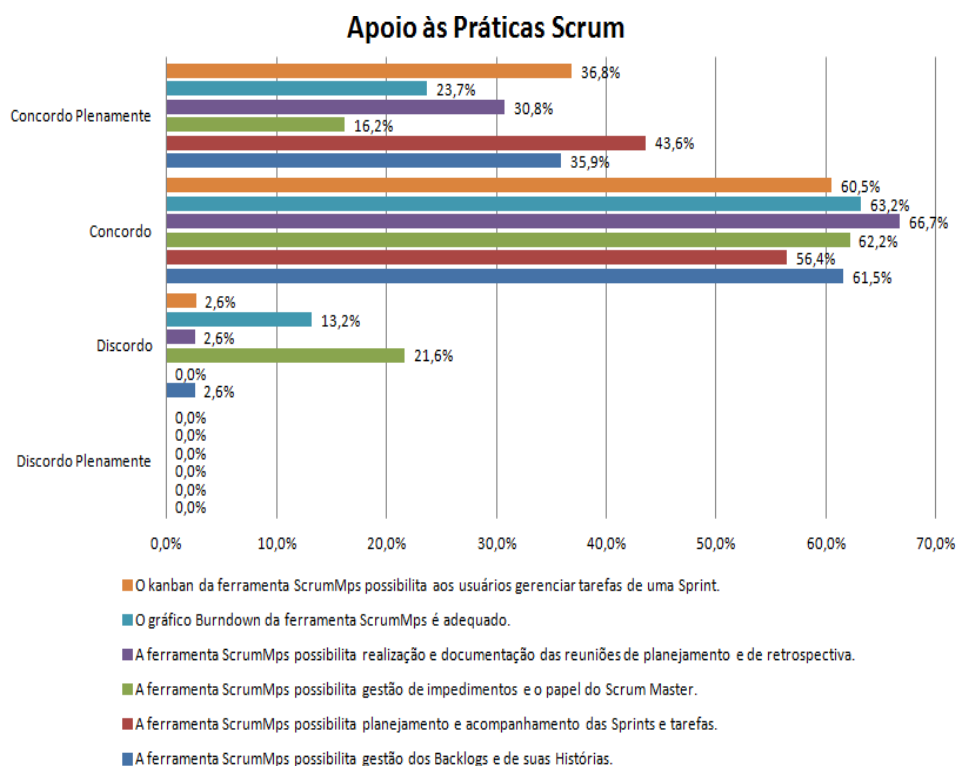


Figura 41 - Avaliação do Apoio da ScrumMps 2.0 às Práticas Scrum

7.5. Usabilidade

Para avaliar a usabilidade da ScrumMps 2.0, os respondentes deram nota de 0 (mínima) a 5 (máxima) para as seguintes afirmações: i) A ferramenta ScrumMps é intuitiva e de fácil aprendizado; ii) A terminologia usada na ferramenta ScrumMps é adequada aos usuários-alvo; iii) A ferramenta ScrumMps adota um padrão consistente de interface (simetria); e iv) A navegabilidade e o acesso às funções da ferramenta ScrumMps é adequada. O resultado é apresentado na Figura 42.

Analisando a avaliação, pode-se perceber que apenas a afirmação que a ScrumMps é intuitiva e de fácil aprendizado foi avaliada com nota 3 pela maioria dos respondentes (16 respondentes - 41%). Porém, as afirmações foram

avaliadas pela maioria dos respondentes com nota 4 ou 5, 20 respondentes (49,3%), 33 respondentes (86,8%), 28 respondentes (71,8%) e 22 respondentes (46,4%), respectivamente. Cabe ressaltar que há diferença em alguns percentuais, pois há variação na quantidade de respondentes para cada afirmação, sendo 39 respondentes, 38 respondentes, 39 respondentes e 39 respondentes, respectivamente.

