

UNIVERSIDADE DO EXTREMO SUL CATARINENSE

CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANDERSON RICARDO DOS SANTOS RODRIGUES

**MAPEAMENTO DOS REQUISITOS DA GERÊNCIA DE PROJETOS DO NÍVEL E
DO MPS.BR APLICANDO AS PRÁTICAS DO SCRUM**

CRICIÚMA

2012

ANDERSON RICARDO DOS SANTOS RODRIGUES

**MAPEAMENTO DOS REQUISITOS DA GERÊNCIA DE PROJETOS DO NÍVEL E
DO MPS.BR APLICANDO AS PRÁTICAS DO SCRUM**

Trabalho de Conclusão de Curso apresentado para obtenção do Grau de Bacharel no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense.

Orientadora: Prof^a MSc. Ana Cláudia Garcia Barbosa.

CRICIÚMA

2012

ANDERSON RICARDO DOS SANTOS RODRIGUES

**MAPEAMENTO DOS REQUISITOS DA GERÊNCIA DE PROJETOS DO NÍVEL E
DO MPS.BR APLICANDO AS PRÁTICAS DO SCRUM**

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Engenharia de Software.

Criciúma, 26 de Junho de 2012.

BANCA EXAMINADORA


Profª. Ana Cláudia Garcia Barbosa - Mestre - (UNESC) - Orientador


Prof. Gustavo Bisognin - Mestre - (UNESC)


Prof. Luciano Antunes - Mestre - (UNESC)

**Agradeço aos meus pais Ricardo e Maria
Margarete e a todos que me motivaram para o
desenvolvimento deste trabalho.**

AGRADECIMENTOS

Primeiramente agradeço a Deus por ter me dado a vida e as ferramentas necessárias para que eu pudesse realizar esta pesquisa.

Agradeço também a meus pais Ricardo e Maria Margarete, pois sem o suporte dos mesmos eu não chegaria até aqui. Toda a educação e o aprendizado que obtive com ambos influenciaram muito em meu caráter e na pessoa em que me tornei.

Tenho muito a agradecer também à minha orientadora Ana Cláudia que me auxiliou e me orientou na elaboração deste trabalho.

Agradeço à minha namorada por ter sido paciente e me apoiar nos momentos difíceis durante a realização desta pesquisa e a todas as pessoas que também de alguma forma me acompanharam e me influenciaram nesta jornada.

RESUMO

As empresas de software buscam nos dias atuais uma constante melhoria na produtividade e qualidade no processo de desenvolvimento de seus produtos a fim de obter maior competitividade no mercado e proporcionar uma melhor satisfação de seus clientes. Fornecer um produto com qualidade e entrega dentro do prazo definido com o cliente continua sendo um problema para as organizações. Modelos de qualidade e metodologias ágeis de desenvolvimento de software podem ser utilizadas pelas empresas como solução para tal problema. Enquanto os modelos de qualidade definem requisitos e normas para o processo, as metodologias ágeis de desenvolvimento definem valores e práticas que contribuem para o desenvolvimento de um produto de qualidade. Em virtude da importância dos modelos de qualidade e metodologias ágeis de desenvolvimento foi realizada esta pesquisa com o objetivo de verificar o relacionamento entre ambas. Neste caso foi realizado mapeamento ou avaliação dos requisitos da Gerência de Projetos do nível E do modelo de qualidade Melhoria do Processo de Software Brasileiro aplicando-se as práticas da metodologia ágil de desenvolvimento Scrum. Para que fosse possível realizar o mapeamento foram definidos critérios de avaliação e posteriormente foram descritos e avaliados individualmente todos os requisitos da Gerência de projetos do nível E aplicando-se as práticas do Scrum. Ao final das avaliações é apresentada a análise das mesmas e a conclusão definindo se com a metodologia ágil Scrum é possível atender os requisitos da Gerência de Projetos do nível E do modelo de qualidade Melhoria do Processo de Software Brasileiro.

Palavras chaves: Engenharia de Software; Modelo de Qualidade; Metodologia Ágil de Desenvolvimento de Software.

ABSTRACT

Today, software companies seek for constant improvement in productivity and quality in the process of developing their products to achieve greater market competitiveness and provide better customer satisfaction. Providing a quality product and timely delivery defined with the customer remains a problem for organizations. Quality models and agile software development can be used by companies as a solution to the problem. While quality models define requirements and standards for the process, the agile development methodologies define values and practices that contribute to the development of a quality product. Given the importance of quality models and agile development methodologies this research was conducted in order to verify the relationship between them. In this case, mapping was performed or evaluation requirements of the Project Management level E of the Brazilian Software Process Improvement quality model applying the practices of Agile Scrum development methodology. To make it possible to perform the mapping, evaluation criteria were defined and posteriorly described and individually evaluated all the requirements of the Project Management level E applying the practices of Scrum. At the end of the evaluation, the analysis and conclusion are presented to see if with the Scrum agile methodology it is possible to meet the requirements of the Project Management level E of the Brazilian Software Process Improvement quality model.

Keywords: Software Engineering, Quality Model, Agile Methodology of Software Development.

LISTA DE ILUSTRAÇÕES

Figura 1 - Níveis do CMMI.....	16
Figura 2 - Documentos do modelo de qualidade MPS BR.....	17
Figura 3 - Estrutura dos conjuntos de processos de cada nível do MPS.....	18
Figura 4 - Fluxo de fases do MA-MPS.....	21
Figura 5 - Sub-processos do processo de aquisição.....	22
Figura 6 - Processo do Scrum.....	28
Figura 7 - Funcionamento do Scrum.....	32
Figura 8 - Resultado da análise do mapeamento dos requisitos do nível G.....	41
Figura 9 - Resultado da análise do mapeamento dos requisitos do nível E.....	41
Figura 10 - Cruzamento dos resultados das análises níveis G e E do MPS.BR.....	42

LISTA DE TABELAS

Tabela 1 - Níveis de maturidade do modelo MPS.....	19
Tabela 2 - Critérios para avaliação dos requisitos da GPR.....	48

LISTA DE ABREVIATURAS E SIGLAS

AP	Atributo de Processo
ARC	Appraisal Requirements for CMMI
CMM	Capability Maturity Model for Software
CMMI	Capability Maturity Model Integration
EIA/IS	Eletronic Industries Alliance Ínterim Standard
GRE	Gerência dos Requisitos
GPR	Gerência de Projetos
IPD-CMM	Integrated Product Development Capability Maturity Model
MARES	Metodologia de Avaliação de Processos de Software
MA-MPS	Método de Avaliação para Melhoria do Processo de Software
MNC	Modelo de Negócio Cooperado
MNE	Modelo de Negócio Específico
MN-MPS	Modelo de negócio para Melhoria do Processo de Software
MPS	Melhoria de Processo de Software
MPS.BR	Melhoria do Processo de Software Brasileiro
MR-MPS	Modelo de Referência para Melhoria do Processo de Software
RAP	Resultado de Atributo de Processo
S&SC	Software e Serviços Correlatos
SCAMPI	Standard CMMI Appraisal Method for Process Improvement
SECM	System Engineering Capability Model
SOFTEX	Associação para Promoção da Excelência do Software Brasileiro
SW-CMM	Capability Model for Software
EAP	Estrutura Analítica do Projeto
WBS	Work Breakdown Structure

SUMÁRIO

1	INTRODUÇÃO	8
1.1	OBJETIVO GERAL.....	9
1.2	OBJETIVOS ESPECÍFICOS.....	9
1.3	JUSTIFICATIVA.....	9
1.4	ESTRUTURA DO TRABALHO	10
2	ENGENHARIA DE SOFTWARE	11
2.1	CONCEITO	11
2.2	FASES DE DESENVOLVIMENTO DE SOFTWARE.....	12
2.2.1	Fase de Definição	12
2.2.2	Fase de Desenvolvimento	13
2.2.3	Fase de Manutenção	13
3	PROJETO MPS.BR	15
3.1	CMMI.....	15
3.1.1	Conceito	15
3.2	MPS.....	17
3.3	MPS.BR.....	22
3.4	NÍVEL G	23
3.4.1	Gerência de Projetos (GPR)	24
3.5	NÍVEL E.....	25
4	MÉTODO ÁGIL DE DESENVOLVIMENTO SCRUM	27
4.1	CONCEITO DE MÉTODOS ÁGEIS.....	27
4.2	SCRUM	28
4.2.1	Product Backlog	29
4.2.2	Sprint	29
4.2.3	Fases	30
4.2.3.1	Pré-Game	30
4.2.3.2	Game	31
4.2.3.3	Pós-Game.....	31
4.2.4	Funcionamento	31
5	TRABALHOS CORRELATOS	33
5.1	PROPOSTA DE UM MODELO DE REFERÊNCIA PARA DESENVOLVIMENTO DE SOFTWARE COM FOCO NA CERTIFICAÇÃO DO MPS.BR	33

5.2	UMA SOLUÇÃO SISTÊMICA PARA A GERÊNCIA DE ATIVOS NO CONTEXTO DA IMPLEMENTAÇÃO DOS NÍVEIS DO MPS.BR.....	33
5.3	MAPEAMENTO DOS PROCESSOS DE DESENVOLVIMENTO ÁGEIS EM RELAÇÃO AO MODELO DE MELHORIA DO PROCESSO DE SOFTWARE DO BRASIL (NÍVEL G)	34
6	MAPEAMENTO DAS PRÁTICAS DO SCRUM EM RELAÇÃO AOS REQUISITOS REFERENTE À GERÊNCIA DE PROJETOS DO NÍVEL E DO MPS.BR.....	35
6.1	CRITÉRIOS DE AVALIAÇÃO DOS REQUISITOS	35
6.2	DEFINIÇÃO E AVALIAÇÃO DOS REQUISITOS DO NÍVEL E.....	35
6.2.1	Definição da GPR4.....	35
6.2.1.1	Avaliação da GPR4.....	36
6.2.2	Definição da GPR8.....	36
6.2.2.1	Avaliação da GPR8.....	37
6.2.3	Definição da GPR20.....	37
6.2.3.1	Avaliação da GPR20.....	38
6.2.4	Definição da GPR21.....	38
6.2.4.1	Avaliação da GPR21.....	39
6.2.5	Definição da GPR22.....	39
6.2.5.1	Avaliação da GPR22.....	40
6.3	ANÁLISE DAS AVALIAÇÕES	40
7	CONCLUSÃO	43
	REFERÊNCIAS.....	44
	APÊNDICE A - MAPEAMENTO DAS PRÁTICAS DO SCRUM EM RELAÇÃO AOS REQUISITOS REFERENTE À GERÊNCIA DE PROJETOS DO NÍVEL G DO MPS.BR.....	48
	APÊNDICE B - MAPEAMENTO DOS REQUISITOS DA GERÊNCIA DE PROJETOS DO NÍVEL E DO MPS.BR APLICANDO AS PRÁTICAS DO SCRUM	60

1 INTRODUÇÃO

Mesmo com a evolução das ferramentas e técnicas nos últimos anos, a produção de software confiável e correto entregue dentro do prazo estipulado ainda é muito difícil. Métodos ágeis para desenvolvimento de software são uma solução proposta para as metodologias de desenvolvimento de software tradicionais que são aplicados em situações em que os requisitos para a construção do sistema são estáveis e com evolução previsível. (SOARES, 2004).

Scrum é uma metodologia de desenvolvimento ágil cujo objetivo é fornecer um processo constituído por valores, práticas e técnicas para projeto e desenvolvimento software. A idéia principal do Scrum é que o desenvolvimento de softwares envolve muitos requisitos, recursos e tecnologia, que podem mudar durante o processo de desenvolvimento. Isto torna o processo de desenvolvimento imprevisível e complexo, requerendo flexibilidade para acompanhar as mudanças (SOARES, 2004).

As empresas têm como um de seus principais objetivos fornecer um produto com qualidade a seus clientes, a fim de obter competitividade no mercado. Para as empresas de software, alcançar esta competitividade pela qualidade implica tanto na melhoria da qualidade de seus produtos e serviços quanto nos métodos de produção e distribuição do software. A Melhoria de Processo de Software (MPS) é um modelo de melhoria a avaliação do processo de software destinado preferencialmente a pequenas e médias empresas. A Melhoria do Processo de Software Brasileiro (MPS.BR) é um projeto que visa a melhoria do processo de software do país e sua novidade está na estratégia de implementação, criada para a realidade brasileira. A base técnica utilizada para a construção deste modelo de qualidade foram as normas ISO/IEC 12207, ISO/IEC 15504 além do conteúdo de outros modelos de referência como Capability Maturity Model Integration (CMMI) (WEBER, 2005).

O modelo MPS possui três componentes: Modelo de referência para Melhoria do Processo de Software (MR-MPS), Método de avaliação para Melhoria do Processo de Software (MA-MPS) e Modelo de negócio para Melhoria do Processo de Software (MN-MPS). No MR-MPS estão definidos os sete níveis de maturidade representados por letras de A até G no qual A é o nível mais alto e G o mais baixo (SOFTEX, 2009). O objetivo do trabalho será a análise ou mapeamento dos requisitos descritos na Gerência de Projetos do nível E do MPS.BR aplicando a metodologia ágil Scrum.

1.1 OBJETIVO GERAL

Realizar o mapeamento dos requisitos do sub-processo Gerência de Projetos do nível E do MPS.BR aplicando as práticas da metodologia de desenvolvimento ágil Scrum afim de analisar se as práticas da metodologia ágil atendem aos resultados esperados por este sub-processo.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa são:

- a) compreender o método ágil de desenvolvimento Scrum;
- b) compreender o modelo MPS e o projeto MPS.BR;
- c) análise e mapeamento dos requisitos da GPR do nível E do MPS.BR aplicando as práticas do Scrum;

1.3 JUSTIFICATIVA

Um desafio constante da área de Engenharia de Software é melhorar o processo de desenvolvimento de software e mesmo com a constante evolução de métodos, técnicas e ferramentas, a entrega de software em prazos estabelecidos nem sempre é conseguida. Existe a necessidade de desenvolver software de forma mais rápida e produtiva, mas com qualidade e esse desenvolvimento pode ser obtido utilizando métodos ágeis (COSTA FILHO, 2005).

O Scrum é usado em trabalhos complexos onde não é possível prever tudo o que irá ocorrer e oferece um ambiente de trabalho e um conjunto de práticas que melhora a visibilidade dos processos no projeto. Isto permite aos praticantes analisar o que pode acontecer ao longo do projeto e fazer os devidos ajustes para manter o mesmo em andamento ao longo do tempo, visando alcançar seus objetivos (GIORDANI, 2010).

O nível G é o primeiro nível de maturidade do MR-MPS e ao final da implantação do mesmo a empresa deve ser capaz de gerenciar parcialmente seus projetos de desenvolvimento de software. Neste nível a empresa pode utilizar seus próprios padrões e procedimentos, não sendo necessário que se possua padrão em nível organizacional (SOFTEX, 2011). O nível E é o terceiro nível de maturidade do MR-MPS onde neste há a evolução dos requisitos e dos resultados esperados do sub-processo Gerência de Projetos em relação aos níveis G e F do MPS.BR. Definiu-se o nível E pois toda a evolução para um dos

níveis de maturidade implica em mudanças no processo de desenvolvimento de software dentro das empresas onde neste nível há mudanças consideráveis em relação aos níveis G e F. Com estas mudanças pretende-se analisar se as práticas do Scrum atendem aos requisitos do nível E do MPS.BR pois nos níveis G e F as práticas da metodologia ágil atendem grande parte dos requisitos.

