

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR
CURSO SUPERIOR DE TECNOLOGIA EM DESENVOLVIMENTO DE SISTEMAS

FABIO EDUARDO GAZONI

**ANÁLISE DA COMPATIBILIDADE ENTRE O MODELO MPS.BR
NÍVEL G E A METODOLOGIA DE DESENVOLVIMENTO SCRUM**

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA
2011

FABIO EDUARDO GAZONI

**ANÁLISE DA COMPATIBILIDADE ENTRE O MODELO MPS.BR
NÍVEL G E A METODOLOGIA DE DESENVOLVIMENTO SCRUM**

Trabalho de Diplomação do Curso Superior de Tecnologia em Desenvolvimento de Sistemas – CSTDS – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Me. Alan Gavioli.

MEDIANEIRA
2011



TERMO DE APROVAÇÃO

Análise da compatibilidade entre o modelo MPS.BR nível G e a metodologia de desenvolvimento Scrum

Por
Fabio Eduardo Gazoni

Este Trabalho de Diplomação (TD) foi apresentado às 8:30 h do dia 01 de novembro de 2011 como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Manutenção Industrial, da Universidade Tecnológica Federal do Paraná, *Campus* Medianeira. Os acadêmicos foram argüidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado com louvor e mérito.

Prof. Me. *Alan Gavioli*
UTFPR – *Campus* Medianeira
(Orientador)

Prof. Me. Neylor Michel
UTFPR – *Campus* Medianeira
(Convidado)

Prof. Me. Alessandra G. Hoffmann
UTFPR – *Campus* Medianeira
(Convidado)

Prof. M. Eng. Juliano Rodrigo Lamb
UTFPR – *Campus* Medianeira
(Responsável pelas atividades de TCC)

AGRADECIMENTOS

Agradeço primeiramente a Deus por todas as bênçãos presentes na minha vida, tais como família, amigos, pessoas exemplares que ele colocou em meu caminho e pela sua própria presença.

Ao meu orientador, Alan Gavioli, por ter aceitado o desafio prontamente comigo de desenvolver este trabalho de conclusão de curso, juntamente com todo o esforço necessário para o amadurecimento do mesmo.

Ao Professor Everton Coimbra de Araujo, por ser um exemplo de pessoa e profissional para mim em sala de aula.

Agradeço a minha namorada Tatiane por todo amor, carinho e compreensão durante a jornada da graduação, necessários para conseguir alcançar esse título de tecnólogo muito sonhado.

Aos meus pais, irmãs e sobrinhos por toda paciência e compreensão durante esse período.

Aos meus “irmãos” Fernando, Fabião, Gabriel, Gaucho e Henrique, pelos longos anos de amizade sincera, pelos milhares de conselhos e conversas construtivas.

Gostaria de terminar estes agradecimentos com a seguinte mensagem dita a mim em recente momento difícil: *“Fabio, você é empresário kra e não aceite nada menos que isso!”*, Fernando Aparecido Fovis.

RESUMO

Neste estudo de conclusão de curso, será abordado o estudo sobre a compatibilidade entre o modelo MPS.BR nível G e a metodologia de desenvolvimento Scrum, comparando o modelo MPS.BR no que seu nível mais baixo exige e o que a metodologia Scrum oferece, servindo como base para futuros esclarecimentos de dúvidas sobre essa união. O início do trabalho aborda sobre o modelo MPS.BR, seguindo pela metodologia Scrum e finalizando com a comparação prática entre os dois e a conclusão. Esperando chegar a uma estatística simples sobre essa união, servindo como base para futuros estudos e elaboração de um guia prático para micro empresas, em busca de uma maior agilidade perante o cliente, com a certificação MPS.BR.

ABSTRACT

In this study of course completion, we will describe the study on the compatibility between the model MPS.BR level G and Scrum development methodology, comparing the model MPS.BR in its lowest level and requires that Scrum offers, serving as basis for further clarification of doubts about this union. The beginning of the paper focuses on the model MPS.BR, following the Scrum methodology and ending with the comparison between the two and practical conclusion. Hoping to reach a simple statistic about this union, serving as a basis for future research and development of a practical guide for small companies, seeking greater flexibility to the customer, with the certification MPS.BR.

LISTA DE SIGLAS

CMMI	Capability maturity model integration.
SOFTEX brasileiro.	Associação para promoção da excelência do software

LISTA DE FIGURAS

Figura 1 - Construção do MPS.BR	15
Figura 2 - Estrutura do MPS.BR.....	16
Figura 3 - Níveis de maturidade	17
Figura 4 - Ciclo da metodologia Scrum.....	26
Figura 5 - Planejamento do Sprint	27
Figura 6 - Exemplo de Product Backlog.....	28
Figura 7 - Sprint Backlog	29
Figura 8 - Burndown chart.....	30

LISTA DE TABELAS

Tabela 1 - Níveis de maturidade do MPS.BR	17
Tabela 2 - Níveis de maturidade e processos.	18
Tabela 3 - Compatibilidade entre GPR do MPS.BR e Scrum	45
Tabela 4 - Compatibilidade entre GRE do MPS.BR com Scrum	46

SUMÁRIO

1	INTRODUÇÃO.....	11
1.1	OBJETIVO GERAL.....	11
1.2	OBJETIVOS ESPECÍFICOS.....	11
1.3	JUSTIFICATIVA.....	12
2	MPS.BR.....	14
2.1	ESTRUTURA MPS.BR.....	15
2.2	DESCRIÇÃO.....	17
2.3	DESCRIÇÃO DOS PROCESSOS.....	19
2.4	NÍVEL G.....	19
2.4.1	Gerência de projetos.....	19
2.4.2	Gerência de requisitos.....	22
3	SCRUM.....	23
3.1	ATIVIDADES.....	25
3.2	REUNIÕES.....	26
3.2.1	Planejamento do Sprint.....	26
3.2.2	Revisão do Sprint.....	27
3.2.3	Retrospectiva do Sprint.....	28
3.3	ARTEFATOS.....	28
3.3.1	Product backlog.....	28
3.3.2	Sprint Backlog.....	29
3.3.3	Burndown Chart.....	30
3.4	FUNÇÕES.....	30
3.5	RESPONSABILIDADES.....	31
3.5.1	Product Owner.....	31
3.5.2	Team.....	32
3.5.3	SCRUM Master.....	32
4	COMPATIBILIDADE ENTRE O MODELO MPS.BR NÍVEL G E O PROCESSO DE DESENVOLVIMENTO SCRUM.....	34
4.1	GERÊNCIA DE PROJETO (GPR).....	34
4.2	GERÊNCIA DE REQUISITOS.....	43

4.3	COMPATIBILIDADE DO MAPEAMENTO SEGUNDO AS EXTENSÕES PROPOSTAS.....	44
4.3.1	Resultados observados nesta monografia.....	45
4.4	ESTUDO DE CASO DA EMPRESA POWERLOGIC COM A APLICAÇÃO E INTEGRAÇÃO DO SCRUM COM CMMI/MPS.BR NÍVEL G.....	46
4.4.1	Histórico da Empresa.....	47
4.4.2	Evolução dos Processos na empresa PowerLogic.....	48
4.4.3	Casos de Sucesso da empresa PowerLogic.....	49
4.4.4	Desafios de compatibilidade Scrum & CMMI / MPS.BR encontrados na PowerLogic	49
4.4.5	Estudo de caso na certificação da Powerlogic.....	50
4.4.6	Melhorias Percebidas.....	52
4.4.7	Conclusão.....	53
5	CONSIDERAÇÕES FINAIS.....	54
5.1	CONCLUSÃO.....	54
5.2	TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO.....	54
	REFERÊNCIAS BIBLIOGRÁFICAS.....	55

1 INTRODUÇÃO

Nos dias atuais é crescente o número de empresas que buscam alcançar melhores resultados com o aprimoramento de seus processos produtivos e de estrutura organizacional. As empresas brasileiras de TI (Tecnologia da informação) estavam até 2003, sendo obrigadas a utilizar modelos internacionais de qualidade na busca de melhoria nos serviços, produtos e processos de software. Porém os altos custos e complexidade mostravam-se como uma grande barreira. Atento a esse problema foi criado o programa MPS.BR, sendo uma iniciativa brasileira coordenada pela Softex e que envolve universidades, grupos de pesquisa e empresas, apresentando-se como alternativa, incentivando a melhoria da qualidade e processos no desenvolvimento de software onde o foco é em micro, pequenas e médias empresas. Outra forma bastante adotada na busca por melhorias de resultado é a utilização de metodologias ágeis de desenvolvimento, que permitem entrega de funcionalidades de maior importância no menor tempo possível, como o SCRUM. O modelo MPS.BR e a metodologia SCRUM podem ser utilizadas de maneira complementar nas empresas? Dizendo respectivamente, o que fazer e como fazer, visando à melhor eficiência e eficácia dos processos? Desta forma, o presente trabalho pretende estudar a viabilidade e compatibilidade entre o modelo MPS.BR nível G e a metodologia de desenvolvimento SCRUM.

1.1 OBJETIVO GERAL

Analisar a compatibilidade entre o modelo MPS.BR nível G e a metodologia SCRUM para o desenvolvimento de software.

1.2 OBJETIVOS ESPECÍFICOS

Conhecer o modelo de melhoria de processo do software brasileiro – MPS.BR;

Compreender os níveis e processos do modelo MPS.BR;

Conhecer a metodologia de desenvolvimento SCRUM;

Conceituar a aplicação das metodologias ágeis;

Analisar se é possível compatibilizar a metodologia SCRUM em cada processo do modelo MPS.BR nível G.

1.3 JUSTIFICATIVA

Com a expansão do setor de TI a competição tornou-se muito mais acirrada e globalizada, trazendo a qualidade e tempo de resposta como um objetivo diferencial. Mas, como buscar a sobrevivência na rotina diária? Esta resposta passa por diversos temas, entre eles, manter e entregar produtos com alta qualidade nos prazos estabelecidos.

De acordo com Santana (2009), é necessário que as empresas invistam em qualidade, que por consequência exigem melhorias em seus processos e a integração de metodologias visando automatizar o desenvolvimento de software.

Para aprimorar e disseminar a melhoria de qualidade, um modelo brasileiro veio para melhorar o processo de software, o Programa MPS.BR, sendo um importante aliado para as micros, pequenas e médias empresas brasileiras, ajudando a atingirem um maior nível de capacidade e maturidade (SOFTEX, 2011, p.6).