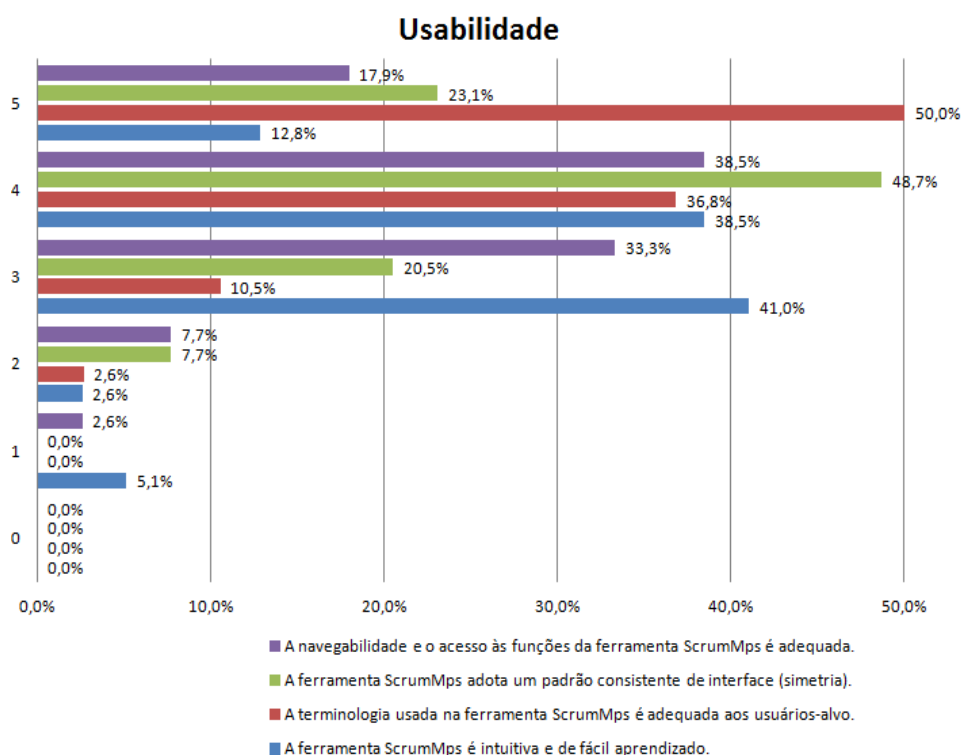


Figura 42 - Avaliação da Usabilidade da ScrumMps 2.0

7.6. Objetivo

Para avaliar o objetivo da ScrumMps 2.0, foram elaboradas uma afirmação e uma pergunta. A afirmação foi "A ferramenta ScrumMps contribui significativamente com a aplicação conjunta dos processos do MPS.BR com o Scrum", para qual os respondentes escolheram uma das seguintes alternativas: i)

"Discordo Plenamente"; ii) "Discordo"; iii) "Concordo"; e iv) "Concordo Plenamente". A pergunta foi "Você utilizaria ou indicaria a ferramenta ScrumMps para uma empresa que visa implantar o modelo MPS.BR em conjunto com o Scrum?", para qual os respondentes escolheram a alternativa "Sim" ou a alternativa "Não". Os resultados são apresentados na Figura 43 e na Figura 44, respectivamente.

Analisando a avaliação, pode-se perceber que a ScrumMps 2.0 atende ao objetivo proposto: contribui com a aplicação conjunta das práticas do Scrum com os processos definidos no MPS.BR, pois 36 respondentes (94,8%), sendo que 31 respondentes (81,6%) concordam e 5 respondentes (13,2%) concordam plenamente. Cabe ressaltar que há diferença em alguns percentuais, pois há variação na quantidade de respondentes para afirmação e para a pergunta, sendo 38 respondentes para ambas, respectivamente.