1.4 ESTRUTURA DO TRABALHO

A presente pesquisa está estruturada em capítulos da seguinte maneira:

O capítulo 1 é a introdução, onde nela é descrita a definição do problema, objetivo geral, objetivos específicos e a justificativa deste trabalho.

No capítulo 2 é descrito o conceito de engenharia de software, além de descrever as fases de definição, desenvolvimento e manutenção.

O capítulo 3 descreve o modelo MPS e o CMMI que é a base de construção do modelo. Também neste capítulo são conceituados os níveis G e E do modelo MPS.BR, juntamente seus processos e requisitos que compõe o nível em questão.

O capítulo 4 define a metodologia de desenvolvimento ágil Scrum. No conteúdo deste capítulo também é descrito todos os componentes e o funcionamento da metodologia.

No capítulo 5 são descritos três trabalhos correlatos, no qual foram usados como fonte de pesquisa para a possibilidade de aplicação do modelo MPS.BR na metodologia ágil de desenvolvimento Scrum.

O capítulo 6 é referente ao mapeamento dos requisitos da Gerência de Projetos do nível E do MPS.BR. Após os mapeamentos há a análise destes juntamente com o cruzamento destas informações com análises obtidas do mapeamento do nível G do MPS.BR, a fim de verificar se o Scrum se adéqua aos requisitos do modelo de qualidade.

Após os mapeamentos há a conclusão do trabalho referente às análises dos mapeamentos do nível E e ao cruzamento destas informações com análises obtidas do mapeamento do nível G do MPS.BR.

Por fim há o apêndice com o mapeamento dos requisitos do nível G do MPS.BR e as referências bibliográficas utilizadas para a montagem e descrição deste projeto de pesquisa.

2 ENGENHARIA DE SOFTWARE

Esta é a parte inicial e introdutória dos conceitos necessários para que se possa compreender o objetivo deste trabalho. Neste capítulo será fornecido um conceito sobre engenharia de software, onde serão descritos o que são cada um de seus componentes e suas fases de desenvolvimento.

2.1 CONCEITO

A engenharia de software pode ser conceituada como uma metodologia de desenvolvimento e manutenção de software, com as seguintes características: roteiro dinâmico de soluções tecnológicas; adequação aos requisitos funcionais do cliente e seus respectivos processos; efetivação de padrões de qualidade; produtividade; fundamentação na tecnologia da Informação disponível, viável, oportuna e customizada; planejamento e gestão de atividades, recursos, custos e prazos (REZENDE, 2005).

A engenharia de software abrange um conjunto de três elementos – métodos, ferramentas e procedimentos – que possibilita ao gerente formas para a construção de um software de alta qualidade e produtividade.

Os métodos de engenharia de software descrevem formas de “como fazer” ou “como construir” um software, no qual envolvem um amplo conjunto de tarefas que incluem: planejamento e estimativa de projeto, análise de requisitos de software, projeto da estrutura de dados, algoritmo de processamento, codificação, teste e manutenção (PRESSMAN, 1995).

As ferramentas são instrumentos que proporcionam um apoio automatizado e detalhes de “como fazer” para se construir um software. Existem diversas técnicas para sustentar os métodos como por exemplo orientação a objetos e com estes, suas respectivas ferramentas como linguagens de programação.

Os procedimentos definem a seqüência que os métodos serão aplicados, os produtos a serem disponibilizados, controle de qualidade e avaliação de software. São estes que antecedem e sucedem o software (REZENDE, 2005).

Pode-se concluir então que engenharia de software é uma metodologia ou um roteiro composto de diversas técnicas que são aplicadas em suas diversas fases de construção de um sistema. A seguir serão conceituadas e definidas as fases de desenvolvimento de software.

2.2 FASES DE DESENVOLVIMENTO DE SOFTWARE

O processo de construção de um software possui três fases genéricas sendo elas definição, desenvolvimento e manutenção. As mesmas são definidas a seguir.

2.2.1 Fase de Definição

Esta fase focaliza o *o que*, ou seja, durante a definição o analista procura verificar quais informações deverão ser processadas e armazenadas pelo sistema, define a interface, quais são as restrições existentes no projeto e quais critérios de validação são exigidos para se definir um sistema bem-sucedido. Durante este processo, dependendo do padrão de definição aplicado, três etapas Análise do Sistema, Planejamento do Projeto de Software e Análise de Requisitos ocorrerão de alguma forma (PRESSMAN,1995).

A etapa Análise do Sistema é onde são transformadas suas duas principais entradas, a política do cliente e os encargos do projeto, em uma especificação estruturada do projeto. Nesta, é feita a modelagem do sistema com os Diagramas de Fluxo de Dados (DFD), Dicionário de Dados, bem como de outras ferramentas (REZENDE, 2005).

A etapa Planejamento do Projeto de Software é realizada assim que o escopo do projeto é estabelecido, onde os riscos são analisados, os recursos são alocados, os custos são estimados e as tarefas e programação de trabalho são definidas. Dentro de cada tarefa, a atividade do projeto é focada no desenvolvimento de uma hierarquia de módulos de ‘programas’ e interfaces entre estes módulos para implementar a especificação criada na fase anterior. Esta etapa pode ser também definida como Projeto Lógico (o que fazer) e início do Projeto Físico (como fazer) (PRESSMAN, 1995).

A etapa Análise de requisitos trata-se na identificação das regras e das necessidades dos usuários de informações e comunicação destas necessidades no processo de construção do software (SELNER, 1999). O escopo definido para o projeto proporciona uma direção onde um detalhamento e aprofundamento das informações e das funções e funcionalidades do sistema é necessária antes que o trabalho se inicie (REZENDE, 2005).

Conclui-se então que a fase de definição procura se aprofundar e planejar toda a infra-estrutura do sistema, realizando a análise do sistema e definindo os requisitos para que o mesmo atenda as necessidades dos usuários ou clientes.

2.2.2 Fase de Desenvolvimento

Nesta fase, o desenvolvedor focaliza-se em como a estrutura de dados e a arquitetura de software tem de ser projetados, como o projeto será traduzido numa linguagem de programação e como os testes têm de ser realizados. A fase de desenvolvimento ocorre em três passos: Projeto de software, Codificação e Realização de testes de software.

O projeto define os requisitos do software em um conjunto de representações podendo ser estas gráficas, tabulares ou baseadas em linguagem e que descrevem a estrutura de dados, a arquitetura, o procedimento algorítmico e as características de interface.

Na codificação as representações do projeto devem ser repassadas ou transcritas em uma linguagem de programação convencional ou não-procedimental que resulte em instruções que possam ser ‘compreendidas’ e executadas pelo computador (PRESSMAN, 1995).

A realização de testes de software é realizada logo após a implementação do software onde por meio desta tenta-se descobrir defeitos de função, lógica e implementação. Esta etapa consiste na elaboração de roteiros de teste, construção de rotinas automáticas de teste e na execução dos testes (REZENDE, 2005).

2.2.3 Fase de Manutenção

Um software sofre manutenções, sejam estas por simples correções, alteração legal ou por estar gerando problemas para o cliente. Segundo Rezende (2005, p. 48) “a manutenção de software existente demanda mais de 70% de todo o esforço despendido por uma organização (ou departamento) de software...”

As alterações nos sistemas são inevitáveis, e para isto deve-se desenvolver mecanismos e processos para avaliar, controlar e fazer modificações, afim de reduzir o esforço despendido em manutenção. Há vários tipos de manutenção de software, podendo ser:

- a) manutenção por legislação;
- b) manutenção por melhoria ou implementação;
- c) manutenção por correções de erros;
- d) manutenção corretiva;
- e) manutenção adaptativa;
- f) manutenção perfectiva;
- g) manutenção preventiva ou preditiva; e

h) manutenção estruturada.

Conforme descrito neste capítulo foi abordado o conceito de Engenharia de Software além de suas fases primitivas de desenvolvimento. No próximo capítulo será dada a introdução nos conceitos principais do modelo Melhoria do Processo de Software Brasileiro (MPS.BR) além de destacar os níveis G e E e seus requisitos no qual são os objetos de estudo deste trabalho.

3 PROJETO MPS.BR

Neste capítulo será conceituado o projeto MPS.BR, os níveis G e E e seus requisitos. Também serão descritas suas bases de construção que são os modelos MPS e CMMI.

3.1 CMMI

O Capability Maturity Model for Software (CMM) é um modelo para melhoria dos processos de software criado em 1991, porém para as empresas existia-se a dificuldade da aplicação conjunta de diferentes modelos.

O Projeto CMM Integration (CMMI) foi criado para integrar diferentes modelos (DONEGAN et al, 2005) onde o objetivo do CMMI Product Team era a combinação de três modelos originais: Capability Maturity Model for Software (SW-CMM); Eletronic Industries Alliance Ínterim Standard (EIA/IS); Integrated Product Development Capability Maturity Model (IPD-CMM) em um framework de melhoramento simplificado para empresas que perseguem ampla iniciativa de melhoria de processos (ATAÍDES, 2006).

Com a utilização de informações de outros modelos populares e respeitados como referência, o CMMI Product Team criou um grupo de modelos integrados no qual podem ser adotados por aqueles que utilizam outros CMMs quanto para os que são novos no conceito.

Além do desenvolvimento de um framework comum para suportar a integração de outras disciplinas específicas do modelo CMMI, a equipe procurou garantir que todos os produtos desenvolvidos fossem consistentes e compatíveis com o ISO/IEC 15504 (ATAÍDES, 2006).

3.1.1 Conceito

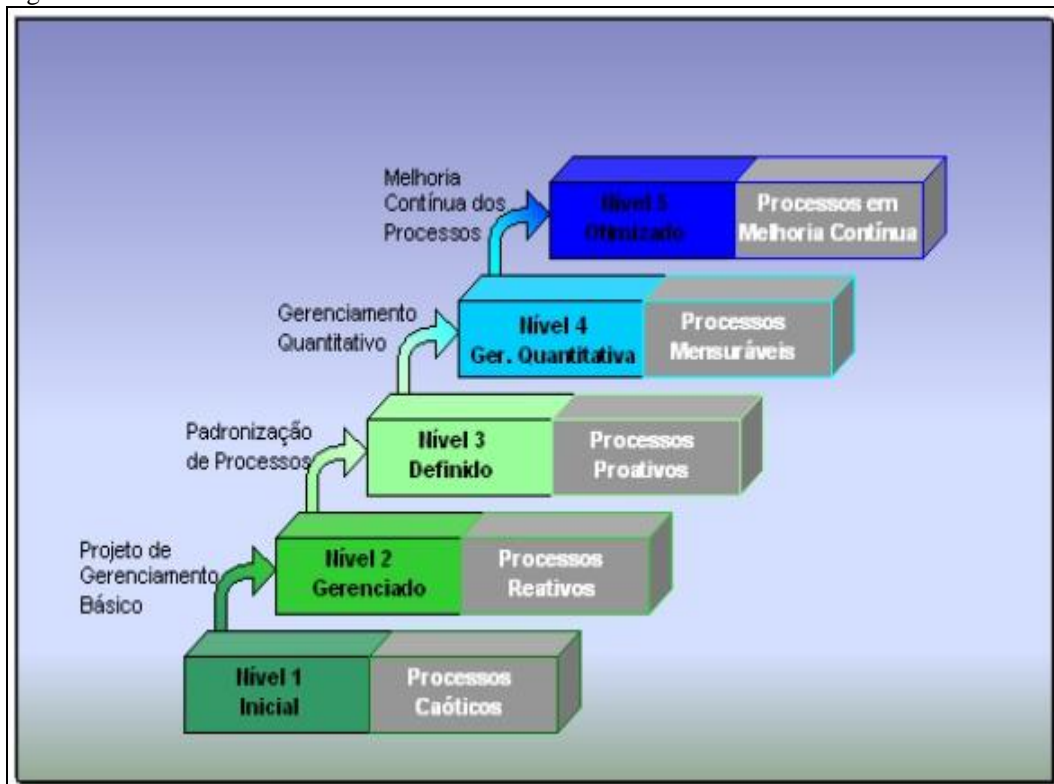
O Capability Maturity Model for Software Integrarion (CMMI) é um modelo para avaliação e melhoria da maturidade dos processos de software de uma organização e uma integração dos modelos: Capability Model for Software (SW-CMM), System Engineering Capability Model (SECM), e Integrated Product Development CMM (IPD-CMM) (ATAÍDES, 2006). O CMMI é um modelo que foi criado para integrar vários modelos criados a partir do sucesso alcançado pelo SW-CMM (DONAGAN; BANDEIRA; SAMPAIO, 2005).

Este modelo foi projetado para guiar as organizações no processo de melhoria, definindo a maturidade do atual processo e identificando os pontos críticos para a qualidade e melhoria do processo (ATAÍDES, 2006).

Um processo pode ser definido como um conjunto de atividades, métodos, práticas e mudanças que se pode usar para desenvolver e manter os produtos e/ou serviços. Estes processos ou práticas demonstram o quanto uma organização é madura para desenvolver seus produtos/serviços. Segundo o CMMI, existem três tipos de análises sobre os processos de software: Capacidade, Desempenho e Maturidade (ATAÍDES, 2006).

O modelo possui cinco níveis de maturação, onde cada nível fornece os fundamentos necessários para que se possa amadurecer e progredir para o próximo nível (ZANATTA; VILAIN, 2005) A figura 1 demonstra esta situação.

Figura 1 – Níveis do CMMI



Fonte: ATAÍDES, A. D. (2006, p. 22)

Estes níveis oferecem para uma organização um caminho para prever o desempenho futuro dentro Engenharia de Software.

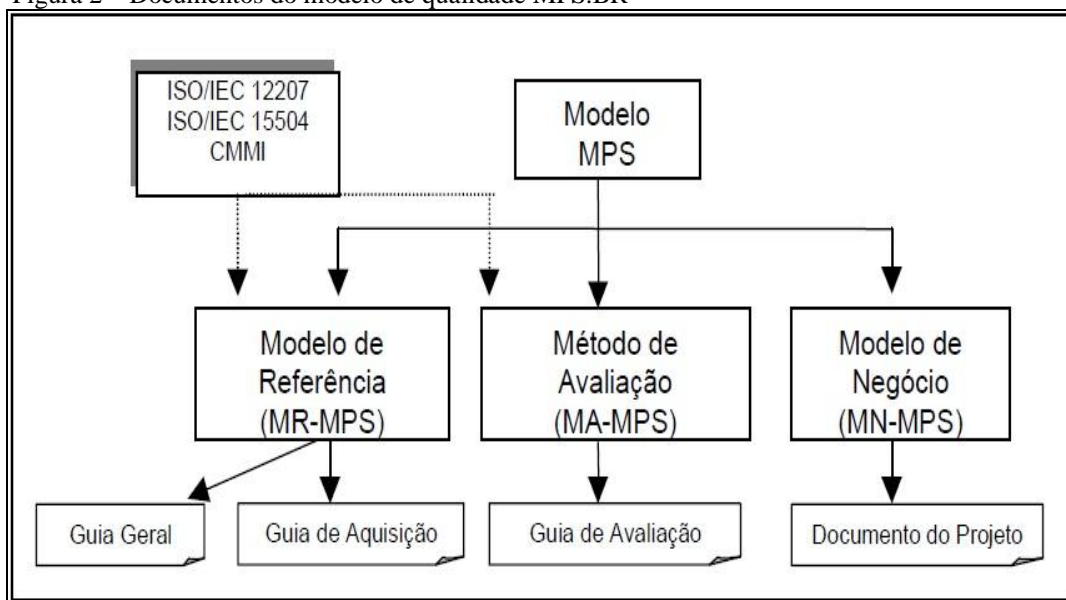
3.2 MPS

A Melhoria do Processo de Software (MPS) é um modelo de melhoria e avaliação de processo de software, voltado preferencialmente a pequenas e médias empresas, normalmente grupos de empresas em um Modelo de Negócio Cooperado (MNC), mas também vem sendo aplicado em organizações de grande porte no Modelo de Negócio Específico (MNE) (SOFTEX, 2009).