Segundo as pesquisas realizadas pela SOFTEX (TRAVASSOS; KALINOWSKI, 2008, p.8), “Resultados de Desempenho de Organizações que Adotaram o MPS”, evidenciam diversos benefícios obtidos nas empresas que investiram em excelência de seus processos.

Outra prática que tem sido adotada significativamente por empresas de TI de todo o mundo, são as metodologias ágeis para o desenvolvimento de software, destacando-se o SCRUM, que é um *framework* de desenvolvimento baseado em iterações curtas, recomendado para projetos dinâmicos e que exigem uma entrega rápida de valor ao cliente (SCHAWBER, 2009, p.1).

São vários os benefícios para o modelo ou metodologia apresentada, mas antes dos benefícios, apresentam-se os desafios, principalmente por existir uma mudança comportamental de colaboradores e processos, onde esta mudança se faz necessário nestes contextos e uma das principais dificuldades é saber, na prática, como implantar tais modelos e metodologias.

A utilização do modelo MPS.BR juntamente com a metodologia SCRUM na prática, permanece cercada de dúvidas, principalmente por poucos trabalhos sobre o assunto, deixando a seguinte questão: é possível obter o nível G do MPS.BR utilizando o SCRUM? Pode a metodologia SCRUM auxiliar no atendimento dos propósitos e resultados esperados no modelo MPS.BR?

Esta monografia de conclusão de curso objetiva mostrar de que forma pode ser contribuída essa combinação para o sucesso das empresas no setor de TI, se o MPS.BR e o SCRUM podem caminhar juntos. Para isso, o trabalho terá foco nos processos parcialmente gerenciados do modelo MPS.BR, sendo a base para os níveis restantes.

2 MPS.BR

Segundo o MPS.BR (2011, p.6), O MPS.BR – Melhoria de Processo do Software Brasileiro iniciou seu desenvolvimento em dezembro de 2003, sendo idealizado pelas instituições SOFTEX, Riosoft, COPPE/UFRJ, CESAR, CenPRA e CELEPAR. Tendo o objetivo principal as micro, pequenas e médias empresas de software brasileiras que tem recursos financeiros escassos para investir em melhoria de processos, mas que necessitam de tais melhorias. O MPS.BR surgiu para atender as necessidades de implantar a engenharia de software nas empresas brasileiras seguindo o contexto de inserção das mesmas, visando padrões internacionais para definição, avaliação e melhoria de processos de software. O modelo de descrição é baseado em três guias:

Guia geral: descreve de maneira geral o MPS.BR, e contém detalhes sobre o modelo de referência (MR-MPS), descrevendo seus componentes e as definições necessárias para sua compreensão e aplicação.

Guia de aquisição: descreve recomendações para a execução de compras de software e serviços correlatos. Foi desenvolvido para guiar instituições que pretendem comprar software ou serviços.

Guia de avaliação: Descreve o processo de avaliação, requisitos que serão avaliados, o método e os formulários para guiar a avaliação.

O MPS.BR tem algumas diferenças em relação a outros modelos. Segundo MPS.BR (2011, p.7), essas diferenças são:

Sete níveis de maturidade, o que possibilita a implantação mais gradual e adequada a pequenas empresas;

Compatibilidade plena com CMMI e com a norma ISO/IEC 15504;

Criados para o Brasil, onde a maioria das empresas que desenvolvem software são micro, pequenas e médias;

Custo acessível;

Avaliação bienal (dois anos) das empresas;

Forte interação entre universidades e Empresas.

2.1 ESTRUTURA MPS.BR

Segundo Koscianski (2006, p.142), o modelo MPS.BR foi construído seguindo as normas NBR ISO/IEC 12207, ISO/IEC 15504 e o CMMI. Para a construção do modelo foi necessário um esforço conjunto da SOFTEX, do Governo e Universidades, para então serem utilizadas nas empresas brasileiras, como na figura a 1:

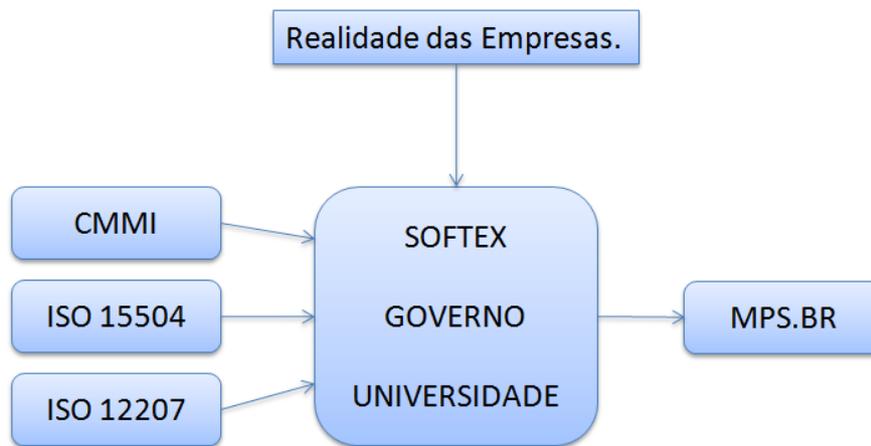


Figura 1 - Construção do MPS.BR
Fonte: Koscianski (2006).

A ISO/IEC 12207 tem como objetivo estabelecer uma estrutura para os processos de ciclo de vida do software, contendo a estrutura de processos, atividades e tarefas que servem para serem aplicadas durante a aquisição de um software, durante o fornecimento, desenvolvimento, operação e manutenção. MPS.BR (2011, p.15)

A ISO/IEC 15504 por sua vez traz um modelo de referência de processo que identifica e descreve um conjunto de processos considerados universais e fundamentais de boas práticas na engenharia de software. MPS.BR (2011, p.15)

O CMMI é o mais recente modelo de maturidade para desenvolvimento de software do SEI (Software Engineering Institute – Carnegie Mellon University - EUA), um dos maiores influenciadores em gestão de processos de software em todo mundo. MPS.BR (2011, p.15)

O MPS.BR é composto por três partes: Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS), como dito

anteriormente. Cada parte do componente foi desenvolvida com guias e documentos de projeto, conforme Koscianski (2006, p.143).

O MR-MPS (Modelo de Referência de Melhoria de Processo de Software) determina os requisitos necessários a serem cumpridos pelas organizações que desejem estar de acordo com o MR-MPS. Possui também níveis de maturidade definindo-as, dos processos e suas respectivas capacitações, que foram baseadas nas normas NBR ISO/IEC 12207 e suas emendas 1 e 2, na ISO/IEC 15504 e, a qual se adequa ao CMMI-SE/SW. Por fim, este modelo é composto também pela Guia de Aquisição, que é um documento complementar para ajudar as organizações a adquirirem um software ou serviços correlatos, conforme Koscianski (2006, p.143).

O mesmo autor ressalta que o MA-MPS (Método de Avaliação) descreve os processos de avaliações, quais serão os requisitos a serem avaliados e por fim quais requisitos estariam em conformidade com o modelo MR-MPS. Ele está detalhado no Guia de Avaliação e foi baseado no documento ISO/IEC 15504.

O MN-MPS (Modelo de Negócio) descreve regras de implantação do modelo pelas empresas consultoras, de software e avaliação, conforme Koscianski (2006, p.143).

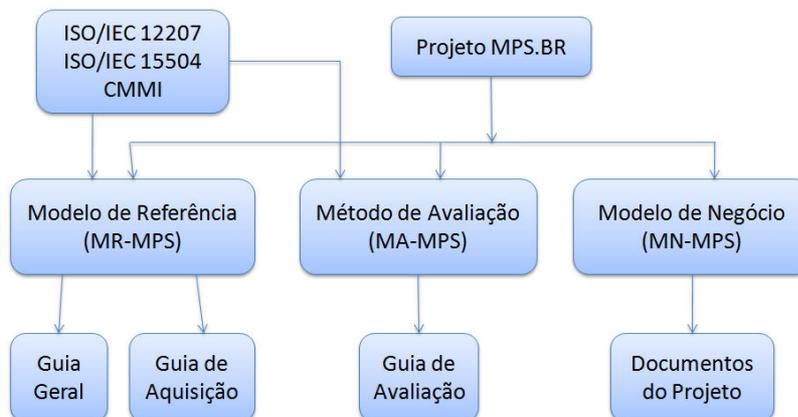


Figura 2 - Estrutura do MPS.BR
Fonte: Koscianski (2006).

A figura 2 mostra a estrutura que originou o MPS.BR, sendo os itens superiores pré-cursors, para o programa que tem-se hoje em mãos, mostrando as influencias herdadas para os itens inferiores.

2.2 DESCRIÇÃO

Segundo Koscianski (2006, p.144), o MR-MPS determina os níveis de maturidade, aonde unem os processos e a capacidade de processos, como mostra a figura 3:



Figura 3 - Níveis de maturidade
Fonte: Koscianski (2006).

Segundo Koscianski (2006, p.145), os níveis de maturidade em que a organização se encaixa, ajudam a prever seu futuro desempenho. Os níveis de maturidade estão divididos em sete, de A (maior) a G (menor), mostrando seus objetivos de melhoria, conforme a Tabela 1.:

Tabela 1 - Níveis de maturidade do MPS.BR

Fonte: (Koscianski, 2006).

A	Em Otimização
B	Gerenciado quantitativamente
C	Definido
D	Largamente definido
E	Parcialmente definido
F	Gerenciado

G	Parcialmente gerenciado
---	-------------------------

Segundo os autores do modelo, os sete níveis do MPS.BR permitem uma implantação mais tranquila e gradual do que o CMMI, aliviando a tarefa em empresas de pequeno porte. Para cada nível, é definido um perfil de processos e um perfil de capacitação de processos. Tais informações indicam onde a organização deve empenhar mais esforços para melhoria, de forma a atender os objetivos de negócio. A tabela 2 mostra os níveis .

Tabela 2 - Níveis de maturidade e processos.

Fonte: (Koscianski, 2006).

A	Inovação e implantação na organização.
	Análise de causas e resolução.
B	Desempenho do processo organizacional.
	Gerência quantitativa do projeto.
C	Análise de decisão e resolução.
	Gerência de riscos.
D	Desenvolvimento de requisitos.
	Solução técnica.
	Integração de produto.
	Instalação de produto.
	Liberação de produto.
	Verificação.
	Validação.
E	Treinamento.
	Avaliação e melhoria do processo organizacional.
	Definição do processo organizacional.
	Adaptação do processo para gerência de projeto.
F	Medição.
	Gerência de configuração.
	Aquisição.

	Garantia da qualidade.
G	Gerência de requisitos.
	Gerência de projeto.