Contribuição da ScrumMps 2.0 na Aplicação Conjunta do Scrum e MPS.BR

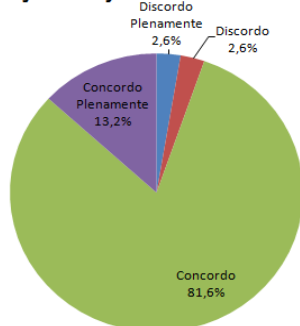


Figura 43 - Avaliação da Contribuição da ScrumMps na Aplicação conjunta do Scrum e MPS.BR

Utilização da ScrumMps 2.0 em uma Organização Real

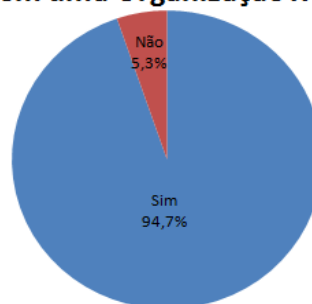


Figura 44 - Avaliação da Utilização da ScrumMps em uma Organização Real

7.7. Considerações Finais

Neste capítulo, foram apresentados os resultados obtidos da compilação das respostas do questionário aplicado para avaliar a ScrumMps 2.0. Foram

avaliadas algumas questões, tais como, o seu apoio às práticas do Scrum e dos processos do MPS.BR, a sua usabilidade e a sua contribuição.

Além dessas questões, foi disponibilizada uma questão dissertativa aos respondentes para fazerem críticas/sugestões. Dos 40 entrevistados, 17 (42,5%) responderam essa questão. Fazendo uma análise das respostas, podem ser destacados os seguintes pontos:

- Podem ser realizadas melhorias na usabilidade da ScrumMps 2.0. Por exemplo: i) melhorar a escolha de datas ao inserir uma tarefa ou um projeto; ii) melhorar a tela (interface) de reuniões; iii) criar um *Dashboard* das principais informações dos projetos na tela inicial da ScrumMps 2.0; e iv) melhorar a tela (interface) de planejamento das *Sprints*;
- Migrar a camada de interface da ScrumMps 2.0 para HTML⁷ 5 (*HyperText Markup Language*) e CSS⁸ 3 (*Cascading Style Sheets*) para aumentar sua aceitação no mercado;
- Alguns Resultados Esperados dos processos GPR, GRE e GPP não foram contemplados na ScrumMps 2.0;
- Como melhorias futuras da ScrumMps 2.0, podem ser implementados novos recursos relacionados a Gerência de Requisitos (com controle de mudanças), Gerência de Riscos (com monitoramento e controle dos riscos), Garantia de Qualidade (cadastro e monitoramento de ações corretivas) e análise de viabilidade dos projetos;
- Não existe ajuda do sistema. Seria interessante a ScrumMps 2.0 conter um manual de ajuda para auxiliar os usuários que não conhecem Scrum e/ou MPS.BR.

⁷ <http://www.w3.org/html>

⁸ <http://www.w3.org/Style/CSS>

8. CONSIDERAÇÕES FINAIS

A ScrumMps 2.0, desenvolvida neste trabalho, é uma ferramenta *Web* para gerenciamento de projetos baseada no Scrum e no MPS.BR. A metodologia ágil Scrum vem sendo acolhida pelas empresas de software com o intuito de auxiliar no gerenciamento do projeto baseando-se em pequenas iterações chamadas de *Sprints*. Por sua vez, o MPS.BR é um modelo criado com o objetivo de melhorar o processo de desenvolvimento de software.

Neste capítulo, são apresentadas algumas observações resultantes deste trabalho. Breve conclusão do trabalho é apresentada na Seção 8.1. Algumas contribuições obtidas com a realização deste trabalho são citadas na Seção 8.2. Limitações são explicadas na Seção 8.3. Alguns trabalhos futuros a serem desenvolvidos como desdobramento deste são sugeridos na Seção 8.4.

8.1. Conclusões

Neste trabalho, foi desenvolvida a ScrumMps 2.0. Para isso, foi realizada a reengenharia da ScrumMps 1.0 e um estudo do Scrum, do modelo MPS.BR e de ferramentas para gerência de projetos que apoiam o Scrum para identificar novos requisitos a serem desenvolvidos neste trabalho. Após o levantamento desses requisitos, foi realizada a engenharia reversa da ScrumMps 1.0 para corrigir erros encontrados e desenvolver novas funções. São apresentados na Tabela 5 as principais funções desenvolvidas na ScrumMps 1.0 e na ScrumMps 2.0. A ScrumMps 1.0 apoiou processos Gerência de Projetos e Gerência de Requisitos do Nível G do MPS.BR e a ScrumMps 2.0 apoiou processos Gerência de Projetos e Gerência de Portfólio de Projetos dos Níveis G e F do MPS.BR.