A base técnica utilizada para a construção do modelo MPS foram as normas ISO/IEC 12207 e ISO/IEC 15504 com as quais esta em conformidade. Para complementar, o modelo assimila o conteúdo de outros Modelos de Referência de Processo, como o CMMI (SOFTEX, 2009).

O modelo define também regras para sua implementação e avaliação. Como mostra a figura 2, o MPS possui três componentes: Modelo de Referência para Melhoria do Processo de Software (MR-MPS), Método de Avaliação para Melhoria de Processo de Software (MA-MPS) e Modelo de Negócio para Melhoria de Processo de Software (MN-MPS) (SOFTEX, 2009). Estes componentes foram descritos em documentos específicos.

Figura 2 – Documentos do modelo de qualidade MPS.BR

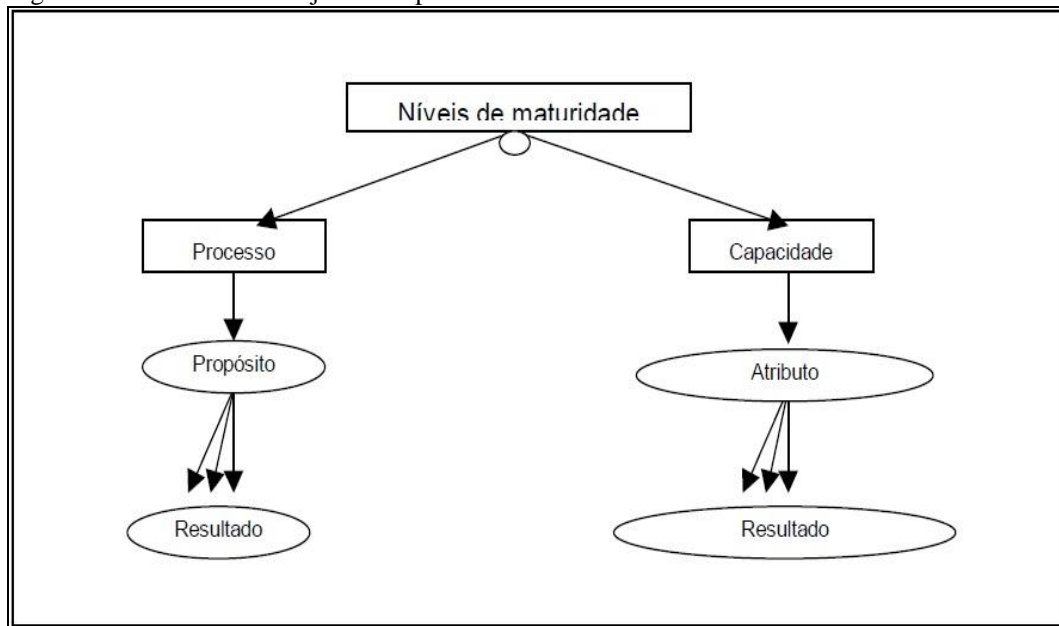


Fonte: SOFTEX (2009, p.4)

No Modelo de Referência para Melhoria de Processo de Software (MR-MPS) estão descritos os requisitos que as empresas precisam atender para estar em conformidade com o modelo MPS (SOFTEX, 2009). Este por sua vez é composto por sete níveis de maturidade que são seqüenciais e cumulativos. Cada um destes níveis é composto por um

conjunto de processos em um determinado nível de capacidade onde o progresso e o atendimento de cada nível de maturidade é obtido somente quando são atendidos todos os requisitos e atributos relacionados àquele nível (SANTANA, TIMÓTEO, VASCONCELOS, 2006). A figura 3 demonstra esta estrutura.

Figura 3 – Estrutura dos conjuntos de processos de cada nível do MPS



Fonte: SOFTEX (2009, p. 5)

Cada nível de maturidade estabelece patamares de evolução de processos. Os níveis são definidos em: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). Estes por sua vez são baseados nos quatro níveis de maturidade da representação por estágio do CMMI (níveis 2 a 5), onde os níveis F, C, B e A do modelo MPS correspondem respectivamente aos níveis 2, 3, 4 e 5 do CMMI; os níveis E e D são níveis intermediários entre os níveis 2 e 3 do CMMI e o nível G é um nível intermediário entre os níveis 1 e 2 do CMMI (SOFTEX, 2009).

A capacidade do processo de melhoria ou do nível de maturidade possui cinco Atributos de Processo (AP): AP 1.1 (o processo é executado); AP 2.1 (o processo é gerenciado); AP 2.2 (os produtos de trabalho do processo são gerenciados) e AP 3.1 (o processo é definido); AP 3.2 (o processo-padrão está implementado). Estes atributos são baseados nos atributos de processo ISO/IEC 15504 e correspondem às práticas genéricas do CMMI-SE/SW (SOFTEX, 2009). A tabela 1 contém a representação dos níveis de maturidade e seus respectivos atributos.

Tabela 1. Níveis de maturidade do modelo MPS

Nível	Nome e sigla dos Processos	Atributos de Processo (Capacidade)
A (mais alto)	Inovação e Implantação na Organização – IIO, Análise e Resolução de Causas - ARC	AP 1.1 AP 2.1, AP 2.2, AP 3.1 e AP 3.2
B	Desempenho do Processo Organizacional – DEP, Gerência Quantitativa do Projeto - GQP	AP 1.1 AP 2.1, AP 2.2, AP 3.1 e AP 3.2
C	Gerência de Riscos – GRI, Análise de Decisão e Resolução – ADR	AP 1.1 AP 2.1, AP 2.2, AP 3.1 e AP 3.2
D	Desenvolvimento de Requisitos – DRE, Solução Técnica – STE, Validação – VAL, Verificação – VER, Integração do Produto – ITP, Instalação do Produto – ISP, Liberação do Produto – LIP	AP 1.1 AP 2.1, AP 2.2, AP 3.1 e AP 3.2
E	Treinamento – TER, Definição o Processo Organizacional – DFP, Avaliação e Melhoria do Processo Organizacional – AMP, Adaptação do Processo para Gerência do Projeto – APG	AP 1.1 AP 2.1, AP 2.2, AP 3.1 e AP 3.2
F	Gerência de Configuração – GCO, Garantia da Qualidade – GQA, Medição – MED, Aquisição – AQU	AP 1.1 AP 2.1 e AP 2.2
G (mais baixo)	Gerência de Projeto – GPR, Gerência de Requisitos - GRE	AP 1.1 e AP 2.1

Fonte: SOFTEX (2009, p. 6)

O Modelo de Referência está adequado ao CMMI baseado nas normas ISSO/IEC 12207 e ISSO/IEC 15504.

O Método de Avaliação para Melhoria de Processo de Software (MA-MPS) contém a descrição do processo de avaliação, os requisitos para verificação dos avaliadores e os requisitos para conformidade com o MR-MPS. Este está descrito no Guia de Avaliação (SOFTEX, 2009).

No último dos três componentes, o Modelo de Negócio para Melhoria de Processo de Software (MN-MPS), possui a descrição das regras dos domínios: o domínio do Projeto MPS.BR, coordenado pela SOFTEX; o domínio das Instituições Implementadoras do modelo MPS e Instituições Avaliadoras do Modelo MPS; e o domínio das empresas que queiram utilizar o modelo MPS para melhorar seus processos de software. Maiores detalhes destas regras estão disponíveis como um documento do Projeto MPS.BR (SOFTEX, 2009).

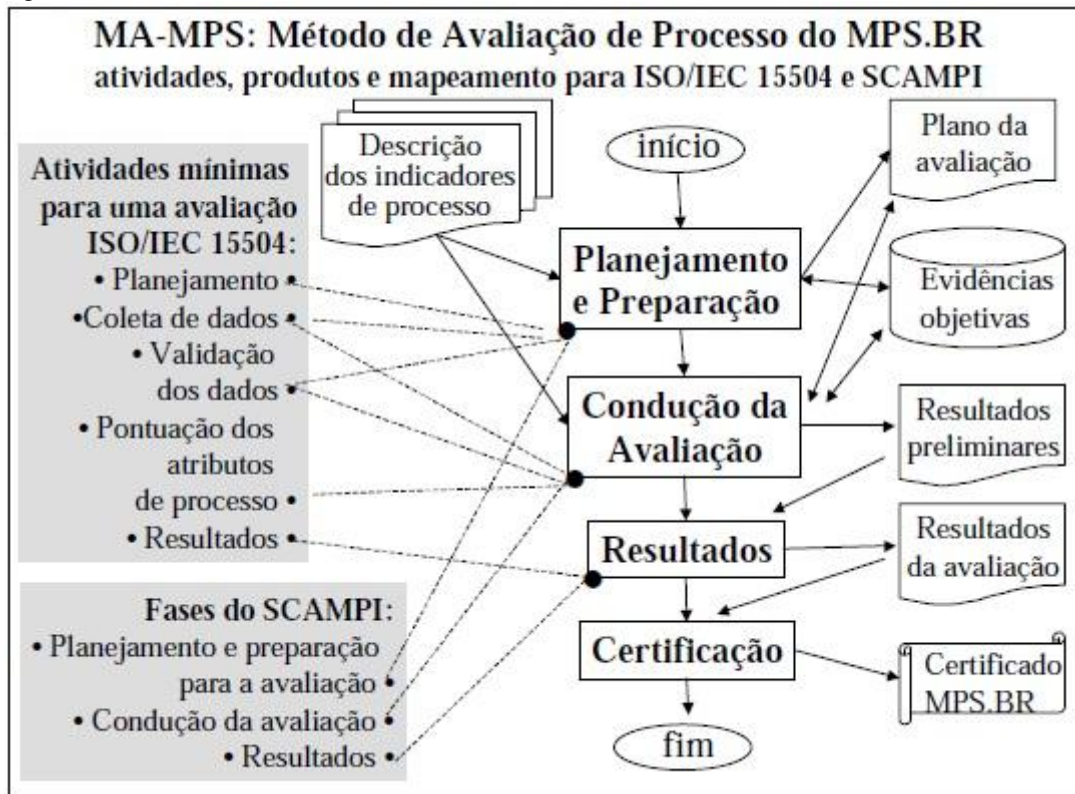
Cada modelo possui um guia contendo as descrições e regras específicas para cada modelo. Começando pelo documento do Guia Geral, conforme a figura 4 contém uma descrição geral do modelo MPS, um detalhamento do MR-MPS e os requisitos que as empresas devem atender para estar de acordo com o modelo MPS (SOFTEX, 2009).

O Guia de Avaliação contém a descrição do Método de Avaliação para Melhoria do Processo de Software (MA-MPS) e é composto pelos seus requisitos, atividades, indicadores para avaliação e características da qualificação dos avaliadores. Os requisitos do MA-MPS tem como base a norma ISO/IEC 15504-2, os requisitos definidos na Appraisal Requirements for CMMI (ARC) do CMMI e os requisitos específicos do Projeto MPS.BR. Assim, o método é compatível com a ISO/IEC 15504 e ARC do CMMI e adequado a pequenas e médias empresas (SOFTEX, 2009).

O método de avaliação descrito no Guia de Avaliação tem suas atividades baseadas no Standard CMMI Appraisal Method for Process Improvement (SCAMP) com elementos dos métodos QuickLocus e Metodologia de Avaliação de Processos de Software (MARES). O SCAMP é o método de avaliação padrão para gerar classificação comparável para as representações por estágio e contínua do CMMI. O QuickLocus é um outro método de avaliação voltado para empresas de software com até 50 pessoas e o MARES é um método para execução de avaliação de processos de software em pequenas empresas (SOFTEX, 2009).

São quatro as fases que compõem o processo de avaliação, sendo elas: Preparação e planejamento, Condução da avaliação, Resultados e certificação. A figura 4 demonstra o fluxo de fases, os principais produtos de entrada e saída e o mapeamento das mesmas nas atividades mínimas requeridas pela ISO/IEC 15504 e nas fases do SCAMP (SOFTEX, 2009).

Figura 4 – Fluxo de fases do MA-MPS



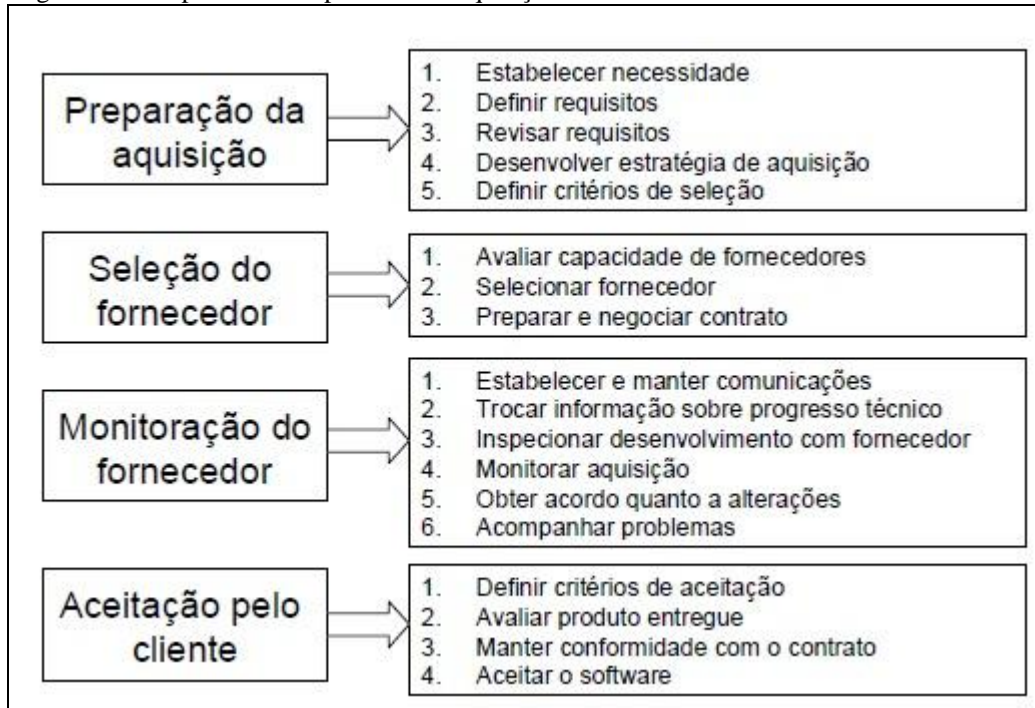
Fonte: SOFTEX (2009, p. 8)

O Guia de Aquisição descreve o Processo de Aquisição cujo objetivo é obter Software e Serviços Correlatos (S&SC) que satisfaça a necessidade do cliente. Este processo inicia com a identificação da real necessidade expressa pelo cliente em questão e finaliza com a aceitação do produto ou serviço (SOFTEX, 2009). Segundo SOFTEX (2009) são definidos como resultados de implementação bem sucedida do processo de aquisição:

- a) as necessidades de aquisição, as metas, os critérios de aceitação do S&SC e as estratégias de aquisição são definidos;
- b) um contrato que expresse claramente a expectativa, as responsabilidades e as obrigações de ambos (cliente e fornecedor) é desenvolvido;
- c) um S&SC que satisfaça a necessidade expressa do cliente é adquirido;
- d) a aquisição é monitorada de forma que as condições especificadas, tais como custo, cronograma e qualidade, são atendidas;
- e) os produtos entregues pelo fornecedor são aceitos; e
- f) qualquer pendência identificada tem uma conclusão satisfatória, conforme acordado entre o cliente e o fornecedor.