2.3 DESCRIÇÃO DOS PROCESSOS

Cada nível de maturidade dos processos é classificado de G (mais básico) até A (mais avançado) do modelo MPS.BR, como dito anteriormente. A seguir será explicado os processos de nível G, mostrando os resultados obtidos pela execução de cada processo na organização começando pelo mais básico até o mais avançado, também mostrando os resultados obtidos pela execução de cada processo de forma correta na organização.

2.4 NÍVEL G

O MPS.BR nível G é o mais baixo do modelo, sendo composto apenas pelas práticas de gerência de projetos e gerência de requisitos, conforme Koscianski (2006, p.146).

2.4.1 Gerência de projetos

O objetivo do processo Gerência de Projetos é executar as tarefas e recursos envolvidos no projeto de um serviço e/ou produto, tais como, identificar, estabelecer, coordenar e monitorar as atividades. Esse processo tem resultado esperado em nove áreas: aquisição, comunicação e qualidade, custos, escopo, integração, recursos humanos, riscos e o tempo, conforme Koscianski (2006, p.146).

Espera-se com este processo, melhorias tais como no escopo, aonde será melhor definido, estimando mais corretamente as tarefas e alocação adequada dentro da equipe. O que conseqüentemente traz cronogramas mais controláveis e realísticos. Com o cronograma mais preciso, os custos são melhores estipulados e os recursos a serem adquiridos para o projeto, tais como software e hardware. Os

riscos são identificados e analisados, então são gerados os planos para que tais riscos sejam minimizados ou até evitados, conforme Koscianski (2006, p.146).

Existem diversas ferramentas automatizadas que podem ajudar no cumprimento dessas tarefas, tais como facilitadores de reuniões e gerenciadores de cronograma, que ajudam, por exemplo, no gerenciamento de reuniões de brainstorming, conforme Koscianski (2006, p.146).

Segundo (Guia Geral, 2011, p.25), ações pertinentes para a implantação deste processo são:

Definir o escopo do trabalho para o projeto;

Utilizando métodos apropriados, dimensionar as tarefas e produtos de trabalho do projeto;

Definir o modelo e as fases do ciclo de vida do projeto;

(Até o Nível F) Com base em dados históricos ou referências técnicas, estimar o esforço e o custo para a execução das tarefas e dos produtos de trabalho;

(Apartir do Nível E) Baseando no repositório de estimativas e no conjunto de ativos de processo organizacional, fazer o planejamento e as estimativas das atividades do projeto;

Estabelecer e manter o orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle;

Identificar os riscos do projeto e o seu impacto, probabilidade de ocorrência e prioridade de tratamento, determinando-os e documentando os mesmos;

Planejar os recursos humanos para o projeto, considerando o perfil e o conhecimento necessário para executá-lo;

Planejar os recursos e o ambiente de trabalhos necessários para execução do projeto;

Identificar os dados relevantes do projeto e planejar quanto à forma de coleta, armazenamento e distribuição. Estabelecer um mecanismo para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;

Com a integração de planos específicos, estabelecer um plano geral para a execução do projeto;

Avaliar a viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis. Se necessário, realizar ajustes;

Revisar o plano do projeto com todos os interessados e obter compromisso com ele;

Utilizar o plano do projeto para gerenciá-lo e outros planos que afetam o e documentar os resultados;

Gerenciar o envolvimento das partes interessadas no projeto;

Realizar revisões em marcos do projeto, conforme estabelecido no planejamento;

Estabelecer e tratar com as partes interessadas os registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas;

Estabelecer, implementar e acompanhar ações para corrigir desvios em relação ao planejamento e para prevenir a repetição dos problemas identificados, até a sua conclusão;

(A partir do nível E) Estabelecer um processo definido para o projeto, de acordo com a estratégia para adaptação do processo da organização;

(A partir do nível B) Com base na estabilidade histórica, em dados de capacidade e em outros critérios previamente estabelecidos, selecionar os subprocessos mais adequados para compor o processo definido para o projeto;

(A partir do nível E) Contribuir para os ativos de processo organizacional, produtos de trabalho, medidas e experiências documentadas;

(A partir do nível B) Estabelecer e manter os objetivos para a qualidade do produto e para o desempenho do processo definido para o projeto;

(A partir do nível B) Escolher os subprocessos do processo definido para o projeto e que serão gerenciados estatisticamente e identificar os atributos por meio dos quais cada subprocesso será gerenciado estatisticamente;

(A partir do nível B) Monitorar o projeto para determinar se seus objetivos para qualidade e para o desempenho do projeto é atingido. Quando necessário, ações corretivas são identificadas;

(A partir do nível B) Estabelecer e manter o entendimento da variação dos subprocessos escolhidos para gerência quantitativa, utilizando medidas e técnicas de análise estatisticamente previamente selecionadas;

(A partir do nível B) Monitorar o desempenho dos subprocessos escolhidos para gerência quantitativa, determinar a sua capacidade de satisfazer os seus

objetivos para qualidade e para o desempenho. Identificar ações quando for necessário tratar deficiências dos subprocessos;

(A partir do nível B) Incorporar ao repositório de medidas da organização, dados estatísticos e de gerência da qualidade.

2.4.2 Gerência de requisitos

Na grande maioria do desenvolvimento de software os requisitos são mutáveis, por diversos fatores ocorrem essas alterações: mudanças tecnológicas, de legislação ou de estrutura organizacional. Porém, as adaptações são o fator mais comum de alterações de requisitos e mudanças solicitadas pelos stakeholders, conforme Koscianski (2006, p.146).

É importante salientar que as alterações de requisitos são muito comuns em projetos e devem ser bem recebidas desde que se tenha um controle, conforme Koscianski (2006, p.146).

De acordo com Kotonya e Sommerville (1996, p.19), algumas das características a considerar para que uma especificação tenha qualidade são a rastreabilidade, a consistência e a precisão.

Segundo (Guia Geral, 2011, p.28), ações pertinentes para a implantação deste processo:

Entender, avaliar e aceitar os requisitos, junto aos fornecedores de requisitos, utilizando critérios e objetivos;

Obter o comprometimento da equipe técnica com os requisitos aprovados;

Estabelecer e manter a rastreabilidade bidirecional entre os requisitos e os produtos de trabalho;

Realizar revisões em planos e produtos de trabalho do projeto, visando identificar e corrigir inconsistências em relação aos requisitos;

Gerenciar ao longo do projeto as mudanças nos requisitos.

3 SCRUM

Scrum foi criado como um estilo de gerenciamento de projetos nas empresas de fabricação de automóveis e produtos de consumo, por Takeuchi e Nonaka (1986) e publicadas no artigo "The New Product Development Game" na revista Harvard Business Review Janeiro-Fevereiro de 1986. Eles perceberam que projetos utilizando equipes pequenas e multidisciplinares produziram os melhores resultados, e associaram estas equipes altamente eficientes à formação de uma jogada do *Rugby*, que exige colaboração do time inteiro para a sua execução. O primeiro teste do Scrum foi realizado em 1993, na empresa de desenvolvimento de software Easel, onde foi documentado, implementado e incorporado os estilos de gerenciamento observados por Takeuchi e Nonaka. Em 1995, a definição de Scrum foi formalizada por Ken Schwaber, que ajudou a implantá-lo no desenvolvimento de softwares em todo o mundo, conforme SCHAWBER (2009, p.3).

O Scrum segue os princípios do desenvolvimento ágil, que são:

“Garantir a satisfação do consumidor entregando rapidamente e continuamente softwares funcionais;”

“Softwares funcionais são entregues frequentemente (semanas, ao invés de meses);”

“Softwares funcionais são a principal medida de progresso do projecto;”

“Até mesmo mudanças tardias de escopo no projeto são bem-vindas.”

“Cooperação constante entre pessoas que entendem do 'negócio' e desenvolvedores;”

“Projetos surgem através de indivíduos motivados, entre os quais existe relação de confiança.”

“Design do software deve prezar pela excelência técnica;”

“Simplicidade;”

“Rápida adaptação às mudanças;”

“Indivíduos e interações mais do que processos e ferramentas;”

“Software funcional mais do que documentação extensa;”

“Colaboração com clientes mais do que negociação de contratos;”

“Responder a mudanças mais do que seguir um plano.” Conforme <http://engenharia-de-software.nuvvo.com/lesson/11123-desenvolvimento-agil-de-software> (acessado em 19 de agosto de 2011).

Scrum é uma abordagem ágil para desenvolvimento de software. Ao invés de um processo completo ou metodologia, é um *framework*¹. Então, ao invés de fornecer uma descrição completa e detalhada de como tudo deve ser feito no projeto, deixa esta tarefa para a equipe de desenvolvimento de software. Isso é feito porque a equipe saberá a melhor forma de resolver os problemas que serão apresentados. É por isso que, por exemplo, uma reunião de planejamento do *sprint* é descrito em termos de resultado desejado (o compromisso de conjunto de recursos a serem desenvolvidas no próximo *sprint*) em vez de um conjunto de critérios de entrada, definições de tarefas, critérios de validação e critérios de saída (ETVX), como seria previsto na maioria das metodologias, *sprint* é um ciclo, de atividades que devem ser executado, conforme Cohn(1998, p.1).

O Scrum depende de uma auto-organização. A equipe Scrum é auto-organizável em que não há nenhum líder geral da equipe que vai decidir qual pessoa vai fazer qual tarefa ou como um problema será resolvido. Estas são questões que serão decididas pela equipe como um todo. A equipe é multifuncional, para que todos os envolvidos possam ter idéias referentes ao projeto, conforme Cohn(1998, p.1).

Essas equipes ágeis de desenvolvimento são apoiadas por dois indivíduos específicos: o *Scrum Master* e *Product Owner*. O *Scrum Master* pode ser pensado como um treinador para a equipe, ajudando os membros da equipe a utilizar Scrum para executar em seu mais alto nível. O *Product Owner* representa a empresa, clientes ou usuários e orienta a equipe para construir o produto correto, conforme Cohn(1998, p.1).

Projetos Scrum progredem em uma série de *sprints*, que são *timebox* de iterações não mais do que um mês. No início de uma *sprint*, os membros da equipe comprometem a entregar um número de características que foram listadas no projeto do *Product Backlog*. No final do *sprint*, esses recursos estão feitos, ou seja, estão codificados, analisados e integrados no produto em desenvolvimento ou sistema. No final do *sprint* uma revisão da *Sprint* é realizada durante o qual a equipe

demonstra a nova funcionalidade para o proprietário do produto e outras partes interessadas que fornecem *feedback* que podem influenciar o próximo *sprint*, conforme Cohn(1998, p.1).