Na ScrumMps 2.0, o objetivo foi auxiliar as organizações que queiram utilizar a metodologia ágil Scrum e implantar o modelo MPS.BR. Porém, por

causa da dificuldade em desenvolver recursos que abordem as técnicas e os conceitos do Scrum e do MPS.BR, existem pontos que podem ser melhorados e adicionados na ferramenta.

Tabela 5 – Principais Recursos Desenvolvidos na ScrumMps

ScrumMps 1.0	ScrumMps 2.0
Cadastro e Manutenção de Usuários	Avaliação do Portfólio de Projetos
Gerenciamento do <i>Product Backlog</i>	Análise do Portfólio de Projetos
Planejamento das <i>Sprints</i>	Gráfico <i>Burndown</i>
Gerenciamento de Tarefas (Kanban)	Registro das Reuniões das <i>Sprints</i>
Cadastro e Manutenção dos Projetos	Gerenciamento de Riscos
	<i>Bug Tracker</i>

Analisando a avaliação da ScrumMps 2.0, pode-se perceber que a ferramenta apoiou adequadamente os seguintes processos dos Níveis G e F do MPS.BR: i) Gerência de Projetos; ii) Gerência de Requisitos; e iii) Gerência de Portfólio de Projetos. Além disso, há apoio às principais práticas do Scrum, tais como, (i) Gerenciamento do *Product Backlog*, (ii) Planejamento e Gerenciamento das *Sprints*, (iii) Gráfico *Burndown* e (iv) Registro de Reuniões. Na avaliação, foi identificado que a ScrumMps 2.0 possui usabilidade adequada e que atingiu o objetivo de contribuir com a aplicação conjunta dos processos do MPS.BR e do Scrum. Por outro lado, percebeu-se que há melhorias a serem feitas na usabilidade e funções a serem desenvolvidas para abordar melhor os processos do MPS.BR e as práticas de Scrum. Algumas dessas melhorias são apresentadas na seção 8.3.

8.2. Contribuições

Conforme apresentado neste trabalho, há crescente iniciativa de empresas para implantar o MPS.BR em conjunto com práticas ágeis de Scrum. Em contrapartida, observa-se falta de ferramentas que apoiem a utilização do Scrum com um modelo de melhoria de processos como o MPS.BR. Dessa forma, este

trabalho deixa como contribuição a evolução de uma ferramenta livre e de código aberto para gerência de projetos baseada no Scrum e no MPS.BR, bem como a viabilidade de desenvolvimento de uma ferramenta que combine Scrum e MPS.BR.

A ScrumMps 2.0 foi desenvolvida para *Web* utilizando as linguagens de programação Adobe Flex e Java. Os recursos Gráfico *Burndown* e Reuniões, disponíveis na ScrumMps 2.0, contribuíram para melhor abordagem da metodologia ágil Scrum. Os recursos Gerência de Portfólio de Projetos e Gerência de Riscos, também disponíveis, contribuíram para melhor abordagem de processos do Nível G e do Nível F do MPS.BR.

8.3. Limitações da ScrumMps 2.0

Esta seção é destinada a apresentar algumas limitações da ScrumMps 2.0 identificados durante o desenvolvimento deste trabalho:

- A ScrumMps 2.0 não possui um *site* na *internet* (hospedagem fixa). Este *site* seria útil para: i) divulgar a ferramenta no mercado; e ii) atrair colaboradores para continuar o desenvolvimento;
- O *framework* Flex foi utilizado para o desenvolvimento da camada de interface da ScrumMps 2.0, porém ele foi descontinuado pelo fabricante. Assim, a sua aceitação no mercado tende a ser reduzida;
- A ScrumMps 2.0 não guarda informações das ações realizadas (*log*) pelos usuários na manutenção de Projetos, Tarefas, Histórias e *Bugs*;
- Não é possível salvar as análises do portfólio, ou seja, o usuário não consegue fazer comparações entre as análises feitas;
- A ScrumMps 2.0 não conta com recursos para facilitar o processo de desenvolvimento de equipes distribuídas, por exemplo, fórum de notícias, módulo para realizar a *Daily Scrum* e *Planning Poker*;

- As telas (interfaces) de Planejamento das *Sprints* e Reuniões não são intuitivas;
- A ScrumMps 2.0 não apresenta o gráfico *Burndown* do projeto.