Conforme a figura 5, o processo está descrito através de quatro subprocessos: Preparação da aquisição; Seleção do fornecedor; Monitoração do fornecedor; e Aceitação pelo cliente.

Figura 5 – Sub-processos do processo de aquisição



Fonte: SOFTEX (2009, p. 10)

Além da definição e detalhamento do processo de aquisição e S&SC, o Guia descreve outros pontos relevantes como problemas comuns que ocorrem na aquisição do software, aquisição de software livre/código aberto, aquisição e a engenharia de software baseada em componentes e a evolução do nível de capacidade no processo de aquisição de S&SC (SOFTEX, 2009).

Concluindo, foi conceituado o modelo MPS e seus componentes que são base do Projeto MPS.BR para Melhoria do Processo de Software Brasileiro. A seguir será descrito o que é este projeto e quais seus objetivos.

3.3 MPS.BR

O MPS.BR é um projeto desenvolvido em 2004 pelas seguintes instituições-âncora, integrantes do Comitê Gestor MPS: SOFTEX (coordenadora do projeto), COPPE/UFRJ (coordenadora da ETM – Equipe técnica do modelo) e RIOSOFT no Rio de Janeiro/RJ, CenPRA e Agente SOFTEX em Campinas/SP, CESAR em Recife/PE e

CELEPAR em Curitiba/PR. Em 2005-2006, o projeto contou com apoio financeiro do MCT através de recursos do FNDCT/CT-INFO/VERDE-AMARELO/FINEP (SOFTEX, 2009).

Segundo SOFTEX (2009), o projeto tem como objetivo a melhoria do processo de software brasileiro, a um custo acessível, e possui duas metas:

- a) desenvolvimento e aprimoramento do modelo MPS, compatível com CMMI e em conformidade com as normas ISO/IEC 12207 e ISO/IEC 15504;
- b) a implementação e avaliação do modelo MPS, a um custo acessível em todas as regiões do país, com foco em grupos de pequenas e médias empresas.

Estudos mostram que para aumentar a maturidade e qualidade dos processos nas empresas brasileiras é necessário um esforço significativo e que, nos últimos anos, as empresas de software no Brasil favoreceram a ISO 9000, em virtude de outras normas e modelos voltadas para a melhoria de processos de software.

Segundo dados do Ministério da Ciência e Tecnologia em 2003, duzentas e quatorze empresas que desenvolviam software no Brasil possuíam certificação ISO 9000 e trinta com certificado SEI/CMU de avaliações CMM. Estas últimas compreendiam em 2003: vinte e quatro empresas no nível 2, cinco no nível 3, uma no nível 4 e nenhuma no nível 5 do CMM (SOFTEX, 2009).

Como solução do problema, em Dezembro de 2003, a Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) que é uma entidade privada sem fins lucrativos e que promove ações com abrangência nacional visando transformar o Brasil em um centro de excelência mundial na produção e exportação de software, propôs o projeto MPS.BR.

Não é objetivo do projeto definir novas normas e modelos, o mesmo concentra-se apenas em definir uma estratégia de implementação, criada e adaptada para a realidade brasileira (SOFTEX, 2009).

3.4 NÍVEL G

O nível G é o primeiro nível de maturidade do modelo MPS, onde este possui dois processos que são a Gerência dos Requisitos (GRE) e a Gerência de Projetos (GPR). A Gerência de Projetos possui dezessete práticas que devem ser cumpridas, enquanto a Gerência de Requisitos possui apenas sete práticas. Este possui apenas dois atributos de processos definidos como AP 1.1 (O processo é executado) e AP 2.1 (O processo é gerenciado). Todos estes itens pertencentes ao nível em questão serão descritos adiante (SANTANA; TIMÓTEO;

VASCONCELOS, 2006). Como o objetivo do trabalho é apenas com relação aos requisitos da GPR do MPS.BR, será descrito apenas este processo.

3.4.1 Gerência de Projetos (GPR)

A Gerência de Projetos, como já descrito anteriormente, é um dos processos pertencentes ao nível G de maturidade do modelo MPS. Esta tem como objetivo a coordenação de tarefas, atividades e recursos que um projeto de software necessita para que se possa produzir o produto (SOFTEX, 2009).

Este processo envolve várias atividades, como: o desenvolvimento de um plano geral de controle do projeto; a obtenção do comprometimento e o mantimento do mesmo durante toda a execução do projeto; e conhecer o progresso e andamento do projeto de tal maneira que caso ocorra desvios ações corretivas possam ser tomadas (SANTANA; TIMÓTEO; VASCONCELOS, 2006). Abaixo estão as dezessete práticas que devem ser cumpridas deste processo:

- a) GPR1: O escopo do trabalho para o projeto está definido;
- b) GPR2: As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;
- c) GPR3: O modelo e as fases do ciclo de vida do projeto são definidos;
- d) GPR4: 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;
- e) GPR5: O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos;
- f) GPR6: Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;
- g) GPR7: Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;
- h) GPR8: Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;
- i) GPR9: Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;
- j) GPR10: Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos;

- k) GPR11: A viabilidade de atingir as metas do projeto é explicitamente avaliada considerando restrições e recursos disponíveis. Se necessário, ajustes são realizados;
- l) GPR12: O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido;
- m) GPR13: O progresso do projeto é monitorado com relação ao estabelecido no Plano do Projeto e os resultados são documentados;
- n) GPR14: O envolvimento das partes interessadas no projeto é gerenciado;
- o) GPR15: Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento;
- p) GPR16: Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas;
- q) GPR17: Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até sua conclusão.

O progresso da execução do projeto é definido com a comparação de informações reais de produtos de trabalho e tarefas, custo, esforço e cronograma definidos no projeto. As revisões de início de fase do projeto são fundamentais para verificar se todas as condições necessárias para que a fase possa ser iniciada foram atendidas, assim como revisões de fim de fase também têm como objetivo verificar se todas as condições necessárias para o fim de fase do projeto foram atendidas. Sempre que necessário deve-se realizar um re-planejamento e uma nova análise de viabilidade do projeto (SOFTEX, 2009).

Todas estas práticas serão detalhadas e avaliadas posteriormente na seção de mapeamento destes requisitos. Posteriormente será descrito o nível E com seus requisitos, pois somente neste nível há a evolução dos requisitos deste processo, no nível F não há alterações.

3.5 NÍVEL E

Até o nível F não há exigência de que as organizações devem ter os seus processos-padrão, assim os gerentes podem definir processos diferentes para os seus projetos, embora devam obedecer ao estabelecido na política organizacional definida na organização

para os processos. Nos níveis G e F muitas vezes a organização não tem ainda a preocupação de possuir processos integrados em um processo de desenvolvimento e manutenção.

O nível E do MPS.BR tem como foco principal a padronização dos processos da organização por meio da definição de processos padrão que devem ser definidos através das melhores práticas e processos existentes na organização, que consiste no primeiro passo de uma avaliação contínua e melhoria dos processos.

A definição de processos padrão não significa que todos os projetos tenham que usar os processos padrão da mesma forma. A definição dos processos padrão deve incluir a definição de diretrizes para a adaptação aos projetos, e a cada projeto é necessário definir o processo a ser utilizado no projeto a partir dos processos e diretrizes definidas. O nível E do MPS.BR exige ainda a construção de um repositório de medidas e de uma biblioteca de ativos.

No nível E, há a evolução dos requisitos descritos na GPR. Além da GPR há os processos: Definição do Processo Organizacional; Avaliação e Melhoria do Processo Organizacional e Medição. Estes, apesar da relação com a GPR não serão descritos, pois não condiz com o objetivo deste trabalho que foca na descrição e análise apenas dos requisitos da GPR (SOFTEX, 2011).

A Gerência de Projetos do nível E possui os seguintes requisitos:

- a) GPR4: O planejamento e as estimativas das tarefas do projeto são feitos baseados no repositório de estimativas e no conjunto de ativos de processo organizacional;
- b) GPR8: Os recursos e o ambiente de trabalho necessários para executar os projetos são planejados a partir dos ambientes padrão de trabalho da organização;
- c) GPR20: Equipes envolvidas no projeto são estabelecidas e mantidas a partir das regras e diretrizes para estruturação, formação e atuação;
- d) GPR21: Experiências relacionadas aos processos contribuem para os ativos de processo organizacional;
- e) GPR22: Um processo definido para o projeto é estabelecido de acordo com a estratégia para adaptação do processo da organização;

Todas estas práticas serão definidas e avaliadas posteriormente, onde será descrito se a metodologia ágil Scrum atende estes requisitos por meio de suas práticas. Para o próximo capítulo será descrito o processo da metodologia ágil Scrum.

4 MÉTODO ÁGIL DE DESENVOLVIMENTO SCRUM

Os métodos ágeis de desenvolvimento de software vêm nos últimos anos ganhando cada vez mais espaço nas indústrias de software. Assim como os métodos tradicionais de desenvolvimento de software estes tem também como objetivo a construção de sistemas com rapidez e qualidade.

Neste capítulo será abordado o conceito de métodos ágeis de desenvolvimento de software e estudado um dos objetos de estudo deste trabalho, o Scrum.

4.1 CONCEITO DE MÉTODOS ÁGEIS

Os métodos ágeis de desenvolvimento de software são uma solução para problemas encontrados na utilização de métodos convencionais de desenvolvimento. Mesmo com a evolução de técnicas e ferramentas nos últimos anos, a produção de software com qualidade e entregue dentro dos prazos e custos ainda é muito difícil (SOARES, 2004).

Estes métodos por sua vez abordam o processo de desenvolvimento de software de forma diferenciada dos métodos convencionais de desenvolvimento (COSTA FILHO et al, 2005). A idéia dos métodos ágeis é o foco nas pessoas e não em processos ou algoritmos. Além desta idéia há o objetivo de se gastar menos tempo com documentação e mais tempo com implementação.

Uma de suas principais características é a adaptação ao invés da previsibilidade dos métodos convencionais de desenvolvimento. Desta forma elas procuram se adaptar a novos fatores ocorridos durante o processo de desenvolvimento ao contrário dos métodos convencionais que tentam prever estes fatores antes do início do desenvolvimento.

Os métodos ágeis variam em termos de práticas e ênfases e compartilham algumas características como o desenvolvimento iterativo e incremental, comunicação e redução de produtos intermediários, como documentação extensiva. Assim existem maiores chances de se atender os requisitos do cliente, que muitas vezes se alteram ao longo do processo de desenvolvimento. (SOARES, 2004).

A seguir será descrito o conceito e funcionamento da metodologia de desenvolvimento ágil Scrum, que é considerado também um método específico para o gerenciamento do processo de desenvolvimento de software.

4.2 SCRUM

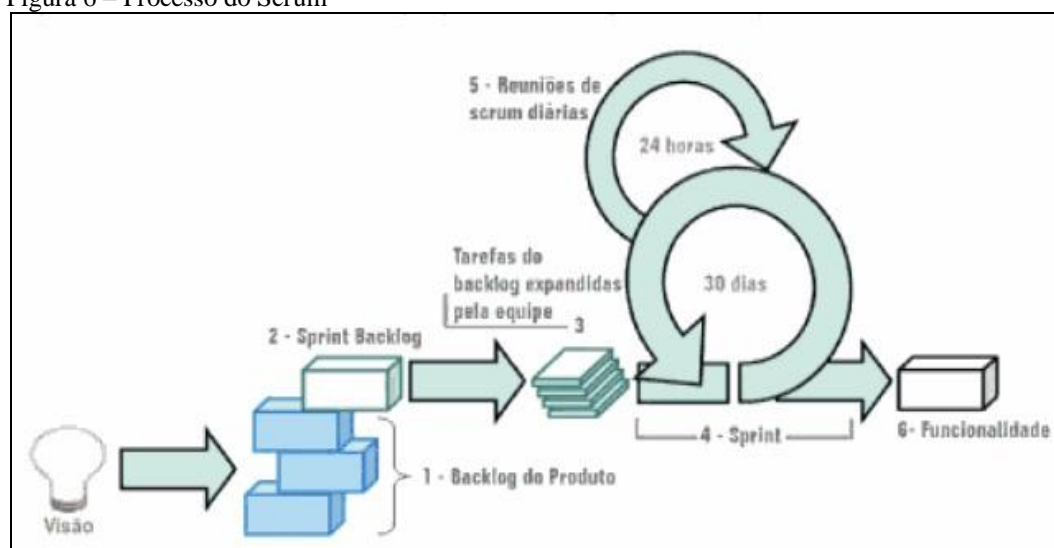
O Scrum foi criado em 1996 por Ken Schwaber e Jeff Sutherland e reúne atividades de monitoramento e feedback com reuniões rápidas e diárias com toda a equipe afim de identificar problemas e correções de impedimentos encontrados durante o processo de desenvolvimento (SOARES et al, 2007).

Este método ágil recebeu o nome de Scrum pois este é o nome usado pelo jogo chamado Rugby onde os jogadores se reúnem em círculo para planejar a próxima jogada. Esta é uma forma de mostrar que o projeto deve ser conduzido em pequenos círculos, porém com uma visão de longo prazo que é vencer o jogo (MARTINS, 2007).

Duas características importantes do Scrum é que ele é adaptativo e empírico. A maioria dos métodos são bastante prescritivas, onde isto faz com que os componentes da equipe sigam um roteiro pré-definido de passos, tornando o processo pouco flexível as mudanças.

A abordagem do Scrum é oposta ao modelo cascata sendo o processo iterativo onde inicia-se a análise com o levantamento de alguns requisitos disponíveis, assim que alguma análise tiver sido feita já é iniciado o processo de desenvolvimento. Sendo assim, pode-se dizer que o projeto trabalha em um pequeno pedaço de cada vez (MARTINS, 2007). A figura 6 resume o processo do Scrum.

Figura 6 – Processo do Scrum



Fonte: MARTINS (2007, p. 270)

A seguir será descrito todo o funcionamento do Scrum, suas fases, seus componentes e os papéis de cada componente durante o processo de desenvolvimento.

4.2.1 Product Backlog

O Product Backlog é uma lista inicial de requisitos que pode ser ajustada pelo Dono do Produto (Product Owner) durante o projeto a fim de mudar a seqüência dos itens que serão trabalhados.

Estes requisitos não necessitam estar descritos de forma completa e contém apenas uma breve descrição do que precisa ser feito. Maiores detalhes a respeito destes requisitos são verificados posteriormente com os usuários do produto ou com as “pessoas do negócio”. O Product Owner prioriza os requisitos de maior importância para o projeto onde estes vão para o topo da lista (MARTINS, 2007).