3.1 ATIVIDADES

O artefato básico de um projeto Scrum, é o próprio produto. A equipe deverá trazer o produto ou o sistema para um estado potencialmente utilizável no final de cada *sprint*.

No desenvolvimento ágil, o *Product Backlog* é uma lista completa das funcionalidades que continua a ser adicionado ao produto. O *Product Backlog* é priorizado pelo *Product Owner* para que a equipe trabalhe sempre nos mais importantes recursos em primeiro lugar. A popular e bem sucedida maneira para criar um *Product Backlog* é preenchê-la com *user stories*, que são breves descrições de funcionalidades a partir da perspectiva de um usuário ou cliente.

No primeiro dia de uma *sprint* e durante a reunião de planejamento do *sprint*, os membros da equipe criam o *sprint backlog*. O *sprint backlog* pode ser pensado como a equipe de lista de tarefas para o *sprint*. Considerando que o *Product Backlog* é uma lista de recursos a ser construído (muitas vezes escrito na forma de *user stories*), a *Sprint Backlog* é a lista de tarefas que a equipe precisa realizar para oferecer a funcionalidade que se comprometeram a entregar durante o *sprint*.

Dois outros artefatos primários são o gráfico de *sprint burndown* e gráfico de *release Burndown*. Gráficos *Burndown* mostram a quantidade de trabalho restante seja em um *sprint* ou uma *release*. Eles são uma ferramenta muito eficaz para determinar rapidamente se um *sprint* ou *release* está na programação para que todos os trabalhos previstos concluída até a data desejada.

¹ Framework é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica.



Figura 4 - Ciclo da metodologia Scrum

Fonte: Cohn (1998)

Na figura 4 tem-se o ciclo de vida do Scrum, aonde o *Product Backlog* é elaborado junto com o *Product Owner*, ou seja, o maior interessado do produto final do projeto, os requisitos geram uma(s) *Sprint Backlog*, o qual é uma lista de tarefas que a equipe se compromete a fazer. Estas tarefas devem ser calculadas para durar entre 2 a 4 semanas, tendo todo dia reuniões diárias e desta forma obtendo os resultados pouco a pouco, planejados no *Product Backlog*.

3.2 REUNIÕES

3.2.1 Planejamento do Sprint

O objetivo do planejamento do *Sprint* é analisar e avaliar o *Product backlog* e definir o objetivo do Sprint. Aonde a equipe seleciona os itens do Product Backlog com os quais se comprometem a concluir, formando o *Sprint Backlog*, segundo Cohn(1998, p.3).

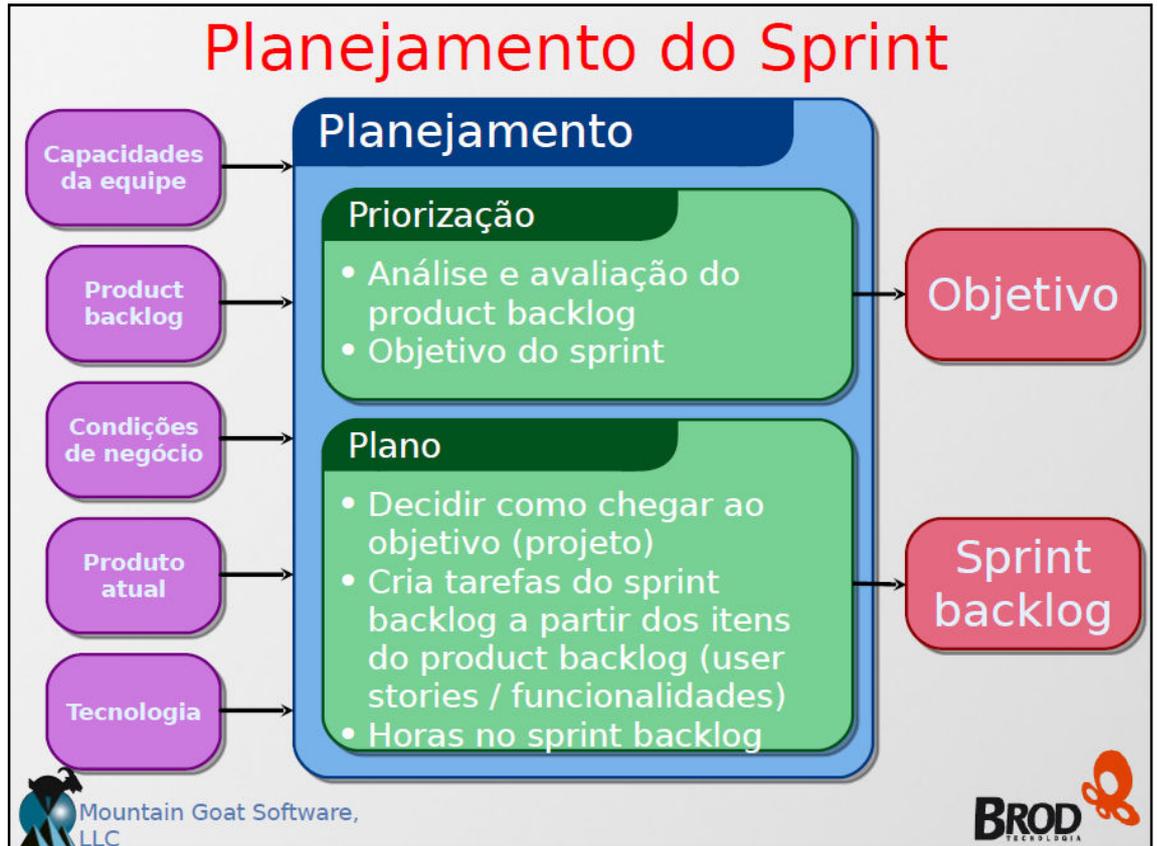


Figura 5 - Planejamento do Sprint

Fonte: Cohn (1998)

A figura 5 pode elucidar melhor como ocorre o planejamento do *Sprint*, onde tem-se como entrada para a elaboração, a capacidade da equipe, o *Product Backlog*, as condições de negócio, o Produto Atual e a Tecnologia, o planejamento do *Sprint* e seguindo pela saída, que podemos identificar também como resultados desta etapa, onde obtemos o Objetivo do *Sprint* e o *Sprint Backlog* que caracteriza a conclusão do Planejamento.

3.2.2 Revisão do Sprint

Nas revisões do *Sprint* a equipe apresenta os resultados obtidos no percorrer do *Sprint*, aonde são demonstrada as novas funcionalidades ou sua arquitetura. As revisões são informais, tendo em média duas horas de preparação e sem o uso de *slides*. O time todo participa e todos os envolvidos são convidados, conforme Cohn(1998, p.4).

3.2.3 Retrospectiva do Sprint

A retrospectiva do *Sprint* ocorre periodicamente e é observado o que funciona e o que não funciona, essa reunião levam aproximadamente 30 minutos e é feita após cada *Sprint*, destacando que toda a equipe participa, discutindo o que gostaria de iniciar a fazer, parar de fazer e continuar fazendo, conforme Cohn(1998, p.4).

3.3 ARTEFATOS

3.3.1 Product backlog

Os requerimentos são listados em uma lista de todo o trabalho desejado no projeto, na forma em que cada item tenha seu peso de acordo com a vontade do cliente ou usuários, priorizado pelo dono do produto aonde é repriorizado no início de cada *Sprint*, conforme Cohn(1998, p.5).

Exemplo de Product Backlog

Item do Backlog	Estimativa
Permitir que o usuário faça uma reserva	3
Permitir que o usuário cancele a reserva	5
Permitir a troca de datas da reserva	3
Permitir que empregadod do hotel gerem relatórios de lucratividade	8
Melhorar manipulação de erros	8
...	30
...	50

Mountain Goat Software, LLC 

Figura 6 - Exemplo de Product Backlog

Fonte: Cohn (1998)

A figura 6 exemplifica um *Product Backlog*, que tem-se os Itens do *backlog* com sua estimativa de prioridade, lembrando do conceito de Schawber (2004), que o *Product Backlog* nunca está por finalizado, enquanto o produto estiver em uso ou construção.

3.3.2 Sprint Backlog

No *Sprint Backlog* cada indivíduo escolhe o trabalho que fará assim estes nunca serão atribuídos, atualizando diariamente a estimativa do trabalho restante para que qualquer membro da equipe possa adicionar, apagar ou mudar tarefas. O trabalho aparece a partir do *Sprint*, desta forma, se uma tarefa não é clara, define-se como um item com uma quantidade maior de tempo e subdivide-a depois, atualizando as coisas a serem feitas na medida em que se tornam mais conhecidas, conforme Cohn(1998, p.5).

Sprint Backlog					
Tarefas	Seg	Ter	Qua	Qui	Sex
Codificar interface de usuário	8	4	8		
Codificar regra de negócio	16	12	10	4	
Testar	8	16	16	11	8
Escrever help online	12				
Escrever a classe foo	8	8	8	8	8
Adicionar log de erros			8	4	



Figura 7 - Sprint Backlog

Fonte: Cohn (1998)

Na figura 7 tem-se um exemplo de *Sprint Backlog*, que cada indivíduo assume a responsabilidade de trabalho e atualizando a estimativa de trabalho final.

3.3.3 Burndown Chart

O Burndown chart é um gráfico de evolução do projeto feito a partir do *Sprint Backlog* aonde se permite ter uma maior visão sobre o andamento do mesmo, conforme Cohn(1998, p.5).