8.4. Trabalhos Futuros

A seguir, são apresentados alguns tópicos que podem ser considerados para dar continuidade deste trabalho:

- Implementar outras funções referentes aos processos do MPS.BR, como: i) Garantia da Qualidade; e ii) Gerência de Recursos Humanos;
- Melhorar a função de Gerência de Riscos da ScrumMps 2.0 para atender o Nível C do MPS.BR;
- Modificar a camada de interface da ScrumMps 2.0 para as tecnologias HTML e CSS por serem, atualmente, tecnologias mais aceitas no mercado;
- Armazenar no banco de dados da aplicação o histórico das ações dos usuários e exibi-las nas telas (interfaces) de manutenção de Projetos, Riscos, Histórias, Tarefas e *Bug*;
- Implementar funções que atendam outras práticas de Scrum, como: i) *Daily Scrum*; ii) *Planning Poker*; e iii) Gráfico *Burndown* do projeto;
- Criar opção para o usuário "salvar" as análises do portfólio;
- Implementar um fórum de notícias/discussão (*wiki*) para melhorar a comunicação entre os *stakeholders* do projeto;
- Realizar um mapeamento de quais práticas do Scrum que apoiam certos atributos de processos ou resultados esperados do MPS.BR. Posteriormente, identificar na ScrumMps 2.0 quais recursos apoiam o Scrum e o MPS.BR em conjunto, quais recursos apoiam somente o Scrum e quais recursos apoiam somente o MPS.BR;
- Desenvolver um módulo para dispositivos móveis da ScrumMps 2.0 para os sistemas IOS e Android;

- Realizar um estudo de caso do uso da ScrumMps 2.0 em uma organização que utiliza a metodologia ágil Scrum e vise implantar o modelo MPS.BR.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGILE MANIFESTO. **Manifesto for Agile Software Development**. 2001. Disponível em: <<http://agilemanifesto.org/>>. Acessado em: 17 de outubro de 2012.
- Agrawal, M.; Chari, K. Software Effort, Quality, and Cycle Time: **A Study of CMM Level 5 Projects**. In: IEEE Transactions on Software Engineering. v. 33. n. 3. pp. 145-156. 2007.
- Ambler, S. W. **Agile Teams Mini-Survey Results**. In: Dr. Dobb's Journal. 2008. Disponível em: <<http://www.ambyssoft.com/surveys/agileTeams2011.html>>. Acessado em: 8 de dezembro de 2012.
- Balle, A. R. **Análise de Metodologias Ágeis: Conceitos, Aplicações e Relatos sobre XP e Scrum**. Monografia de Graduação. Universidade Federal do Rio Grande do Sul. 79p. 2011.
- Bassi Filho, D. L. **Experiências com Desenvolvimento Ágil**. Dissertação de Mestrado. Instituto de Matemática e Estatística da Universidade de São Paulo. 150p. 2008.
- Beck, K. **Programação Extrema (XP) Explicada - Acolha as Mudanças**. Bookman. 182p. 2004.
- Belchior, A. D. **Um Modelo Fuzzy para Avaliação da Qualidade de Software**. Tese de Doutorado. Universidade Federal do Rio de Janeiro - COPPE. 185p. 1997.
- Boehm, B. **A View of 20th and 21st Century Software Engineering**. In: 28th International Conference on Software Engineering. pp. 12-29. 2006.
- Catunda, E.; Nascimento, C.; Cerdeiral, C.; Santos, G.; Nunes E.; Shots, N.; Shots, M.; Rocha, A. R. **Implementação do Nível F do MR-MPS com Práticas Ágeis do Scrum em uma Fábrica de Software**. In: X Simpósio Brasileiro de Qualidade de Software. 2011. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2011/SBQS2011-RE10_82940.pdf>. Acessado em: 20 de abril de 2013.
- Cohn, M. **Agile Estimating and Planning**. Prentice Hall. 368p. 2005.