4.2.2 Sprint

Este pode ser definido como um conjunto de atividades de desenvolvimento conduzidas em um período pré-definido de tempo chamado time box (caixa de tempo), que pode variar de uma a quatro semanas ou mais. Este intervalo de tempo é definido de acordo com os riscos, complexidade do produto e grau de volatilidade dos requisitos. Segundo MARTINS (2007) cada Sprint consiste em uma ou mais equipes executando as seguintes atividades:

- a) desenvolver: definir o que precisa ser feito para implementar os itens do Backlog; no caso de software pode incluir análise, projeto técnico (design), programação, testes e documentação. O desenvolvimento consiste de microprocessos de descoberta, invenção e implementação.
- b) empacotamento: empacotar o produto; no caso de software implica na criação dos programas executáveis e outras formas de distribuição.
- c) revisão: todas as equipes reúnem-se para apresentar e discutir o trabalho feito e revisar o progresso, levantar e resolver problemas, e adicionar novos itens ao Backlog. O risco é revisado e as ações de controle são deflagradas.
- d) ajustes: consolidar as informações obtidas na reunião de revisão e fazer as mudanças necessárias no planejamento.

O sprint se inicia com uma reunião de planejamento entre o Product Owner e a equipe de desenvolvimento que trabalhará no projeto. Esta por sua vez é dividida em duas partes onde na primeira o Product Owner apresenta os requisitos de maior prioridade. A equipe então define o que poderá ser desenvolvido considerando a capacidade de produção.

Na segunda parte o time planeja o seu trabalho e define o Sprint Backlog, que são as tarefas necessárias para implementar as funcionalidades selecionadas no Product Backlog (SOARES et al, 2007).

No Scrum cada Sprint corresponde a uma iteração de normalmente trinta dias. Ao final de cada Sprint a equipe pode gerar uma nova versão e distribuí-la a seu(s) cliente(s). Esta versão tem que estar em condições de prover algum valor para o negócio ou que possa ser utilizada para prover algum direcionamento (MARTINS, 2007).

O Backlog Sprint é um quadro que é utilizado diariamente pela equipe para que cada membro aponte quantas horas ainda falta para a conclusão de cada item. Ele é uma forma de acompanhamento e uma de suas vantagens do Backlog Sprint é que este pode ser apresentado graficamente.

4.2.3 Fases

O Scrum possui três fases: pré-game, game e pós-game. Nas fases game e pós-game todos os processos, mais as entradas e saída, são bem definidos. Os conhecimentos sobre como realizar estas fases são explícitos e podem compreender mais de uma iteração do processo.

Na fase intermediária game, a maioria dos processos são indefinidos e tratados como uma caixa preta que requerem apenas controles externos. Nesta fase o produto é construído em Sprints que podem ser modificados a qualquer momento durante o planejamento. As mudanças podem ocorrer por conta da complexidade, ações da concorrência, prazo, qualidade, custos, etc (MARTINS, 2007).

4.2.3.1 Pré-Game

O pré-game inclui dois sub-passos: Planejamento e Arquitetura. O planejamento inclui a definição do sistema que esta sendo desenvolvido. O Product Backlog é criado contendo todos os requisitos conhecidos onde estes requisitos são priorizados e o esforço necessário para a implementação destes é estimado. O Product Backlog é constantemente atualizado onde alguns itens podem ser retirados, adicionados ou ter sua ordem de prioridade alterada. Em toda a iteração a atualização do Product Backlog é revisada pelo time para verificar os compromissos para a próxima iteração.

A arquitetura é planejada baseando-se nos itens que constituem o Product Backlog. No caso de alterações, as mudanças necessárias para implementação dos itens são identificados ao longo com os problemas que os mesmos podem causar (ABRAHAMSSON; SALO; RONKAINEN, 2002).

4.2.3.2 Game

Esta fase é composta por um ou mais Sprints onde o objetivo é o desenvolvimento dos itens ou das funcionalidades de um novo release, respeitando as restrições de prazo, requisitos, qualidade, custos e características dos concorrentes. Segundo MARTINS (2007), esta é composta pelos seguintes macropassos:

- a) fazer reunião com as equipes para rever o planejamento dos releases.
- b) revisar os padrões com os quais o sistema precisa ser compatível e fazer os ajustes necessários.
- c) realizar os Sprints até que o produto esteja pronto para distribuição.

4.2.3.3 Pós-Game

Quando a gerência decide que o produto está adequado para a distribuição, ela declara o fechamento do release e a fase de pós-game começa. Nesta fase o produto é preparado para a distribuição, incluindo integração, documentação, testes, marketing entre outras coisas (MARTINS, 2007).

4.2.4 Funcionamento

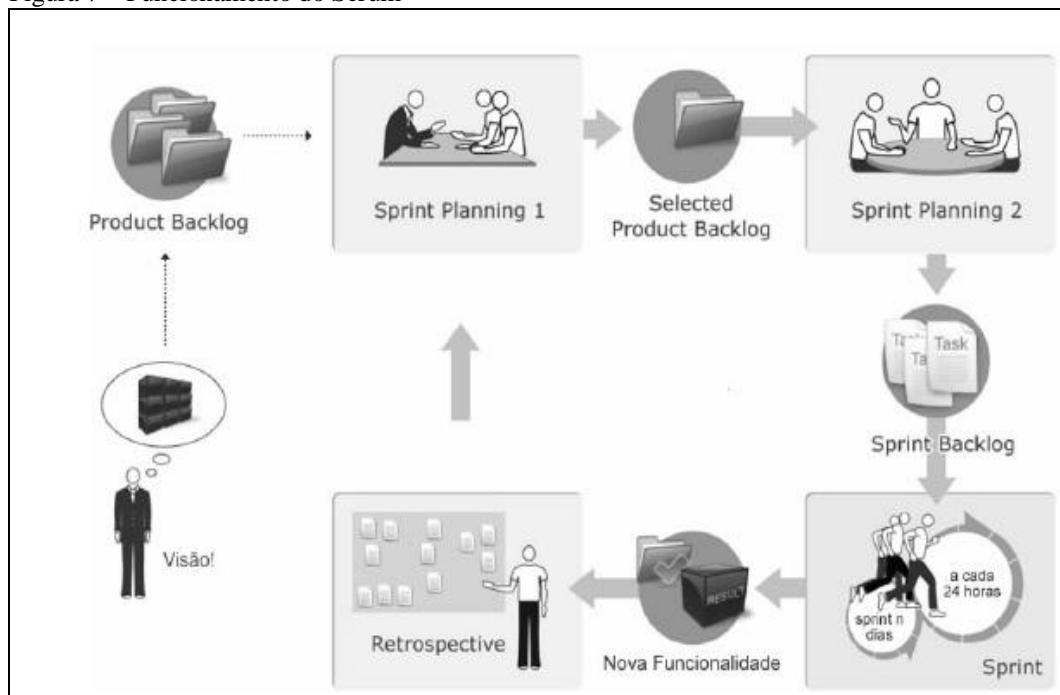
No Scrum existem três papéis, os quais são: o Product Owner que é o dono do produto; o Time que são todos os colaboradores que farão parte da equipe de planejamento e desenvolvimento do projeto; e o Scrum Master que é o responsável que garante que todos sigam as regras do Scrum além de ser o responsável por remover os impedimentos do projeto.

Um projeto no Scrum inicia-se com uma visão do produto, onde esta contém a lista de requisitos estabelecida pelo cliente. Após o levantamento de alguns requisitos é montado o Product Backlog que é priorizado e dividido em releases.

Como já descrito, o Scrum é um método iterativo e cada iteração é chamada de Sprint. O Sprint inicia-se com uma reunião de planejamento entre o Product Owner e o Time.

Esta reunião é dividida em duas partes onde na primeira parte o Product Owner apresenta os requisitos de maior importância e prioriza aqueles que devem ser implementados, logo após o Time define o que poderá entrar no próximo Sprint levando em conta a sua capacidade de produção. Na segunda parte o time planeja o seu trabalho definindo o Sprint Backlog, que são as tarefas necessárias para implementar os itens selecionados no Product Backlog (SOARES et al, 2007). A Figura 7 exemplifica o funcionamento do método.

Figura 7 – Funcionamento do Scrum



Fonte: SOARES et al (2007, p. 4)

No Scrum, durante as iterações são realizadas reuniões diárias de até quinze minutos onde segundo Soares et al (2007) cada membro do time responde a três perguntas: O que eu fiz no projeto desde a última reunião? O que irei fazer até a próxima reunião? Quais são os impedimentos?

Ao final de um Sprint é realizada uma reunião de revisão onde o time apresenta os resultados alcançados na iteração, desta forma são verificadas as funcionalidades e as adaptações podem ser efetuadas. Logo após, o Scrum Master realiza uma reunião de retrospectiva com o objetivo de verificar pontos em que o Time pode melhorar para as próximas iterações (SOARES et al, 2007).

5 TRABALHOS CORRELATOS

Neste capítulo serão abordados os trabalhos correlatos que foram encontrados durante a pesquisa e que serviram de base para uma melhor compreensão dos objetivos à serem atingidos deste trabalho.

5.1 PROPOSTA DE UM MODELO DE REFERÊNCIA PARA DESENVOLVIMENTO DE SOFTWARE COM FOCO NA CERTIFICAÇÃO DO MPS.BR

Neste trabalho Neto (2008) descreve que desde os anos noventa as empresas de software realizam esforços na busca por um alto padrão de qualidade, porém para pequenas empresas de software estes esforços ainda são insuficientes. Com isto, o objetivo do trabalho tem seu foco na elaboração de um modelo de referência para pequenas empresas de software para certificação do MPS.BR.

No trabalho também são realizados estudos com metodologias ágeis de desenvolvimento de software para o auxílio na busca pelo modelo para certificação do MPS.BR. Também são realizadas pesquisas de campo com algumas empresas de software, e o modelo proposto no trabalho foi aplicado em uma das empresas com sucesso.

5.2 UMA SOLUÇÃO SISTÊMICA PARA A GERÊNCIA DE ATIVOS NO CONTEXTO DA IMPLEMENTAÇÃO DOS NÍVEIS DO MPS.BR

No trabalho realizado por Jesus (2009) verifica a inexistência de uma ferramenta homologada pela SOFTEX para o auxílio na implementação do modelo de qualidade MPS.BR, e como tal solução visa fornecer a catalogação de ativos e boas práticas para todos os níveis do MPS.BR. Para isto foi feito um estudo sobre o processo de software e as normas e modelos de qualidade com foco principal no modelo MPS.BR que foi desenvolvido para a realidade brasileira.

Como conclusão do trabalho o mesmo criou o Siga-MPS.BR, que é uma solução para a catalogação das boas práticas. Outro objetivo do trabalho também foi a utilização desta ferramenta em certificação oficial para analisar a implementação tida como base para o modelo MPS.BR, que proporciona uma melhor gestão do conhecimento dos ativos organizacionais usados como parâmetros para a execução de tal atividade.

5.3 MAPEAMENTO DOS PROCESSOS DE DESENVOLVIMENTO ÁGEIS EM RELAÇÃO AO MODELO DE MELHORIA DO PROCESSO DE SOFTWARE DO BRASIL (NÍVEL G)

Neste trabalho Santos (2009) fez uma análise dos resultados alcançados utilizando métodos ágeis de desenvolvimento de software para certificação do nível G do modelo MPS.BR. Teve como objetivo do trabalho um guia para que as empresas possam utilizar para consultar e atacar os pontos específicas para adquirir a certificação do nível G.

Na conclusão do trabalho o mesmo verificou que as metodologias ágeis não contemplam todos os objetivos necessários para certificação do nível G pelo fato de que alguns requisitos do modelo MPS.BR não deixam de forma clara a necessidade ou não do registro de informações da execução de tal tarefa.

6 MAPEAMENTO DAS PRÁTICAS DO SCRUM EM RELAÇÃO AOS REQUISITOS REFERENTE À GERÊNCIA DE PROJETOS DO NÍVEL E DO MPS.BR

Neste capítulo serão definidos os critérios de avaliação de conformidade dos requisitos descritos na GPR do nível E do MPS.BR. Posteriormente serão definidos e avaliados individualmente os requisitos GPR4 (evolução do nível G), GPR8, GPR20, GPR21 e GPR22.

6.1 CRITÉRIOS DE AVALIAÇÃO DOS REQUISITOS

Para que se possa ser alcançado o objetivo do trabalho é necessário a definição de critérios de avaliação individual dos requisitos. Para melhor interpretação da avaliação de conformidade dos requisitos, os critérios de avaliação serão os mesmos definidos na apresentação do mapeamento dos requisitos do nível G do MPS.BR.

6.2 DEFINIÇÃO E AVALIAÇÃO DOS REQUISITOS DO NÍVEL E

Será apresentada a definição dos requisitos do nível E juntamente com sua classificação de conformidade com o Scrum aplicando-se os critérios previamente definidos.

6.2.1 Definição da GPR4

Para atender a este requisito, a realização das estimativas e do planejamento do projeto é baseada no processo definido para o projeto, nos ativos de processo e no repositório de medidas da organização onde os ativos de processo e repositório de medidas da organização estão definidos no escopo do processo Definição do Processo Organizacional.

Para a realização das estimativas no nível E, ao contrário do nível G, é necessária a construção de bases históricas. Bases históricas consistentes são baseadas nos projetos já realizados pela organização e além de possuírem o objetivo de facilitar as estimativas também proporcionam um melhor entendimento do comportamento dos projetos da organização.

Em empresas que contam com um pequeno número de projetos, a construção de uma base histórica pode não ser trivial. Uma solução seria analisar as atividades dentro do próprio projeto que está em andamento e não entre outros projetos. Como exemplo, se as

tarefas do projeto forem divididas em iterações podem-se utilizar os dados históricos da primeira iteração como base para a realização do planejamento da segunda iteração (SOFTEX, 2011).

6.2.1.1 Avaliação da GPR4

Assim como já descrito na definição do requisito em questão, diferente do nível G, no nível E é necessário o registro de bases históricas para que se possam construir as estimativas. Conforme Santos (2010) no nível G não é exigido o registro de bases históricas onde devido ao fato de o Scrum não mencionar formas para armazenar estimativas de esforço, o Scrum atende parcialmente o requisito.

O Scrum é um framework onde no mesmo podem-se empregar diversos processos e técnicas para o desenvolvimento de produtos complexos (SANTOS; SANTOS, 2010). Com isto para que se tenha possibilidade de satisfação desta condição, a organização pode encontrar uma forma própria ou buscar em outras fontes métodos adequados para armazenar as informações de custo e esforço dos Sprints ou iterações do Scrum.

Com a armazenagem destes dados abre-se a possibilidade da construção de uma base histórica para que se possa ser feito o planejamento e estimativa nas próximas iterações do processo. Conclui-se que, para o nível E, o Scrum não atende o requisito pelo mesmo fato de que o mesmo não menciona formas de armazenagem de informações de esforço fazendo com que para a construção de bases históricas sejam buscados outros métodos fora da metodologia ágil.