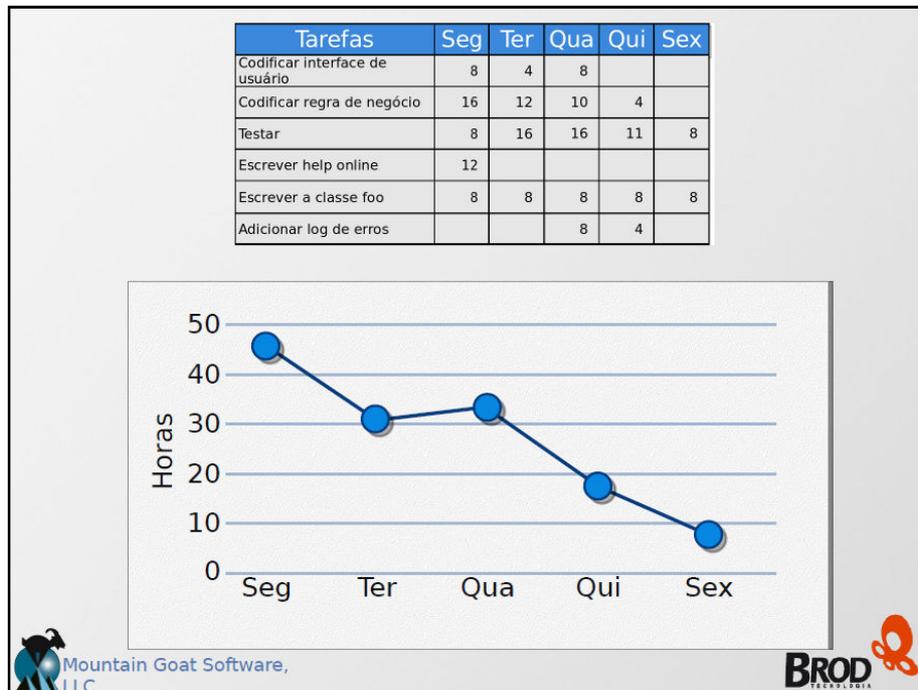


Figura 8 - Burndown chart

Fonte: Cohn (1998)

A figura 8 exemplifica um gráfico de *Burndown*, este gráfico mostra a evolução pessoal de cada *Sprint Backlog*, fornecendo uma visão melhor sobre o andamento do mesmo.

3.4 FUNÇÕES

O principal participante da equipe Scrum é o papel chamado *Scrum Master*. O *Scrum Master* é o treinador da equipe e ajuda os membros da equipe alcançar seu mais alto nível de desempenho. Um *Scrum Master* difere de um gerente de projeto tradicional em muitas maneiras importantes, incluindo que não fornece orientação no dia-a-dia para a equipe e não atribui tarefas a indivíduos. Um bom *Scrum Master* abriga a equipe de distrações externas, permitindo que os

membros da equipe concentrar-se fortemente durante o *sprint* sobre o objetivo que eles escolheram.

Enquanto o *Scrum Master* se concentra em ajudar a equipe a ser o melhor que pode ser, o *Product Owner* designa o trabalho para dirigir a equipe na meta certa. O *Product Owner* faz isso criando uma visão convincente do produto e, em seguida, transmite essa visão para a equipe através do *Product Backlog*.

O *Product Owner* é responsável por garantir que o *Product Backlog* permaneça com prioridades corretas e se ensine sobre as regras do sistema que está sendo construído, seus usuários, a equipe, e assim por diante.

Por último, a própria equipe. Embora os indivíduos de uma equipe Scrum possam chegar a essa equipe com cargos diferentes, enquanto no time os títulos são insignificantes. Cada pessoa contribui da melhor maneira possível para completar o trabalho de cada *sprint*. Isso não significa que um testador irá remodelar o sistema, os indivíduos passam a maior parte (e às vezes todos) de seu tempo trabalhando em qualquer disciplina que trabalhou antes de adotar Scrum. Mas em uma equipe Scrum, os indivíduos devem trabalhar além de suas disciplinas preferidas, sempre fazendo o necessário para o bem da equipe.

Uma maneira conveniente de pensar sobre a natureza destas funções é o exemplo de um carro de corrida. O time é o próprio carro, pronto para acelerar ao longo de qualquer direção é aguçado. O *Product Owner* é o motorista, fazendo com que o carro esteja sempre indo na direção certa. O *Scrum Master* é o mecânico-chefe, mantendo o carro bem afinado e executar o seu melhor.

3.5 RESPONSABILIDADES

3.5.1 Product Owner

Conforme (YOSHIMA, 2007, p.10), o *Product Owner* tem as seguintes responsabilidades:

- Definir funcionalidades do produto;

- Concentrar informações de usuários, *stakeholders* ou do mercado, objetivando ter uma visão única dos requisitos do sistema;

- ROI do projeto;

Priorizar o Product Backlog;

Caso necessário, alterar prioridades fora do *Sprint*;

Aceitar ou rejeitar resultados dos trabalhos.

A característica do *Product Owner* é ser o principal interessado no projeto, cabendo a ele financiar os custos do produto a ser desenvolvido.

3.5.2 Team

Conforme (YOSHIMA, 2007, p.10), o *Team* tem as seguintes responsabilidades:

Garantir que o projeto seja entregue com todas as funcionalidades necessárias.

Definir e especificar o objetivo do *Sprint* e os resultados dos trabalhos;

Demonstrar o resultado do *Sprint* ao *Product Owner* e *Stakeholders*.

Apresentam as seguintes características:

É multifuncional;

Composto por aproximadamente 7 pessoas;

Fazer o que for necessário dentro das diretrizes do projeto para concluir o objetivo do *Sprint*;

É Auto-organizável.

3.5.3 SCRUM Master

Conforme (YOSHIMA, 2007, p.10), o *Scrum Master* tem as seguintes responsabilidades:

Melhorar a produtividade e a vida do time de desenvolvimento, incentivando a criatividade e o conhecimento;

Estimular a cooperação e comunicação entre o time;

Impedir que interferências externas atrapalhem o time;

Remover entraves e impedimentos no projeto;

Garantir que o processo está sendo executado corretamente;

Trazer pessoas apropriadas para as reuniões de acompanhamento (*Daily SCRUM, Sprint Review e Sprint Retrospective*);

Retirar entraves entre o desenvolvimento e o cliente a fim de garantir que o cliente está direcionando as funcionalidades desenvolvidas;

Ajudar o *Product Owner* a maximizar o ROI atingindo os seus objetivos com o SCRUM;

Aplicar práticas de engenharia para que cada pedaço de funcionalidade seja potencialmente implantável.

4 COMPATIBILIDADE ENTRE O MODELO MPS.BR NÍVEL G E O PROCESSO DE DESENVOLVIMENTO SCRUM

Nesta parte é realizado um comparativo entre o modelo MPS.BR nível G e o processo de desenvolvimento Scrum, detalhando o que é requerido pelo MPS.BR e o que está disponível seguindo os conceitos Scrum.

4.1 GERÊNCIA DE PROJETO (GPR)

O objetivo do processo Gerência de Projetos é executar as tarefas e recursos envolvidos no projeto de um serviço e/ou produto, tais como, identificar, estabelecer, coordenar e monitorar as atividades. Esse processo tem resultado esperado em nove áreas: aquisição, comunicação e qualidade, custos, escopo, integração, recursos humanos, riscos e o tempo. (Koscianski, 2006, p.146).

GPR1: Definir o escopo do trabalho para o projeto.

Definir o escopo do trabalho para o projeto, é fazer o levantamento de todas as necessidades, características e funções necessárias para satisfazer a expectativa dos stakeholders, deixando claro o objetivo e os limites levantados.

- Recursos Scrum compatíveis:
 - Planejamento do Scrum, que define as expectativas das partes interessadas, ou seja, o planejamento é uma maneira de sincronizar as expectativas entre as partes interessadas com as da equipe. Para ter o Planejamento é necessário o documento de Visão e o *Product Backlog*, lembrando que o documento de visão descreve por que o projeto está sendo realizado e o estado final desejado e o *Product Backlog* define os requisitos funcionais e não funcionais que o sistema deverá reunir para realizar o documento de visão, priorizando e estimando os requisitos. (SCHAWBER, 2004)

- Compatibilidade:
 - O artefato Planejamento do Scrum pode atender perfeitamente este requisito.

GPR2: Utilizando métodos apropriados, dimensionar as tarefas e produtos de trabalho do projeto.

Dividir o escopo em partes menores para facilitar o gerenciamento, elaborando uma estrutura de fracionamento fornecendo uma referência ao esforço, cronograma e atividades que serão desempenhadas, visando planejar e organizar melhor as tarefas.

- Recursos Scrum compatíveis:
 - O Scrum não tem nenhum artefato ou cerimônia que possa atender perfeitamente ou parcialmente este requisito. (SIQUEIRA, 2007, p.30)
- Compatibilidade:
 - Não pode ser compatível.

GPR3: Definir o modelo e as fases do ciclo de vida do projeto.

Definir o modelo e as fases do ciclo de vida do projeto, serve para definir um conjunto de fases aonde cada fase tem suas atividades para ser atingidas e assim avançar de fase, Permitindo assim maior controle sobre o planejamento do projeto.

- Recursos Scrum compatíveis:
 - O ciclo de vida no Scrum é dividido em quatro fases, Planejamento, Preparação, Desenvolvimento e Entrega, aonde o Planejamento estabelece a visão do projeto e as expectativas, a Preparação serve para avaliar as dimensões do projeto, o Desenvolvimento para funcionalidades iterativas e incrementais através das *Sprints* e a Entrega do produto final ao cliente. (LARMAN, 2004).
- Compatibilidade:
 - O ciclo de vida do Scrum pode atender perfeitamente este requisito.

GPR4: Com base em dados históricos ou referências técnicas, estimar o esforço e o custo para a execução das tarefas e dos produtos de trabalho.

Esta estimativa pode ser afetada pelos parâmetros de produtividade.

- Recursos Scrum compatíveis:
 - *Product Backlog*, que é uma lista de requisitos com estimativa de esforço, nunca estando completa, desta forma impossibilitando a estimativa precisa ou definitiva, ou seja, servindo apenas para fornecer uma visão ao *Product Owner*. (SCHAWBER, 2004)
 - *Sprint Backlog*, que define as tarefas, dando visibilidade ao desenvolvimento. (SCHAWBER, 2004)
 - Gráfico *BurnDown*, que mostra dados históricos em qualquer ponto do tempo e do progresso da equipe no projeto. (SCHAWBER, 2004)
 - O Scrum não faz recomendações de métodos formais de avaliação, como pontos de função, para estimativas e custos dos produtos. (SIQUEIRA, 2007, p.30)
- Compatibilidade:
 - O Scrum pode atender parcialmente este requisito.

GPR5: Estabelecer e manter o orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle.

Possíveis alterações e dependência entre as tarefas devem ser identificadas com métodos específicos, tendo base no cronograma e na estimativa de custos, para elaborar o orçamento do projeto. (SIQUEIRA, 2007, p.31)

- Recursos Scrum compatíveis:
 - *Product Backlog*, que informa os primeiros parâmetros referentes ao cronograma, através da estimativa dos *Sprints*, lembrando que o cronograma fica dinâmico pelos acréscimos naturais no *Product Backlog*. (SCHAWBER, 2004)
 - Gráfico *BurnDown*, que mostra a relação entre trabalho restante em qualquer ponto do tempo e do progresso da equipe no projeto. (SCHAWBER, 2004)

- Compatibilidade:
 - O orçamento não é uma atividade atendida no Scrum, assim é impossível ter base deste item por cronogramas. (SIQUEIRA, 2007, p.31)
 - GPR6: Identificar os riscos do projeto e o seu impacto, probabilidade de ocorrência e prioridade de tratamento, determinando-os e documentando os mesmos.
 - O objetivo deste requisito é analisar, classificar e identificar os riscos e suas prováveis conseqüências por priorização.