- Cohn, M. **Introduction to Scrum - An Agile Process**. 2002. Disponível em: <<http://mountaingoatsoftware.com/topics/scrum>>. Acessado em: março de 2013.
- Cohn, M. **User Stories Applied: For Agile Software Development**. Addison-Wesley. 259p. 2004.
- Colenci Neto, A. **Proposta de um Modelo de Referência para Desenvolvimento de Software com Foco na Certificação do Mps.Br**. Tese de Doutorado. Instituto de Matemática e Estatística da Universidade de São Paulo. 179p. 2008.
- Duarte, K. C.; Falbo, R. A. **Uma Ontologia de Qualidade de Software**. In: Workshop de Qualidade de Software. pp. 275-285. 2000.
- Felipe, P. F. **CMMI e Spice: Um Estudo Comparativo na Abordagem da Engenharia de Requisitos**. Monografia de Curso de Pós-Graduação Lato Sensu. Universidade São Judas Tadeu. 94p. 2006.
- Fowler, M. **The New Methodology**. 2005. Disponível em: <<http://www.martinfowler.com/articles/newMethodology.html#N8B>>. Acessado em: março de 2013.
- Garvin, D. **Competing on the Eight Dimensions of Quality**. Harvard Business Review. v.65. n.6. pp.101-109. 1987.
- Godfrey, A. B.; Kenett, R. S. **A Perspective on Past Contributions and Future Impact. In: Quality and Reliability Engineering International**. v. 23. n. 6. pp. 653-663. 2007.
- Gomes, P. **A Evolução do Conceito de Qualidade**. In: Caderno de Biblioteconomia Arquivística e Documentação. pp. 6-18. 2004.
- ISO/IEC 12207. **Systems and Software Engineering - Software Life Cycle Processes**. 2008.
- ISO/IEC 15504. **Information Technology - Process Assessment. The International Organization for the Standardization and the International Electrotechnical Commission**. 2003.
- ISO/IEC 9126. **Information Technology - Software Product Evaluation - Quality Characteristics and the Guidelines for Their Use**. 2001.

- Jung, C. F. **Metodologia Aplicada a Projetos de Pesquisa: Sistemas de Informação & Ciência da Computação**. Taquara, 2009. Disponível em: <<http://www.jung.pro.br/moodle/mod/resource/view.php?id=102>>
- Kalermo, J.; Rissanen, J. **Agile Software Development in Theory and Practice**. Dissertação de Mestrado. University of Jyväskylä. Finland. 188p. 2002.
- Kniberg, H. **Scrum and XP from the Trenches**. 2007. Disponível em: <<http://tschering.mono.net/upl/10004/110224ScrumAndXpFromTheTrenches.pdf>>. Acessado em: Março de 2013.
- Kniberg, H.; Skarin, M. **Kanban e Scrum - Obtendo o Melhor de Ambos**. C4Media Inc. 139p. 2009.
- Mahnic, V. **A Case Study on Agile Estimating and Planning using Scrum**. In: Electronics and Electrical Engineering. v. 111. pp. 123-128. 2011.
- Mello, M. S. **Melhoria de Processos de Software Multi-Modelos Baseada nos Modelos MPS e CMMI-DEV**. Tese de Doutorado. Universidade Federal do Rio de Janeiro. 232p. 2011.
- Nonaka, I.; Takeuchi, H. **The New New Product Development Game**. In: Harvard Business Review. v. 64. n. 1. pp. 137-149, 1986.
- Oliveira, A. C. G.; Guimarães, F. A.; Fonseca, I. A. 2007. **Utilizando Metodologias Ágeis para Atingir Mps.Br Nível F na Powerlogic**. Disponível em: <http://www.softex.br/portal/softexweb/uploadDocuments/_mpsbr/T1-PowerLogic-WE.pdf>. Acessado em: Março de 2013.
- Oliveira, E.; Lima, R. **Estado da Arte Sobre o Uso do Scrum em Ambientes de Desenvolvimento Distribuído de Software**. In: Revista de Sistemas e Computação. v. 1. n. 2. pp. 106-119. 2011.
- Pereira, P.; Torreão, P.; Marçal, A. S. **Entendendo Scrum para Gerenciar Projetos de Forma Ágil**. In: Mundo PM. v. 1. pp. 3-11. 2007.
- Pommer, J. E. F. **Aplicabilidade da Qualidade de Software: Estudo de Caso com Nível G do Mps-Br como uma Alternativa para Micro e Pequenas Empresas**. Monografia de Graduação. Universidade Regional do Noroeste do Estado do Rio Grande do Sul. 57p. 2011.