6.2.2 Definição da GPR8

Para este requisito têm-se a necessidade para planejar as tarefas, com base na EAP (Estrutura Analítica do Projeto) ou estrutura equivalente, assim como prover os recursos e o ambiente necessários incluindo equipamentos, ferramentas, serviços, componentes e requisitos do processo. A partir do nível E, o planejamento do ambiente de trabalho para o projeto deve estar compatível com o ambiente padrão de trabalho da organização onde este encontra-se definido na DFP7 do processo de Definição do Processo Organizacional.

Todos os recursos precisam estar planejados, mesmo os já disponíveis ou que serão compartilhados com outros projetos, uma vez que se trata de sua alocação para uso. As adaptações ao planejamento do ambiente também precisam estar documentadas.

6.2.2.1 Avaliação da GPR8

Muitos projetos possuem uma lista de requisitos como tipo de sistema, itens de planejamento, tipo de aplicação, ambiente de desenvolvimento, etc. No Scrum, estes requisitos são definidos no Backlog de Produto que nada mais é do que uma lista de requisitos definidos pelo Dono do Produto onde esta pode ser alterada ao longo do projeto (MARTINS, 2007).

Ao longo do projeto o Scrum Master é o responsável por prover os recursos necessários conforme as necessidades e impedimentos que são reportados nas reuniões (SANTOS; SANTOS, 2010). Como no Scrum há uma definição de como os requisitos necessários ao ambiente de desenvolvimento devem ser documentados no planejamento do projeto, concluí-se que assim como no nível G o Scrum satisfaz este requisito para o nível E.

6.2.3 Definição da GPR20

Neste o resultado esperado é o estabelecimento das equipes de acordo com as regras para estruturação, formação e atuação das equipes na empresa. As regras, por sua vez, estão definidas na DFP8 do processo Definição do Processo Organizacional.

Na DFP8 do processo Definição do Processo Organizacional define que a estruturação das equipes está relacionada ao número, tipo de equipes e como elas se relacionam. Quanto à formação das equipes trata-se da definição de um organograma, definição de membros e líderes e a alocação de recursos necessários. Por fim, a atuação define como as equipes são formadas e como elas interagem para o alcance de seus objetivos incluindo a compreensão pelos participantes sobre a contribuição dos padrões de trabalho e participação na equipe.

A visão compartilhada do projeto consiste no entendimento comum de seus princípios diretores como missão, objetivos, comportamento esperado, valores e resultados finais. Esta visão é definida antes da criação da equipe onde a equipe pode ser tratada como sendo toda a organização ou interessados externos ao projeto (SOFTEX, 2011).

6.2.3.1 Avaliação da GPR20

Conforme definido neste requisito o mesmo trata da estruturação, formação e atuação das equipes envolvidas no projeto. Com relação à formação das equipes, no Scrum o mesmo possui para os envolvidos no projeto três papéis principais: o Product Owner, Scrum Master e o Scrum Team.

Redefinindo os papéis de forma rápida, o Product Owner é quem define e prioriza os requisitos; define datas; avalia os Sprints, etc. O ScrumMaster elimina os impedimentos do time; garante que o time esteja funcional e produtivo; protege o time de interferências externas; etc. Já o Scrum Team possui entre cinco e nove membros; é multifuncional; seleciona entre os itens priorizados os que irão ser executados no Sprint; tem o direito de fazer o que quiser dentro do Sprint para cumprir com o objetivo; etc (PEREIRA; TORREÃO: MARÇAL, 2007).

A alocação dos recursos necessários a equipe é feita pelo Product Owner e fica registrada no Product Backlog. Com relação à atuação das equipes, no processo do Scrum a interação entre os participantes de acordo com seus papéis estão bem definidos onde, através de reuniões de planejamento de Sprints, reuniões diárias rápidas, reunião de revisão e reunião de retrospectiva os membros definem o melhor caminho para o alcance de seus objetivos.

Verificando o quesito estruturação das equipes, os Scrum não descreve número de equipes, tipo e nem relacionamento entre as mesmas. Porém conforme verificado, o Scrum possui seus papéis e funções bem definidas atendendo aos quesitos de formação e atuação das equipes, porém não descreve nada a respeito da estruturação das mesmas. Desta forma, entende-se que o Scrum atende parcialmente este requisito.

6.2.4 Definição da GPR21

Para este requisito é necessário que haja a identificação das experiências obtidas na execução do projeto que possam contribuir para o conjunto de ativos de processo organizacionais. As experiências obtidas incluem, mas não se limitam a, medidas coletadas, dados para refinamento de estimativas, lições aprendidas, boas práticas, etc.

Referente ao repositório de medidas, o mesmo deve ser constantemente atualizado seguindo os resultados esperados do processo de Medição. Com a implementação dos resultados esperados no nível E, os dados do repositório tendem a ficar mais organizados,

onde ao seguirem um mesmo padrão as informações coletadas nos projetos devem ter uma estrutura padronizada.

Não apenas o repositório de medias que é realimentado com os dados dos projetos da empresa, as experiências adquiridas também contribuem para os ativos de processo organizacional como, por exemplo, lições aprendidas e coletadas durante as reuniões periódicas das equipes dos projetos. Solicitações de melhoria nos processos identificadas devido ao uso destes processos durante os projetos também contribuem para a melhoria dos ativos de processo organizacional (SOFTEX, 2011).

6.2.4.1 Avaliação da GPR21

No Scrum, o mesmo não menciona a respeito de armazenagem ou a criação e manutenção de um repositório de medidas. Quando a equipe consegue entregar tudo no prazo correto, isto é resultado do empenho e dedicação e não da precisão das estimativas. A imprecisão gera preocupações onde algumas empresas tentam melhorar a precisão das suas estimativas construindo um banco de dados dos projetos com as estimativas e as horas reais utilizadas, e derivas as variações (MARTINS, 2007).

Com o que foi descrito anteriormente, justificando-se o fato de que o Scrum não define formas de criação e manutenção de um repositório de medidas conclui-se que o mesmo não contempla este requisito.

6.2.5 Definição da GPR22

Para este resultado, deve-se definir um processo a partir de um conjunto de processos padrão da empresa, onde o conjunto de processos padrão da organização define-se em um produto do processo Definição do Processo Organizacional, assim como as estratégias para adaptação deste processo e os modelos de ciclo de vida que devem ser utilizados nos projetos.

O processo definido para o projeto deve abranger todas as atividades para o desenvolvimento e manutenção do produto a partir do que é pedido pelo o cliente até a entrega do produto.

A definição de um processo para o projeto, de um processo padrão e de diretrizes para o projeto são descritas no processo Definição do Processo Organizacional. Para este resultado deve-se também garantir que ao longo da execução do projeto as regras de

adaptação sejam seguidas, onde como exemplo, caso durante a execução do projeto se perceba que uma determinada atividade não é mais necessária, deve-se justificar sua exclusão, se possível, utilizando as diretrizes definidas pela empresa (SOFTEX, 2011).

6.2.5.1 Avaliação da GPR22

Definindo-se de forma rápida o Scrum, o mesmo se inicia por uma visão que pode ser técnica ou de marketing, mas que fica mais clara à medida que o projeto evolui. A partir desta visão é construída uma lista de itens priorizados chamada de Product Backlog, que é constituída por requisitos e funcionalidades necessárias para a construção do produto. A partir desta lista são realizadas reuniões e são construídos os Sprints que são compostos por uma seleção de itens que a equipe acredita que possa construir até o final da iteração. Ao final de cada iteração a equipe apresenta o incremento de funcionalidade para o cliente que vai inspecionar e caso necessário sugerir ou solicitar adaptações que por sua vez serão adicionados no Product Backlog. Antes do início de cada iteração a equipe realiza uma reunião de retrospectiva das ações que deram resultado positivo e negativo e define mudanças na sua forma de trabalhar. Durante cada iteração, itens que a equipe definir que não possam ser desenvolvidos podem ser reportados ao Mestre Scrum que pode removê-los a qualquer momento. Novos itens também podem ser adicionados durante cada iteração (MARTINS, 2007).

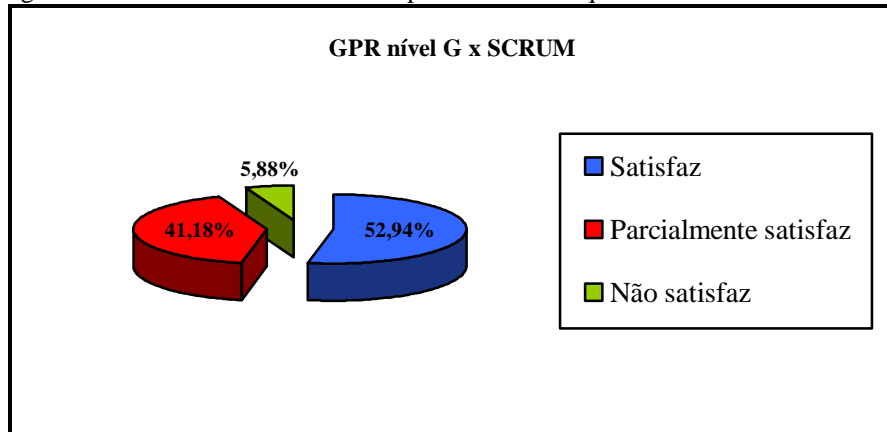
Como descrito na definição do resultado esperado, este por sua vez consiste na definição de um processo para o projeto que engloba todas as atividades para o desenvolvimento e manutenção do produto, para isto, pode-se adotar o Scrum como um processo padrão para o desenvolvimento de software organização. Conclui-se por sua vez que o Scrum atende a este requisito.

6.3 ANÁLISE DAS AVALIAÇÕES

Anteriormente foram definidos e avaliados os requisitos da GPR do nível E com o Scrum. Nesta seção serão apresentados os gráficos evidenciando a conformidade dos requisitos para o nível E e o cruzamento destas informações com as avaliadas para o nível G do MPS.BR.

No mapeamento dos requisitos do nível G chegou-se a uma conclusão de que o Scrum satisfaz 52,94% dos requisitos solicitados pela GPR. A figura 8 demonstra esta avaliação (SANTOS; SANTOS, 2010).

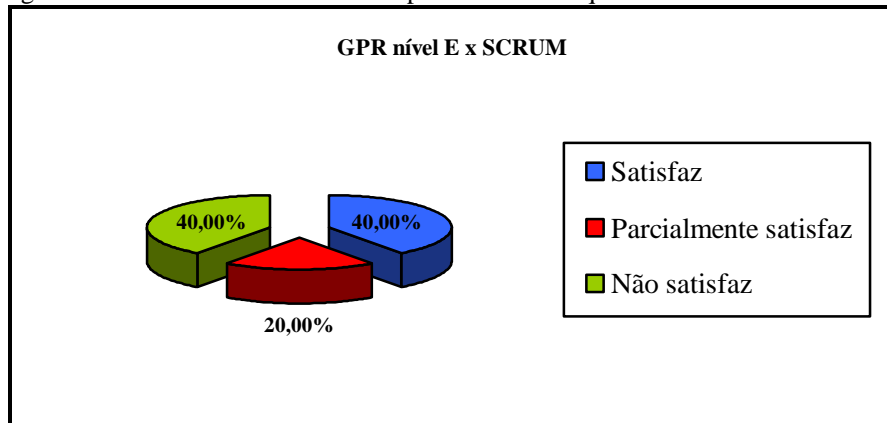
Figura 8 – Resultado da análise do mapeamento dos requisitos do nível G



Fonte: SANTOS e SANTOS (2010)

Analisando somente os requisitos do nível E percebe-se que o mesmo satisfaz apenas 40% dos requisitos, onde do restante atende parcialmente 20% e não atende 40%. A figura 9 representa estas informações.

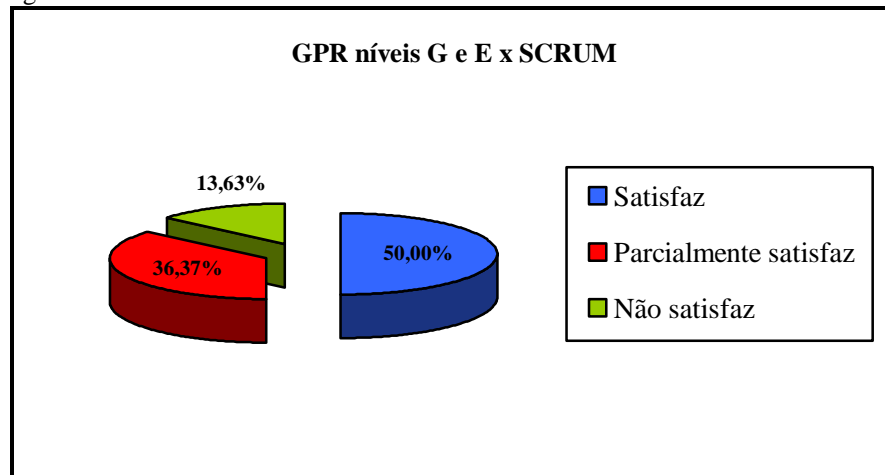
Figura 9 – Resultado da análise do mapeamento dos requisitos do nível E



Fonte: do autor.

Cruzando as informações das avaliações da GPR dos níveis G e E verificamos que o Scrum contempla apenas 50% dos requisitos. O Scrum atende ainda parcialmente 36,37% dos requisitos e não atende 13,63% dos requisitos. A figura 10 demonstra estes dados de forma gráfica.

Figura 10 – Cruzamento dos resultados das análises níveis G e E do MPS.BR



Fonte: do autor.

As informações utilizadas para o cruzamento das análises foram obtidas no trabalho feito por SANTOS (2010) que avalia os requisitos da GPR do nível G. Conforme verificado nesta seção foram apresentadas as análises dos mapeamentos dos requisitos da GPR do nível E do MPS.BR. No próximo capítulo será apresentada a conclusão do trabalho juntamente com sugestões para trabalhos futuros.

7 CONCLUSÃO

Neste trabalho foram realizados mapeamentos ou análises de satisfação dos requisitos descritos no guia de implementação do MPS.BR entre o Scrum e a GPR do nível E. Deve-se levar em consideração que as análises feitas são totalmente passíveis de diferentes interpretações, pois as metodologias ágeis podem sofrer alterações e possuir interpretações, que podem afetar na compreensão e análise dos requisitos.

Pode-se verificar também que ao cruzar as informações das análises do nível G com o nível E o Scrum atende a apenas 50% do total de vinte e dois requisitos. O Scrum não satisfaz a apenas 13,63% dos requisitos.

Para os requisitos com satisfação parcial, deve-se levar em consideração o fato de que o Scrum por ser uma metodologia ágil dispensa a parte de documentação e tem foco na execução e planejamento das tarefas de forma dinâmica com o objetivo de agilizar o desenvolvimento de software. Considera-se também, como exemplo, na GPR20 o Scrum define os papéis dos integrantes das equipes, porém não menciona informações a respeito de número de equipes e nem de relacionamento entre ambas. Fatores como falta de documentação e incompletude de informações contribuíram para que alguns requisitos fossem classificados como parcialmente satisfeitos.

Sugere-se, que como as metodologias ágeis são dinâmicas e de fácil adaptação, as organizações procurem meios próprios ou externos para solução dos problemas de não conformidade ou conformidade parcial de requisitos, já que as mesmas dão possibilidades de customizações a seus adeptos.

Como trabalhos futuros sugere-se estender a análise para os próximos níveis do MPS.BR aplicando Scrum ou realizar a mesma análise porém aplicando práticas de outras metodologias ágeis afim de verificar qual melhor se adéqua ao modelo de qualidade MPS.BR.