- Recursos Scrum compatíveis:
 - Reuniões diárias, que traz os impedimentos do trabalho de cada membro da equipe, juntamente com um breve progresso e planejamento do trabalho. (SCHAWBER, 2004)
 - *Impediments Backlog*, que é um registro de impedimentos do time, feito a partir das reuniões diárias.

- Compatibilidade:
 - O Scrum não tem artifícios explícitos para identificar e monitorar os riscos do projeto, assim pode atender parcialmente este requisito. (SIQUEIRA, 2007, p.31)

GPR7: Planejar os recursos humanos para o projeto, considerando o perfil e o conhecimento necessário para executá-lo.

O objetivo deste requisito é determinar funções e relações hierárquicas para cada envolvido no projeto, sendo externa ou interna a empresa, levantando informações sobre a capacidade para execução das tarefas, caso necessário deverá ter um treinamento.

- Recursos Scrum compatíveis:
 - *Product Backlog*, que é uma lista de requisitos a serem feitos, caso a equipe não tenha capacidade de fazer algum requisito ou criar o próprio *Product Backlog*, deverá ser inserido um requisito de treinamento no *Product Backlog*. (SIQUEIRA, 2007, p.32)

- *Sprint Backlog*, que é uma lista aonde ocorre uma distribuição das tarefas de cada *Sprint*, feita pela equipe, que por si só é alto-organizável. (SCHAWBER, 2004)

- Compatibilidade:

- O Scrum pode atender perfeitamente este requisito.

GPR8: Planejar os recursos e o ambiente de trabalhos necessários para execução do projeto.

O objetivo é especificar e planejar os recursos e tarefas necessárias para atender, tais como ferramentas, serviços, componentes ou despesas.

- Recursos Scrum compatíveis:

- *Scrum Master*, que tem por responsabilidade planejar as reuniões e remover possíveis impedimentos levantados pela equipe. conforme Cohn(1998, p.5).
- *Product Owner*, que tem por responsabilidade levantar os requisitos para a elaboração do *Product Backlog*. Conforme Cohn(1998, p.5).

- Compatibilidade:

- O Scrum pode atender perfeitamente este requisito.

GPR9: Identificar os dados relevantes do projeto e planejar quanto à forma de coleta, armazenamento e distribuição. Estabelecer um mecanismo para acessá-los, incluindo, se pertinente, questões de privacidade e segurança.

O objetivo é identificar e armazenar os dados do projeto, sendo eles, análises, atas de reuniões, relatórios, estudos, estando eles em qualquer formato. Sendo necessária a identificação dos dados mais relevantes para armazenamento.

- Recursos Scrum compatíveis:

- Reuniões. Durante as reuniões a comunicação é pessoalmente feita, com documentações.

- Compatibilidade:

- O Scrum não atende este requisito, pois não existem orientações referentes a gerência de coleta de dados no projeto e distribuição

dos mesmos, assim como armazenamento, proteção e recuperação de dados.

GPR10: Com a integração de planos específicos, estabelecer um plano geral para a execução do projeto.

O objetivo do plano de projeto é manter documentos com informações sobre o projeto, tais como ciclo de vida, cronograma, esforço tamanho, orçamento e riscos no projeto.

- Recursos Scrum compatíveis:
 - Documento de visão, que descreve por que o projeto está sendo realizado e proporciona uma visão do estado final desejado. (SCHAWBER, 2004)
 - *Product Backlog*, que é uma descrição dos requisitos funcionais e não funcionais que o sistema deverá reunir para proporcionar o documento de visão. (SCHAWBER, 2004)
 - Estes dois artifícios mostram o projeto em alto-nível.
 - Sendo conveniente também o Sprint Backlog, gráfico de Burndown e Impediments backlog.

- Compatibilidade:
 - O Scrum pode atender perfeitamente este requisito.

GPR11: Avaliar a viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis. Se necessário, realizar ajustes.

Avaliando o projeto no início, visando os objetivos e características dos resultados, observando aspectos financeiros, técnicos e humanos, conforme o progresso do projeto. Verificar a disponibilidade dos requisitos para atingir a meta.

- Recursos Scrum compatíveis:
 - *Sprint*, a equipe se reúne para avaliar a viabilidade dos requisitos. (SCHAWBER, 2004)
 - Reuniões de Planejamento, que definem as funcionabilidades.
 - Reuniões de Retrospectiva, que responsabilizam em fazer ajustem.

- Compatibilidade:
 - O Scrum pode atender perfeitamente este requisito.

GPR12: Revisar o plano do projeto com todos os interessados e obter compromisso com ele.

O objetivo é revisar o plano do projeto para que todos os envolvidos consigam alcançar suas metas, se necessário conciliar possíveis diferenças entre os envolvidos.

- Recursos Scrum compatíveis:
 - No Scrum, o plano de projeto é representado em no documento de visão e *Product Backlog*, que é a representação do projeto em alto-nível. (SIQUEIRA, 2007, p.35)
 - Planejamento do *Sprint*, que tem por objetivo analisar e avaliar o *Product backlog* e definir o objetivo do *Sprint*. conforme Cohn(1998, p.5).
 - Retrospectiva do *Sprint*, que ocorre periodicamente e é reportado o que funciona e o que não funciona. Conforme Cohn(1998, p.5).
 - Desta forma, a Equipe, *Scrum Master* e *Product Owner* são responsáveis por revisar o plano de projeto. (SIQUEIRA, 2007, p.35)

- Compatibilidade:
 - O Scrum pode atender perfeitamente este requisito.

GPR13: Utilizar o plano do projeto para gerenciá-lo e outros planos que afetam o e documentar os resultados.

Revisar sempre a avaliação dos planos do projeto, procurando coletar dados sobre o que foi realizado em relação ao cronograma e os esforços necessários para o mesmo.

- Recursos Scrum compatíveis:
 - *Product Backlog Burndown*, que monitoram o andamento do projeto, de *Sprint* a *Sprint*, servindo para avaliar o processo e melhor estimar o tempo de previsão. (SCHAWBER, 2004)

- Reuniões diárias, que a equipe comenta diariamente como esta o andamento dos requisitos da *Sprint* assumida. (SCHAWBER, 2004)
- *Sprint Burndown*, mostrando a velocidade da equipe, mostrando tempo estimado e tempo decorrido. (SCHAWBER, 2004)
- Compatibilidade:
 - O Scrum pode atender perfeitamente este requisito.

GPR14: Gerenciar o envolvimento das partes interessadas no projeto.

Descrever todos os interessados no projeto, tais como, clientes, usuários, membros, diretoria, e a fase em que serão necessários e seus respectivos comprometerimentos, trazendo assim informações sobre prazo, custo e recursos necessários.

- Recursos Scrum compatíveis:
 - Scrum, que tem suas práticas e atividades descritas, assim como papéis e responsabilidades de cada envolvido. (SIQUEIRA, 2007, p.35)
 - *Product Owner*, que é o principal interessado, tendo por responsabilidade definir o escopo e as prioridades do projeto. conforme Cohn(1998, p.5).
 - *Scrum Master*, que é o responsável pelo cumprimento das regras do Scrum e garantir a participação dos envolvidos. Conforme Cohn(1998, p.5).
- Compatibilidade:
 - O Scrum pode atender perfeitamente este requisito.

GPR15: Realizar revisões em marcos do projeto, conforme estabelecido no planejamento

Planejar no início e fim de cada fase do projeto ou em atividades importantes a revisão, verificando se podem atender os requisitos exigidos na próxima fase ou se todos os critérios foram atendidos.

- Recursos Scrum compatíveis:
 - *Sprint Review*, no final de todas as *Sprint*, a equipe mostra para o *Product Owner* o progresso e o cumprimento das atividades do projeto. (SCHAWBER, 2004)
- Compatibilidade:
 - O Scrum pode atender perfeitamente este requisito.

GPR 16: Estabelecer e tratar com as partes interessadas os registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas.

Monitorar possíveis imprevistos e problemas no projeto, analisando, registrando e gerenciando sua resolução, tendo como base os planos de ações.

- Recursos Scrum compatíveis:
 - *Impediments Backlog*, os problemas encontrados pelo time, são analisados e reportados. (SIQUEIRA, 2007, p.36)
 - *Scrum Master*, que tem por responsabilidade identificar e remover entraves e impedimentos no projeto. Conforme Cohn(1998, p.5).
- Compatibilidade:
 - O Scrum pode atender perfeitamente este requisito.

GPR 17: Estabelecer, implementar e acompanhar ações para corrigir desvios em relação ao planejamento e para prevenir a repetição dos problemas identificados, até a sua conclusão.

Conforme os problemas forem aparecendo, as suas respectivas ações de correção devem ser analisadas, registrando e acompanhando até sua resolução. Se ações forem necessárias, deve ser analisadas e registradas também.

- Recursos Scrum compatíveis:
 - *Scrum Master*, que também é responsável por ações de correção, mas no Scrum não se tem registro do acompanhamento ou planejamento dessas ações. (SIQUEIRA, 2007, p.36)
- Compatibilidade:
 - O Scrum pode atender parcialmente este requisito.

4.2 GERÊNCIA DE REQUISITOS

O principal objetivo deste processo é gerenciar os requisitos recebidos e gerados no projeto, identificando incoerências nos requisitos, planos de projeto e produtos de trabalho.

GRE1: Entender, avaliar e aceitar os requisitos, junto aos fornecedores de requisitos, utilizando critérios e objetivos.

O objetivo é definir claramente os requisitos do projeto com as pessoas que fornecem as informações, identificando-os e aprovando junto ao patrocinador do projeto.

Plano de projeto, que mantêm o registro de *Product Owner* e como é feita a comunicação com o mesmo, deixando-o ciente das necessidades e expectativas que estão sendo atendidas, descrevendo por que o projeto está sendo realizado e proporcionando uma visão do estado final desejado. (SCHAWBER, 2004)

Product Owner, que é responsável pelo conteúdo, priorização e disponibilidade do *Product Backlog*. (SCHAWBER, 2004)

Product Backlog, que é uma descrição dos requisitos funcionais e não funcionais que o sistema deverá reunir para proporcionar o documento de visão, lembrando que o *Product Backlog* nunca está finalizado. (SCHAWBER, 2004)

- Compatibilidade:
 - O Scrum pode satisfazer estes requisitos.