- Reis, T. C. **Ferramenta Interativa para Suporte ao Scrum e Mps.Br.** Monografia de Curso de Especialização em Informática. Departamento de Ciências Exatas. Universidade Federal de Minas Gerais. 47p. 2011.
- Rincon, A. M. **Qualidade de Software.** In: XI Encontro de Estudantes de Informática do Tocantins. pp. 75-86. 2009.
- Salviano, C. F. **Melhoria e Avaliação de Processo de Software com o Modelo ISO/IEC 15504-5.** UFLA/FAEPE. 56p. 2006.
- Schwaber, K. **Agile Project Management With Scrum.** Microsoft Press. 151p. 2004.
- Schwaber, K.; Sutherland, J. **Scrum Guide.** 2011. Disponível em: <<http://www.scrum.org/>>. Acessado em: 17 de outubro de 2012.
- Silva, F. G.; Hoentsch, S. C.P.; Silva, L. **Uma Análise das Metodologias Ágeis FDD e Scrum sob a Perspectiva do Modelo de Qualidade MPS.BR.** In: Scientia Plena. v. 5. n. 12. 2011.
- SEI. **CMMI for Development (CMMI-DEV).** v1.3. Carnegie Mellon University. 2010a.
- SEI. **CMMI for Services (CMMI-SVC).** v1.3. Carnegie Mellon University. 2010b.
- SOFTEX. **Mps.Br - Melhoria de Processo do Software Brasileiro - Guia Geral MPS de Software.** 2012. Disponível em: <<http://www.softex.br/mpsbr/>>. Acessado em: Março de 2013.
- SOFTEX. **Mps.Br - Melhoria de Processo do Software Brasileiro - Guia de Implementação - Parte 1: Fundamentação para Implementação do Nível G do MR-MPS.** 2011a. Disponível em: <<http://www.softex.br/mpsbr/>>. Acessado em: Março de 2013.
- SOFTEX. **Mps.Br - Melhoria de Processo do Software Brasileiro - Guia de Implementação - Parte 2: Fundamentação para Implementação do Nível F do MR-MPS.** 2011b. Disponível em: <<http://www.softex.br/mpsbr/>>. Acessado em: Março de 2013.
- Souza Neto, O. N. de. **Análise Comparativa das Metodologias de Desenvolvimento de Softwares Tradicionais e Ágeis.** Monografia de Graduação. Universidade da Amazônia. 62p. 2004.

- Sugimori, Y.; Kusunoki, K.; Cho, F.; Uchikawa, S. **Toyota Production System and Kanban System Materialization of Just-In-Time and Respect-For-Human System**. In: International Journal of Production Research. v. 15, n. 6, pp. 553-564. 1977.
- Szimanski, F.; Albuquerque, J.; Furtado, F. **Implementando Maturidade e Agilidade em uma Fábrica de Software Através de Scrum e Mps.Br Nível G**. In: XI Encontro de Estudantes de Informática do Tocantins. Centro Universitário Luterano de Palmas. pp. 161-170. 2009.
- Travassos, G. H.; Kalinowski, M. **iMPS 2011 - Resultados de Desempenho das Empresas que Adotaram o Modelo MPS de 2008 a 2011**. In: Campinas, SP : SOFTEX. 36p. 2012.
- VersionOne. **State of Agile Survey**. 2010. Disponível em: <http://www.versionone.com/pdf/2010_State_of_Agile_Development_Survey_Results.pdf>. Acessado em: 8 de dezembro de 2012.