REFERÊNCIAS

ABRAHAMSSON, Pekka; SALO, Outi; RONKAINEN, Jussi. **Agile Software development methods: Review and analysis**. Otamedia: Vtt Publications, 2002. 107 p. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.161.5931&rep=rep1&type=pdf>>. Acesso em: 15 out. 2011.

ATAÍDES, Adriana da Costa. **UM MÉTODO PARA ACOMPANHAMENTO E CONTROLE DA IMPLANTAÇÃO DO CMMI**. 2006. 125 f. Dissertação (Mestrado) - Universidade de Brasília, Brasília, 2006. Disponível em: <<http://repositorio.bce.unb.br/bitstream/10482/6733/1/ADRIANA%20DA%20COSTA%20ATA%20C3%84DES.pdf>>. Acesso em: 10 out. 2011.

COLENCI NETO, Alfredo. **Proposta de um modelo de referência para desenvolvimento de software com foco na certificação do MPS.BR**. 2008. 179 f. Tese (Doutorado) - Universidade de São Paulo, São Carlos, 2008. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18140/tde-29012009-093822/publico/AlfredoColenciNeto.pdf>>. Acesso em: 19 nov. 2011.

COSTA FILHO, Edes Garcia da et al. **Padrões e Métodos Ágeis: agilidade no processo de desenvolvimento de software**. Disponível em: <http://subversion.assembla.com/svn/TCC_Agil/9673.pdf>. Acesso em: 31 maio 2011.

DONEGAN, Paula et al. **Métricas de Software: Um Mapeamento entre Six Sigma e CMMI**. Fortaleza: 2005. Disponível em: <http://www.simpros.com.br/simpros2005/upload/A10_3_artigo14725.pdf>. Acesso em: 5 set. 2011.

GIORDANI, Fabricio. **PROPOSTA PARA IMPLEMENTAÇÃO DE UMA METODOLOGIA PARA DEFINIÇÃO DE CUSTO E ESFORÇO PARA O PROJETO DE DESENVOLVIMENTO DE APLICAÇÕES EMPRESARIAIS UTILIZANDO METODOLOGIAS ÁGEIS e Scrum para o Desenvolvimento de Software**. Disponível em: <<http://www.bib.unesc.net/biblioteca/sumario/000043/00004307.pdf>>. Acesso em: 23 maio 2011.

JESUS, Anderson Jones Silva de. **UMA SOLUÇÃO SISTÊMICA PARA A GERÊNCIA DE ATIVOS NO CONTEXTO DA IMPLEMENTAÇÃO DOS NÍVEIS DO MPS.BR Trabalho**. 2009. 79 f. Dissertação (Bacharel) - Universidade Federal Do Pará, Belém, 2009. Disponível em: <http://www.spider.ufpa.br/projetos/siga_mpsbr/tcc.pdf>. Acesso em: 19 nov. 2011.

MARTINS, José Carlos Cordeiro. **Técnicas para Gerenciamento de Projetos de Software**. Rio de Janeiro: Brasport, 2007. 427 p.

PEREIRA, Paulo; TORREÃO, Paula; MARÇAL, Ana Sofia. Entendendo o Scrum para Gerenciar Projetos de Forma Ágil. **Mundo Pm**, n. , p.1-11, 06 mar. 2007. Disponível em: <<http://www.sotomcomunica.com.br/homolog/siq/DOCS/EntendendoScrumparaGerenciarProjetosdeFormaAgil.pdf>>. Acesso em: 05 abr. 2012.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995. 1056 p. ISBN 8534602379
Número de Chamada: 005.1 P935e 1995

REZENDE, Denis Alcides. **Engenharia de Software e Sistemas de Informação**. 3° Rio de Janeiro: Brasport, 2005. 313 p.

SANTANA, Célio A.; TIMÓTEO, Aline L.; VASCONCELOS, Alexandre 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: <<http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2006/009.pdf>>. Acesso em: 15 out. 2011.

SANTOS, Claudio Ari Bergossi. **Mapeamento dos processos de desenvolvimento ágeis em relação ao Modelo de Melhoria do Processo de Software do Brasil (nível G)**. 2009. 43 f. Monografia (Bacharel) - Universidade Estadual de Feira de Santana, Feira de Santana, 2009. Disponível em: <http://tccbiscoito.googlecode.com/svn-history/r46/trunk/monografia_biscoito.pdf>. Acesso em: 19 nov. 2011.

SANTOS, Claudio A. B.; SANTOS, José Amancio Macedo. **Mapeamento das práticas do Scrum em relação aos requisitos referente a Gerência de Projetos do nível G do MPS.BR**. Feira de Santana - Ba: 2010. Disponível em: <<http://www.cesmac.com.br/erbase2010/papers/wticg/65120.pdf>>. Acesso em: 05 abr. 2012.

SATO, Danilo Toshiaki. **Uso eficaz de métricas em métodos ágeis de desenvolvimento de software**. Disponível em: <<http://grenoble.ime.usp.br/~gold/orientados/dissertacaoDaniloSato.pdf>>. Acesso em: 23 maio 2011.

SELNER, Claudiomir. **ANÁLISE DE REQUISITOS PARA SISTEMAS DE INFORMAÇÕES, UTILIZANDO AS FERRAMENTAS DA QUALIDADE E PROCESSOS DE SOFTWARE**. 1999. 156 f. Dissertação (Mestrado) - Universidade Federal De Santa Catarina, Florianópolis, 1999. Disponível em: <<http://www.kugel.com.br/download/publicacoes/dissertacao.pdf>>. Acesso em: 15 out. 2011.

SOARES, Felipe S. Furtado et al. **Adoção de SCRUM em uma Fábrica de Desenvolvimento Distribuído de Software**. Recife: 2007. Disponível em: <<http://200.17.137.110:8080/schisto/publicacoes/i-wdds/adocao-de-scrum-em-uma-fabrica-de-desenvolvimento.pdf>>. Acesso em: 20 out. 2011.

SOARES, Michel Dos Santos. **Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software**. Disponível em: <<http://revistas.facecla.com.br/index.php/reinfo/article/view/146/38>>. Acesso em: 23 maio 2011.

SOFTEX (Org.). MPS.BR - Melhoria de Processo do Software Brasileiro: **Guia de implementação - Parte 1: Fundamentação para Implementação do Nível G do MR-MPS**. Campinas, 2009. 31 p. Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte_1_2009.pdf>. Acesso em: 23 jun. 2011.

SOFTEX (Org.). MPS.BR - Melhoria de Processo do Software Brasileiro: **Guia de implementação - Parte 1: Fundamentação para Implementação do Nível E do MR-MPS**. Campinas, 2011. 67 p. Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte_3_2011.pdf>. Acesso em: 04 jun. 2012.

ZANATTA, Alexandre Lazaretti; VILAIN, Patrícia. **Uma análise do método ágil Scrum conforme abordagem nas áreas de processo Gerenciamento e Desenvolvimento de Requisitos do CMMI**. Passo Fundo: 2003. Disponível em: <http://wer.inf.puc-rio.br/WERpapers/pdf_counter.lua?wer=WER05&file_name=alexandre_zanatta.pdf>. Acesso em: 10 out. 2011.

WEBER, Kival Chaves et al. **Modelo de Referência e Método de Avaliação para Melhoria do Processo de Software - versão 1.0 (MR-MPS e MA-MPS)**. In: **IV, 4., 2005, Florianópolis**. Modelo de Referência e Método de Avaliação para Melhoria do Processo de Software - versão 1.0 (MR-MPS e MA-MPS). Florianópolis: Softex, 2005. p. 1 - 14. Disponível em: <http://golden.softex.br/portal/softexweb/uploadDocuments/MR_MA-MPS.pdf>. Acesso em: 26 jun. 2011.

APÉNDICE(S)

APÊNDICE A - MAPEAMENTO DAS PRÁTICAS DO SCRUM EM RELAÇÃO AOS REQUISITOS REFERENTE À GERÊNCIA DE PROJETOS DO NÍVEL G DO MPS.BR

Neste capítulo, o objetivo principal é mapear as práticas da metodologia ágil Scrum em relação aos requisitos do sub-processo GPR do nível G do MPS.BR, mostrando como e quanto as práticas do Scrum atingem os resultados esperados por este sub-processo (SANTOS; SANTOS, 2010).

1 CRITÉRIOS DE CLASSIFICAÇÃO DOS REQUISITOS DA GPR DO NÍVEL G

Para realizar o mapeamento teve-se a necessidade do estabelecimento de critérios de classificação da equivalência entre as práticas do Scrum e o nível G do MPS.BR. Conforme mostra a Tabela 2, foram definidos três critérios para classificação da metodologia ágil em relação aos requisitos da GPR. Os critérios selecionados tiveram como base um trabalho que realizou o mapeamento entre o Scrum e o modelo de qualidade CMMI (SANTOS; SANTOS, 2010).

Tabela 2 – Critérios para avaliação dos requisitos da GPR

Classificação	Critério
(S) Satisfeito	A prática está totalmente em conformidade.
(PS) Parcialmente Satisfeito	Há evidências, embora não esteja plenamente em conformidade.
(NS) Não Satisfeito	Não há evidências.

Fonte: SANTOS e SANTOS (2010).

2 MAPEAMENTO

Nesta seção será demonstrado o mapeamento das práticas do Scrum com os requisitos da GPR do nível G. São dezessete requisitos mapeados onde para o mapeamento será descrita a definição geral dos resultados esperados descritos no guia de implementação da SOFTEX e posteriormente será descrita a análise e classificação do requisito.

2.1 GPR1 – O ESCOPO DE TRABALHO PARA O PROJETO É DEFINIDO

O escopo do projeto define todo o trabalho necessário, e somente ele, para realizar a entrega de um produto que satisfaça as necessidades para o projeto. O escopo é o ponto de partida para o planejamento do projeto onde sua definição deve estabelecer o que está e o que não está incluído no projeto. Ele contém a definição do objetivo, os limites e restrições, todos os produtos que serão entregues e outros gerados pelo projeto, dentre outras informações.

A representação do escopo pode ser feita por meio de uma EAP também conhecida como WBS (Work Breakdown Structure). A EAP disponibiliza um meio de identificação e organização das unidades lógicas de trabalho a serem gerenciadas, que são chamadas de “pacotes de trabalho”. Este resultado pode ser implementado por meio de um Documento de Visão e outro que defina claramente o escopo do trabalho (SOFTEX, 2009).

2.1.1 Avaliação da GPR1

Para que se possa satisfazer este requisito o Scrum utiliza o product backlog, onde este possui uma listagem de todos os requisitos que o sistema deve apresentar, este representa o escopo do projeto. Desta forma, concluí-se que o Scrum atende este requisito (SANTOS; SANTOS, 2010).

2.2 GPR2 – AS TAREFAS E OS PRODUTOS DE TRABALHO DO PROJETO SÃO DIMENSIONADOS UTILIZANDO MÉTODOS APROPRIADOS

O escopo do projeto, identificado na forma dos seus principais produtos de trabalho e das tarefas do projeto, deve agora ser dividido em partes menores, que possam ser facilmente gerenciadas e possíveis de serem dimensionadas.

Como as tarefas têm de ser divididas em partes menores, deve-se estabelecer uma estrutura de decomposição para as mesmas, onde esta pode ser a EAP do projeto ou uma estrutura equivalente. Esta estrutura deve fornecer uma referência para atribuição de tamanho, esforço, cronograma e responsabilidades e é utilizada como estrutura subjacente para gerenciar o projeto.

Este resultado se diz respeito a estimativa de tamanho onde tamanho é definido como a dimensão das funcionalidades sob o ponto de vista do usuário. São contadas classes, objetos, relatórios, etc. No nível G, as estimativas podem ser feitas com base na

complexidade, número de requisitos ou o uso da EAP juntamente com dados históricos e a experiência em projetos anteriores (SOFTEX, 2009).

2.2.1 Avaliação da GPR2

Após a criação do selected backlog, no Scrum, a equipe divide os itens selecionados em tarefas menores para a criação do sprint backlog. No Scrum não há orientação explícita para estabelecer tamanho ou complexidade dos elementos do product backlog e sprint backlog. Nenhum método é mencionado na metodologia ágil para orientar no dimensionamento dos produtos de trabalho como também não mantém dados históricos sobre a decomposição dos requisitos. Devido a estes fatos o Scrum atende parcialmente estes requisitos (SANTOS; SANTOS, 2010).

2.3 GPR3 – O MODELO E AS FASES DO CICLO DE VIDA DO PROJETO SÃO DEFINIDOS

O ciclo de vida de um projeto consiste de fases e atividades que são definidas de acordo com o escopo dos requisitos, as estimativas e os recursos e visa oferecer maior controle gerencial. O ciclo define fases, que geram produtos de trabalho necessários para as fases posteriores. Esta organização em fases permite planejar o projeto, incluindo marcos importante para controle e revisões.

O ciclo de vida dos projetos pode estar predefinido na organização, ou seja, a organização pode definir que todos os projetos adotem o mesmo ciclo de vida. A organização pode ainda predefinir mais de um modelo de ciclo de vida para os projetos, onde neste caso, para cada projeto a empresa irá selecionar aquele que melhor atender à suas características (SOFTEX, 2009).

2.3.1 Avaliação da GPR3

O ciclo de vida do Scrum é constituído por etapas bem definidas (Alliance, 2009). Cada etapa do ciclo fornece a etapa seguinte informações e artefatos necessários para a continuidade do ciclo além de disponibilizar pontos de controle sobre o projeto. Desta forma o Scrum atende este requisito (SANTOS; SANTOS, 2010).

2.4 GPR4 – 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

As estimativas de custo e esforço são geralmente baseadas resultados de análise utilizando modelos e dados históricos aplicados ao tamanho, atividades e outros parâmetros de planejamento. Os dados históricos incluem os dados de custo, esforço e tempo de projetos anteriores.

As estimativas de esforço e custo consideram o escopo, produtos de trabalho e as tarefas estimadas para o projeto; os riscos; mudanças já previstas; o ciclo de vida do projeto; dentre outros. Empresas que estejam implementando o nível G do MPS.BR geralmente não possuem bases de dados históricas, mas para alcançar os níveis superiores de maturidade é necessário que esta base seja construída e os dados obtidos pelos projetos executados (SOFTEX, 2009).

2.4.1 Avaliação da GPR4

As estimativas são divididas em duas etapas pelo Scrum, onde a primeira é realizada no product backlog. A segunda etapa é feita no sprint backlog e são mais precisas do que as primeiras onde a equipa usa como base o desempenho em sprints anteriores, complexidade das tarefas e capacidade para o próximo sprint. O custo e formas de armazenar informações de estimativas de esforço não são mencionados no Scrum.

O nível G do MPS.BR não obriga o registro em bases históricas, desta forma o Scrum atende parcialmente este requisito, pois se trata de estimativa de esforço e não de custo para os produtos de trabalho (SANTOS; SANTOS, 2010).

2.5 GPR5 – O ORÇAMENTO E O CRONOGRAMA DO PROJETO, INCLUINDO MARCOS E/OU PONTOS DE CONTROLE, SÃO ESTABELECIDOS E MANTIDOS

Para este resultado esperado do requisito as dependências entre os itens ou tarefas são estabelecidas e os potenciais gargalos são identificados utilizando-se métodos apropriados. Os gargalos são resolvidos quando possível e a duração do cronograma de atividades é estabelecida. O orçamento do projeto é definido com base no cronograma e na estimativa de custos (SOFTEX, 2009).