GRE2: Obter o comprometimento da equipe técnica com os requisitos aprovados.

Os requisitos são aprovados pelo *Product Owner*, para constituir o *Product Backlog*, porem a equipe não utiliza critérios objetivos para a escolha de requisitos. (SCHAWBER, 2004)

- Compatibilidade:
 - O Scrum pode satisfazer este requisito.

GRE3: Estabelecer e manter a rastreabilidade bidirecional entre os requisitos e os produtos de trabalho.

O Scrum não tem nenhum recurso que se responsabiliza em estabelecer e manter a rastreabilidade dos requisitos.

- Compatibilidade:
 - O Scrum não pode satisfazer estes requisitos.

GRE4: Realizar revisões em planos e produtos de trabalho do projeto, visando identificar e corrigir inconsistências em relação aos requisitos.

Sprint Review, que é feito ao final de cada *Sprint*, apresentando o que foi desenvolvido ao *Product Owner* e outros interessados. (SCHAWBER, 2004)

Reunião Diária, que é feito todo dia, para manter uma comunicação com cada membro da equipe, sobre os progressos de cada requisito. (SCHAWBER, 2004)

Product Burndown e *Sprint Burndown*, que utilizam gráficos para ajudar no entendimento do andamento. (SCHAWBER, 2004).

- Compatibilidade:
 - O Scrum pode satisfazer estes requisitos.

GRE5: Gerenciar ao longo do projeto as mudanças nos requisitos.

Product Backlog, que tem por característica nunca estar terminado, ou seja, conforme surge novos requisitos ou alterações são lançados como uma *Sprint*. (SCHAWBER, 2004)

Sprint, que não são alterados durante o *Sprint*, porem caso aja a necessidade de mudança no *Sprint*, ele é interrompido e revisto em uma nova reunião de *Planning Meeting*, afim de adequar as novas necessidades, sendo registrado na nova versão do *Product Backlog* e *Spring Backlog*. Conforme Cohn(1998, p.5).

- Compatibilidade:
 - O Scrum pode satisfazer este requisito.

4.3 COMPATIBILIDADE DO MAPEAMENTO SEGUNDO AS EXTENSÕES PROPOSTAS.

O modelo MPS.BR foi constituído seguindo as normas NBR ISO/IEC 12207, ISO/IEC 15504 e o CMMI. (KOSCIANSKI, 2006, p.142)

Segundo Schwaber, com o *Scrum Methodology, Certified Scrum Program* (Programa de certificação em Scrum) e do Projeto *Quickstart*, o Scrum é compatível com o CMM (predecessor do CMMI) nível 3. (SCHWABER, 2004)

Também foi notado que existe uma grande aderência entre os ciclos de vida, aonde a Iniciação do MPS.BR nível G compatibiliza com a Visão do Scrum, do Planejamento com *Product Backlog, Sprint Plan #1 e Sprint Plan#2*, da Execução com o Desenvolvimento, do Monitoramento com a Reunião Diária e Cliente *Review* e do Encerramento com a Retrospectiva. (SIQUEIRA, 2007, p.46)

4.3.1 Resultados observados nesta monografia

A tabela 3 mostra o resultado obtido neste estudo com as comparações entre o que pede a gerência de projetos do modelo MPS.BR Nível G com o que o Scrum oferece.

Os itens da tabela abaixo relacionam as conclusões parciais da compatibilidade entre os GPR's do MPS.BR com o SCRUM nível G encontrados neste estudo.

Tabela 3 - Compatibilidade entre GPR do MPS.BR e Scrum

Fonte: Autoria própria.

GPR1	Satisfeito
GPR2	Não Satisfeito
GPR3	Satisfeito
GPR4	Parcialmente Satisfeito
GPR5	Não Satisfeito
GPR6	Parcialmente Satisfeito
GPR7	Satisfeito
GPR8	Satisfeito
GPR9	Não Satisfeito
GPR10	Satisfeito
GPR11	Satisfeito
GPR12	Satisfeito
GPR13	Satisfeito

GPR14	Satisfeito
GPR15	Satisfeito
GPR16	Satisfeito
GPR17	Parcialmente Satisfeito

Assim obtendo o resultado de 64,70% de compatibilidade entre os requisitos de gerencia de projetos do modelo MPS.BR nível G com Scrum, tendo também 17,65% de requisitos parcialmente satisfeitos e 17,65% de requisitos não satisfeitos.

A tabela 4 apresenta o que a gerência de requisitos do modelo MPS.BR Nível G pede com o que o Scrum oferece:

Tabela 4 - Compatibilidade entre GRE do MPS.BR com Scrum

Fonte: Autoria própria.

GRE1	Satisfeito
GRE2	Satisfeito
GRE3	Não Satisfeito
GRE4	Satisfeito
GRE5	Satisfeito

Assim obtendo o resultado de 80% de compatibilidade entre os requisitos de gerência de requisitos do modelo MPS.BR nível G com Scrum de forma satisfatória e 20% de requisitos não satisfeitos.

4.4 ESTUDO DE CASO DA EMPRESA POWERLOGIC COM A APLICAÇÃO E INTEGRAÇÃO DO SCRUM COM CMMI/MPS.BR NÍVEL G.

A PowerLogic é uma empresa de consultoria e desenvolvimento de softwares, fundada em 1994, sediada em Belo Horizonte MG, com a missão de surpreender seus clientes. Atualmente ela tem em média 500 aplicações corporativas em produção, sendo elas desde suítes de aplicativos para o desenvolvimento de softwares até o gerenciamento do ciclo de vida de uma aplicação.

Fabiano de Souza Andrade, do suporte técnico da PowerLogic, afirmou por telefone as seguintes informações:

Primeira Empresa a utilizar o Scrum para alcançar o MPS.BR Nível G;

Recentemente certificada MPS.BR Nível C utilizando Scrum;

Utilizam somente o Scrum como metodologia de desenvolvimento de software;

Perdem 1 dia organizando e executando o Scrum, mas afirmam que vale a pena.

4.4.1 Histórico da Empresa

O histórico da empresa PowerLogic é marcado pelos seguintes eventos:

- O ano 1994, foi marcado pela fundação da empresa PowerLogic em Minas Gerais;
- O ano 1995-1998, foi marcado pela utilização da arquitetura Cliente/Servidor e Downsizing da empresa;
- O ano 1998, foi marcado pelo início de Atuação com Java AppServers na empresa;
- O ano 1999, foi marcado pela conclusão do aplicativo eCompanyPortal/Project 1.0;
- O ano 2000-2001, foi marcado pela maturidade técnica em eBusiness;
- O ano 2002, foi marcado pelo início de desenvolvimento em J2EE Open-Source;
- O ano 2003, foi marcado pela conclusão do aplicativo jCompanyDeveloper1.0 tendo foco em Produtos;
- O ano 2004, foi marcado pela abertura da empresa no mercado acionário de capitais, PowerlogicSA (BNDES Pró-Soft);
- O ano 2004-2006, foi marcado pela atuação a nível Nacional, marcado pelo forte crescimento;
- O ano 2007 (Junho), foi marcado pela obtenção da certificação MPS.Br Nivel F;

- O ano 2007 (Dezembro), foi marcado pela conclusão do aplicativo PowerlogicALM;
- O ano 2010, foi marcado pela obtenção da certificação MPS.BR Nível C com Metodologia Scrum;
- O ano 2010, foi marcado pelo aporte BNDES de 2.4 Milhões de Reais. (Alvim, 2007)

4.4.2 Evolução dos Processos na empresa PowerLogic

A evolução dos processos na empresa PowerLogic é marcado pelos seguintes acontecimentos:

- O ano 1988-1993, foi marcado pelo quadro diretor com expertise em MDS e ferramentas CASE (Projeto de Ferramentas CASE, OO, Análise Essencial, Engenharia da Informação);
- Os anos 1994-2001, foram marcados pelo início de Uso do Processo Unificado e MDS diversas em Projetos de Clientes;
- O ano 2002, foi marcado pelo uso experimental de SCRUM e técnicas ágeis durante a formação da área de Produtos da empresa Powerlogic;
- O ano 2003, foi marcado pela apresentação da Palestra “Gestão Ágil de Projetos –SCRUM na prática”, no congresso “Extremme Programming Brasil”;
- O ano 2004, foi marcado pelo suporte ao SCRUM pelo eCompanyProcess, foi marcada também pela expansão do uso de SCRUM na empresa;
- O ano 2005, foi marcado pelo processo empírico estabelecido, com a utilização de Disciplinas PMBOK complementares. jCompanyQA suportando Integração Contínua. Automação e Gerência de Configuração;
- O ano 2006, foi marcado pela formalização e expansão do processo, segundo MPS.BR;
- O ano 2007 (Junho), foi marcado pela obtenção da certificação MPS.BR Nível F;

- O ano 2007 (Dezembro), foi marcado pelo Início Evolução para MPS.BR Nivel C;
- O ano 2010, foi marcado pela obtenção da certificação MPS.BR Nivel C com Metodologia Scrum. (Alvim, 2007)

4.4.3 Casos de Sucesso da empresa PowerLogic

Segundo Alvim(2007), destaca que atualmente são mais de 500 sistemas em produção na empresa, sendo elas:

Aplicações Internacionalizadas, por exemplo: Utilizadas na Alemanha, Kuala Lumpur, México, Bolívia, França.

Foco corporativo de Missão Crítica, por exemplo: Segurança, Disponibilidade de 24 horas por dia e 7 dias por semana, Performance e Escalabilidade.

Verticais Diversas, por exemplo: Indústria, Educação, Governo Executivo (Municipal, Estadual e Federal), Governo Judiciário, Financeira, Comércio, Cooperativa, Saúde, etc.

Incluindo também mais de uma centena de aplicações em tecnologias eBusinessJava EE Open-Source.

4.4.4 Desafios de compatibilidade Scrum & CMMI / MPS.BR encontrados na PowerLogic

4.4.4.1 Dificuldades referentes à gerência de Projetos

Como gerenciar com base em “planejamento contínuo” (planning) em lugar do “grande plano inicial”(big-bang plan)? (Alvim, 2007)

Como “abraçar” a mudança e “controlá-la” ao mesmo tempo?

4.4.4.2 Dificuldade Referente a Gerencia de Requisitos

Como estimar requisitos que são parcialmente conhecidos?

4.4.5 Estudo de caso na certificação da Powerlogic

4.4.5.1 Ações pertinentes nas Políticas Organizacionais Ágeis:

Aumento importante da imprevisibilidade implicando em um índice altíssimo da incorporação de tecnologia de ponta dos produtos Powerlogic.

Diferenciação com a concorrência, exigindo criatividade com base na construção em exploração e adaptação visando um planejamento constante e evolucionário, não antecipado e prescritivo.

Filosofias de entrega em “prazo fixo” visam exigências de liberação para o mercado (time-to-market) fazendo com que assim haja desenvolvimento iterativo com tempo definido para entrega.

Adaptação e monitoramento ágil a mudanças estratégicas gerando benefícios e com eles taxas notadamente alta.

Na área de produtos eles são definidos e priorizados em ultima instancia pela Powerlogic, mantendo de forma fundamental a presença do cliente em contato com a equipe.

4.4.5.2 Ações pertinentes nas Políticas Ágeis de Gerência de Projetos:

As Gerências de Projetos são praticas que tem por objetivo prever e garantir que se esteja trabalhando constantemente de maneira verdadeira e iterativa.

As iterações devem ter restrição principal de tempo, as iterações de desenvolvimento devem durar 30 dias e as subseqüentes de Garantia de Qualidade devem ter duração de 15 dias.

Devem ser defasadas em 15 dias as datas dos projetos e iterações entre as familias jCompany e eCompany.

Como primeira técnica do processo deve-se estimular e privilegiar o acompanhamento do andamento dos projetos fazendo com que a inspeção do software seja realizada pelo líder (*Scrum Master*) e entre os membros das equipes, utilizando programação em pares. Na segunda técnica as reuniões diárias de 15 minutos conhecidas como Daily Scrum, tem por objetivo garantir o *feedback face-a-face*(tácito), diário, inter-equipe. Deve-se prever acompanhamentos via indicadores indiretos de progresso (Apropriações, percentual de andamento) como terceira prioridade.

A política de “solicitação de mudança formal” leve realiza adaptações de fina granularidade durante um ciclo, sem entraves formais neste nível (micro-gestão), este processo deve prover espaço para o *Scrum Master* e para o *Scrum Team*. Deve-se garantir assim o rastreamento de todas as modificações ocorridas após o “compromisso” (*commitment*), e formalização em situações críticas (ameaça a objetivos fundamentais, alteração de versão de item de configuração, etc.).

Deve ocorrer reunião formal de reflexão das equipes (*Retrospective Meeting*) com frequência durante uma ou duas vezes por iteração para avaliar “o que foi bem” e “o que pode ser melhorado”.

O trabalho deve ser realizado pelos colaboradores em regime de 40 horas semanas, sem horas extras e com custo fixo.

4.4.5.3 Ações pertinentes nas Políticas Ágeis de Gerência de Requisitos

As praticas de gerencias tem por objetivo prover e garantir as técnicas de elicitação de requisitos evolucionárias visando em um planejamento contínuo, buscando evitar um longo plano inicial, preconizado por práticas ágeis. A partir dessas praticas deve-se garantir uma previsão razoável de escopo e prazo para um “plano de projeto” inicial, como consenso entre o representante do cliente (*Product Owner*) e a equipe (*Scrum Team*).

Os requisitos devem ser descritos em linguagens do usuário (bom português), seguindo os preceitos de Estórias de Usuários (*User Stories*) do XP e também do *Product Backlog* do Scrum. Deve ter priorização destas listas por “valor para o negocio” (*business value*) e pelo *Product Owner*.

O *Product Owner* e o *Scrum Team* são requisitos que devem ser refinados em reuniões de planejamento coletivas. Apenas o *Scrum Team* é habilitado para fazer previsões de tamanho, ele deve seguir técnicas ágeis e unidade em *Ideal Days*. A lista de ordenação de requisitos pode sofrer ajustes pelo *Scrum Team*, buscando refletir dependências técnicas e ajustes de custo/benefício a partir do momento em que o “tamanho” do requisito for definido. Tem por objetivo de a priorização maximizar o “business value” que é liberado em cada iteração.

O processo da consolidação de interesses de clientes deve prover facilidades de coleta da lista de requisitos do mercado através de interesses estratégicos da Powerlogic e prospects, que podem receber contribuições de qualquer *stack holder*.

Deve ocorrer o solucionamento de dúvidas microscópica, e tomadas rápidas de decisão para haver sucesso do processo do planejamento contínuo e para a agilidade em remoção de impedimentos, durante o processo também se deve considerar a forte presença do representante do cliente (*Product Owner*).

4.4.6 Melhorias Percebidas

Planejamento da disponibilidade dos recursos levando em consideração impedimentos e horas de retrabalho já identificados, horas gastas em reuniões, atendimentos externos, férias, etc, foi executado. Dessa maneira, foi garantida a participação real de cada membro envolvido;

Alinhamento de metas e planejamento. O Release e Sprint Goals são definidos com o consenso de todos os envolvidos promovendo comprometimento e visibilidade;

Definição de indicadores do Scrum Team e individuais estimularam o alcance de melhores resultados;

Gestão a vista via Agile Radiator, provendo feedback real e imediato e reuniões de inspeção contínua;

A integração contínua trouxe resultados importantes e informações essenciais para o planejamento e acompanhamento do projeto. O commit do código fonte, associado ao requisito que deu origem, propicia rastreabilidade em ambos os lados. Uma matriz de rastreabilidade foi construída para se obter estas informações facilitando a análise de impacto;

Integração de equipe: conceito de pilha entre requisitos estimulando a troca de conhecimento, uma vez que o requisito não possui “dono”;

Reunião de Retrospective Meeting, onde ocorre a coleta das lições aprendidas do projeto. Avalia-se o que deu certo (What went well – WWW) e o que deu errado (What can be improved) alimentando o projeto e também o processo;

Gerência de Qualidade de Processo: esta nova área criou uma zona de desconforto “sadia”, fazendo com que as pessoas dêem o melhor de si e concretize ações para o objetivo maior organizacional;

Gerência de Configuração: esta área garante a integridade dos itens de configuração do Release, após a geração de baselines e integração contínua;

Gerência de Qualidade de Produto: esta área provê suporte à qualidade dos produtos com testes automatizados, manuais e de integração.

4.4.7 Conclusão

Importância da institucionalização do processo para a obtenção de qualidade;

Comprometimento da Alta Direção na implementação do processo.

5 CONSIDERAÇÕES FINAIS

5.1 CONCLUSÃO

Fazendo as devidas comparações aqui apresentadas, percebe-se que há uma viabilidade de uso entre o Modelo MPS.BR e a metodologia Scrum, tendo em vista que encontrou-se uma empresa que conseguiu a certificação MPS.BR Nível G utilizando somente o Scrum e que atualmente já está no nível C e continua seguindo a metodologia Scrum. Também existe uma forte possibilidade que esta união possa proporcionar uma maior agilidade e resposta para certas mudanças ao longo do projeto.

Desta forma, percebe-se que a utilização de Scrum pode ser uma boa opção para micro e pequenas empresas se tornarem mais competitivas e obterem a certificação MPS.BR, a qual tem sido cada vez mais exigida pelo mercado.

5.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO

A combinação entre a metodologia Scrum com o modelo MPS.BR se mostrou satisfatória e viável no nível G, o mais baixo. Para continuação desse trabalho, deverá ser analisado níveis mais altos do modelo MPS.BR, buscando a suas compatibilidades e diferenças no Scrum, podendo também haver pesquisas com empresas que passaram por esse caminho, colhendo os dados de pós e contras dessa combinação.

Pode-se também fazer um trabalho de aplicação das técnicas do Scrum no dia a dia de uma empresa, seguindo as normas do MPS.BR nível G para assim levantar as dificuldades e facilidades em cada etapa, objetivando a Certificação.

REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO - SOFTEX, **GUIA GERAL DO MPS.BR**. 2011, Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_Geral_2011.pdf>. Acesso em: 20 de outubro de 2011.

ALVIM, P. **Aplicação e Integração do SCRUM e CMMI/MPS.BR, A experiência prática da Powerlogic.**, São Paulo: 2008, Disponível em:<http://www.spinsp.org.br/apresentacao/power_logic.pdf>. Acesso em: 25 de agosto de 2011.

COHN M. **Scrum**, Disponível em: <<http://www.mountangoatsoftware.com/scrum>>. Acesso em: 03 março de 2011.

FONSECA, I. **Utilizando Metodologias Ágeis para atingir MPS.BR nível F na Powerlogic.**, Disponível em: <http://www.asrconsultoria.com.br/downloads/pdf/Conf_0412_06_Powerlogic.pdf>. Acesso em: 25 de agosto de 2011.

LARMAN C. **Agile & Iterative Development, A Manager's Guide**, 2ª Ed. Addison-Wesley, ISBN 978-0131111554, 2004.

YOSHIMA, R. **ASPER COM**. Disponível em: <<http://www.aspercom.com/ead/mod/resource/view.php?id=245>>. Acesso em: 04 março de 2011, 2007.

SANTANA, C. A.; TIMÓTEO, A. L.; VASCONCELOS, A. M. L.. Mapeamento do modelo de Melhoria do Processo de Software Brasileiro (MPS.BR) para empresas que utilizam Extreme Programming (XP) como metodologia de desenvolvimento. Recife: 2006, Disponível em: <bibliotecadigital.sbc.org.br/download.php?paper=973>. Acesso em: 20 de outubro de 2010.

SCHWABER, K. **GUIA DO SCRUM**, Disponível em: <www.scrumalliance.org>. Acesso em: 20 de outubro de 2009.

SCHWABER, K. **Agile Project Management with Scrum.**, 1a Ed. Microsoft Press, ISBN:073561993x, 2004.

SIQUEIRA, H. B. A.. **Mapeamento Das Práticas De Scrum Nas Áreas De Processo Do Cmmi E Uma Proposta Para Sua Aderência.** Monografia (Ciências da Computação), Disponível em: <www.cin.ufpe.br/~tg/2007-1/hbas.pdf>. Acesso em: 01 de julho de 2010.

TRAVASSOS, G.H., KALINOWSKI, M. **iMPS: Resultados de Desempenho de Organizações que adotaram o MPS.** 1ª Ed. Campinas: Associação para Promoção da Excelência do Software Brasileiro - SOFTEX, ISBN 978-85-99334-11-9, 2008.

KOSCIANSKI, A. **Qualidade de Software.** 2ª Ed. Novatec, ISBN 978-85-7522-112-9, 2006.

KONTONYA, G. SOMMERVILLE, I. **Requirements Engineering - Processes and Techniques.** 1ª Ed. Wiley, ISBN 978-04-7197-208-2, 1996.