2.5.1 Avaliação da GPR5

No Scrum, o cronograma é obtido através do product backlog onde este é priorizado de acordo com as dependências entre os itens backlog e subdividido em sprints considerando alocação e capacidade de produção da equipe. O cronograma é definido através do número de sprints, onde o tempo médio de cada um varia em 30 dias. Os pontos de controle são as datas finais de cada sprint onde ocorre a reunião de revisão.

Com relação ao orçamento do projeto, o Scrum não contempla, pois para isto, seria necessária a realização de estimativa de custo das tarefas, o que não ocorre. Desta forma define-se que o Scrum atende parcialmente este requisito (SANTOS; SANTOS, 2010).

2.6 GPR6 – OS RISCOS DO PROJETO SÃO IDENTIFICADOS E O SEU IMPACTO, PROBABILIDADE DE OCORRÊNCIA E PRIORIDADE DE TRATAMENTO SÃO DETERMINADOS E DOCUMENTADOS

Para este requisito, os riscos do projeto devem ser registrados, assim como o acompanhamento de seus estados e as ações tomadas. Uma planilha de riscos com alguns dados importantes pode ser utilizada para identificação, monitoração dos riscos encontrados e atualização da lista a medida que novos riscos forem sendo encontrados.

No nível G, os riscos são acompanhados para verificar como os mesmos afetam o projeto para tomar ações, mesmo ainda sem um gerenciamento completo (SOFTEX, 2009).

2.6.1 Avaliação da GPR6

A identificação dos riscos são feitas de forma iterativa, durante as reuniões diárias e documentadas em whiteboards, flipcharts e na lista de impedimentos. Os riscos são acompanhados pelo Scrum Master. Conclui-se que o Scrum atende este requisito (SANTOS; SANTOS, 2010).

2.7 GPR7 – OS RECURSOS HUMANOS PARA O PROJETO SÃO PLANEJADOS CONSIDERANDO O PERFIL E O CONHECIMENTO NECESSÁRIOS PARA EXECUTÁ-LO

O planejamento de recursos humanos define funções, responsabilidades e relações hierárquicas do projeto. Este resultado esperado possui dois pontos principais que são o planejamento prévio das necessidades de pessoal em relação a competências e a alocação dos recursos humanos ao projeto de acordo com o planejamento realizado. Assim, este resultado esperado define que o planejamento de recursos humanos toma como base a análise das competências possuídas e requeridas para desempenhar as tarefas no projeto. Caso uma pessoa seja alocada ao projeto sem ter as competências necessárias, este risco pode ser minimizado com treinamentos e mentoring ou supervisão do trabalho por um membro da equipe mais capacitado (SOFTEX, 2009).

2.7.1 Avaliação da GPR7

A alocação da equipe é realizada pelo Scrum Master no início do projeto durante a fase de staging. O Scrum não descreve práticas para seleção de membros da equipe, e por falta de registro define-se que o Scrum atende parcialmente este requisito (SANTOS; SANTOS, 2010).

2.8 GPR8 – AS TAREFAS, OS RECURSOS E O AMBIENTE DE TRABALHO NECESSÁRIO PARA EXECUTAR O PROJETO SÃO PLANEJADOS

Este requisito faz referência a necessidade de se planejar, com base na EAP, as tarefas e previstos os recursos e o ambiente necessários, como por exemplo, equipamentos, ferramentas, serviços, etc. Todos os recursos precisam estar planejados e caso não aja a necessidade de nenhum recurso para ser adquirido ao projeto deve-se ao menos registrar que a questão foi analisada (SOFTEX, 2009).

2.8.1 Avaliação da GPR8

A infra-estrutura do projeto incluindo os recursos como máquinas; ferramentas; e demais investimentos é feita no início do projeto na fase de staging e estes por sua vez são

adicionados ao product backlog. O Scrum Master é o responsável por prover os recursos ao longo do projeto de acordo com os impedimentos e necessidades reportados nas reuniões. Conclui-se que o Scrum contempla este requisito (SANTOS; SANTOS, 2010).

2.9 GPR9 – OS DADOS RELEVANTES DO PROJETO SÃO IDENTIFICADOS E PLANEJADOS QUANTO A FORMA DE COLETA, ARMAZENAMENTO E DISTRIBUIÇÃO. UM MECANISMO É ESTABELECIDO PARA ACESSÁ-LOS, INCLUINDO, SE PERTINENTE, QUESTÕES DE PRIVACIDADE E SEGURANÇA

Neste requisito, os dados do projeto são as várias formas de documentação exigidas para sua execução, como por exemplo, relatórios; dados informais; estudos; etc. Estes dados podem se encontrar qualquer tipo de formatação e existir em qualquer meio como impressos ou desenhos em diversos materiais; fotografias; meio eletrônico; e multimídia (SOFTEX, 2009).

2.9.1 Avaliação da GPR9

No Scrum, os dados relevantes são: Product Backlog, Sprint Backlog; Impediment Backlog e gráfico Burndown. Estes itens são definidos em etapas específicas do projeto e de forma planejada. Todos os membros da equipe têm acesso a estes itens de forma a manter todos os integrantes da equipe no mesmo nível de ciência do progresso do projeto. Desta forma, o Scrum atende este requisito (SANTOS; SANTOS, 2010).

2.10 GPR10 – PLANOS PARA A EXECUÇÃO DO PROJETO SÃO ESTABELECIDOS E REUNIDOS NO PLANO DO PROJETO

A realização do planejamento do projeto é garantida pelos resultados esperados no escopo do nível G, que prevê a criação do cronograma de atividades, o planejamento de recursos humanos, custos, riscos, dentre outros. A reunião destes documentos é entendida como sendo o Plano de Projeto, onde as tarefas do processo definido para o projeto também podem fazer parte deste Plano (SOFTEX, 2009).

2.10.1 Avaliação da GPR10

O Product Backlog é constituído também pelos requisitos e recursos necessários para a infra-estrutura do projeto. Os riscos são identificados e registrados no Impediment Backlog, que por sua vez não é integrado ao Product Backlog. O Scrum gera um cronograma das atividades presentes no Product Backlog, mas os outros documentos resultados de planejamentos não são gerados pelo Scrum. O Scrum não define práticas para geração do Plano de Projeto. Desta forma, o Scrum não contempla este requisito (SANTOS; SANTOS, 2010).

2.11 GPR11 – A VIABILIDADE DE ATINGIR AS METAS DO PROJETO, CONSIDERANDO AS RESTRIÇÕES E OS RECURSOS DISPONÍVEIS, É AVALIADA. SE NECESSÁRIOS, OS AJUSTES SÃO REALIZADOS

No início do projeto, a partir da visão geral dos objetivos e características dos resultados pretendidos, dos recursos financeiros, técnicos e humanos, pode ser conduzida uma avaliação preliminar, onde à medida que o projeto evolui a viabilidade de sucesso pode ser avaliada mais precisamente. As mudanças nos requisitos podem levar a necessidade de reavaliação do projeto. Em marcos do projeto e durante as atividades de planejamento, pode ser necessária a confirmação da continuidade e viabilidade do projeto (SOFTEX, 2009).

2.11.1 Avaliação da GPR11

No Scrum as avaliações do projeto são feitas durante as reuniões diárias ou daily scrum meetings. Nesta reunião, a equipe relata o que foi feito desde a última reunião e o que será feito até a próxima reunião além de informar a causa para os atrasos. Com base nestes relatos o Scrum Master toma as medidas necessárias para contornar os problemas. Há também reuniões de planejamento de sprints, revisões e retrospectiva, onde em intervalos pequenos de tempo a viabilidade do projeto é reavaliada, levando em consideração mudanças nos requisitos e com a presença do cliente nestas reuniões. Define-se que o Scrum atende este requisito (SANTOS; SANTOS, 2010).

2.12 GPR12 – O PLANO DO PROJETO É REVISADO COM TODOS OS INTERESSADOS E O COMPROMISSO COM ELE É ESTABELECIDO

O compromisso pode envolver todos os interessados envolvidos no projeto, onde estes deverão ter a confiança de que o trabalho pode ser executado dentro do cronograma, desempenho e custos. Algumas empresas costumam realizar uma reunião de início de projeto que pode ser utilizada para resolver os conflitos e obter o comprometimento de seus membros (SOFTEX, 2009).

2.12.1 Avaliação da GPR12

No Scrum, a cada reunião de planejamento do Sprint é realizada uma revisão do projeto com o cliente considerando os requisitos restantes no Product Backlog, onde nesta o cliente seleciona os itens para o próximo sprint e o compromisso com o novo sprint é assumido por todos. Há uma interação intensa entre o cliente e a equipe nas reuniões chamadas de daily scrum meeting, sprint planning e sprint review meeting. O Scrum não produz o Plano de Projeto solicitado na GPR10, desta forma concluí-se que o mesmo atende parcialmente este requisito (SANTOS; SANTOS, 2010).

2.13 GPR13 – O PROGRESSO DO PROJETO É MONITORADO COM RELAÇÃO AO ESTABELECIDO NO PLANO DE PROJETO E OS RESULTADOS SÃO DOCUMENTADOS

Para este requisito, os resultados e critérios de conclusão das tarefas são analisados, as entregas, aderência ao cronograma e o dispêndio de esforços são examinados, assim como a utilização dos recursos.

O acompanhamento do projeto pode ser realizado utilizando-se ferramentas de planejamento, onde se pode verificar a relação entre previsto e realizado. O acompanhamento também pode ser feito por meio de reuniões e comunicação pessoal. É importante a existência de registros destes acompanhamentos (SOFTEX, 2009).

2.13.1 Avaliação da GPR13

No Scrum, os marcos são representados através dos sprints. O Scrum monitora diariamente através das reuniões diárias. Há também monitoramento ao final de cada sprint, onde a equipe demonstra ao Product Owner o trabalho realizado. O Scrum utiliza o gráfico burndown para visualizar e registrar o acompanhamento do projeto, mas este por sua vez não produz o Plano de Projeto como solicitado na GPR10, por este motivo define-se que o mesmo atende parcialmente este requisito.

2.14 GPR14 – O ENVOLVIMENTO DAS PARTES INTERESSADAS NO PROJETO É GERENCIADO

Devem-se identificar os interessados no projeto e em que fases os mesmos estarão envolvidos. Uma vez identificado e planejado o envolvimento, este deverá ser seguido. Os interessados no projeto podem incluir o cliente e o usuário, a direção da empresa e os membros do projeto. Em projetos pequenos estas atividades podem ser diminuídas devido ao pequeno número de interessados e a pouca comunicação necessária em virtude do curto prazo do projeto (SOFTEX, 2009).

2.14.1 Avaliação da GPR14

No Scrum, o cliente está presente no dia-a-dia do projeto. O planejamento do envolvimento e participação dos interessados em fases específicas no projeto é realizado. No início do projeto o Product Owner participa da reunião para criação do Product Backlog e participa também nas reuniões de criação dos sprints e de revisões. Nesta metodologia ágil as etapas e participação dos interessados estão bem definidas, desta forma defini-se que o Scrum atende este requisito (SANTOS; SANTOS, 2010).

2.15 GPR15 – REVISÕES SÃO REALIZADAS EM MARCOS DO PROJETO CONFORME ESTABELECIDO NO PLANEJAMENTO

Revisões em marcos do projeto não devem ser confundidas com o acompanhamento descrito na GPR13, no qual se diz ao acompanhamento no dia-a-dia do

projeto. Os marcos precisam ser previamente definidos ao se realizar o planejamento do projeto (SOFTEX, 2009).

2.15.1 Avaliação da GPR15

No processo do Scrum, ao final de cada sprint é realizada uma reunião de revisão que objetiva avaliar os progressos ou atrasos no projeto em relação ao que foi definido no planejamento do sprint.

Um sprint tem duração de duas a quatro semanas, ficando a encargo do Scrum Master avaliar e definir a duração de um sprint, representando os pontos para as revisões do projeto. O Scrum atende este requisito.

2.16 GPR16 – REGISTROS DE PROBLEMAS IDENTIFICADOS E O RESULTADO DA ANÁLISE DE QUESTÕES PERTINENTES, INCLUINDO DEPENDÊNCIAS CRÍTICAS, SÃO ESTABELECIDOS E TRATADOS COM AS PARTES INTERESSADAS

As atividades de revisão em marcos e de monitoramento respectivamente das GPR15 e GPR13, possibilitam a identificação de problemas nos projetos. Estes problemas devem ser analisados e registrados através do uso de ferramentas específicas, planilhas ou outros tipos de mecanismos de gerenciamento de problemas (SOFTEX, 2009).

2.16.1 Avaliação da GPR16

A identificação e análise dos problemas relativos ao projeto acontecem, no Scrum, nas reuniões de planejamento de sprints, reuniões diárias, reuniões de revisão de sprints e de retrospectiva. Em todas as reuniões há a presença do cliente e da equipe. O Scrum registra os problemas encontrados no impediment backlog, onde as ações e alterações a estes problemas são responsabilidade do Scrum Master. O Scrum atende este requisito (SANTOS; SANTOS, 2010).

2.17 GPR17 – 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

Como resultado do acompanhamento do projeto e revisões em marcos, problemas são identificados, analisados e registrados nas GPR13 e GPR16 respectivamente. As ações corretivas devem ser estabelecidas e gerenciadas até sua conclusão, de forma que resolvam os problemas que possam impedir o projeto de atingir seus objetivos se não resolvidos de forma adequada. Todo o controle dos problemas levantados, ações tomadas, os responsáveis pelas ações e os resultados devem ser registrados (SOFTEX, 2009).

2.17.1 Avaliação da GPR17

Nas reuniões diárias do Scrum busca-se levantar os impedimentos do projeto. Em seguida há uma identificação, tomada de ações e monitoramento destes impedimentos pelo Scrum Master, entretanto estas ações corretivas não são registradas. Concluí-se que o Scrum atende parcialmente este requisito (SANTOS; SANTOS, 2010).

APÊNDICE B - MAPEAMENTO DOS REQUISITOS DA GERÊNCIA DE PROJETOS DO NÍVEL E DO MPS.BR APLICANDO AS PRÁTICAS DO SCRUM

Mapeamento dos Requisitos da Gerência de Projetos do nível E do MPS.BR Aplicando as Práticas do Scrum

Anderson Ricardo dos S. Rodrigues¹, Ana Cláudia G. Barbosa²

¹Acadêmico do Curso de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – Brasil

²Professor MSc. do Curso de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – Brasil

anderson.srod@gmail.com, agb@unesc.net

Abstract. *Today, software companies seek for constant improvement in productivity and quality in the process of developing their products to achieve greater market competitiveness and provide better customer satisfaction. Providing a quality product and timely delivery defined with the customer remains a problem for organizations. Quality models and agile software development can be used by companies as a solution to the problem. While quality models define requirements and standards for the process, the agile development methodologies define values and practices that contribute to the development of a quality product. Given the importance of quality models and agile development methodologies this research was conducted in order to verify the relationship between them. In this case, mapping was performed or evaluation requirements of the Project Management level E of the Brazilian Software Process Improvement quality model applying the practices of Agile Scrum development methodology. To make it possible to perform the mapping, evaluation criteria were defined and posteriorly described and individually evaluated all the requirements of the Project Management level E applying the practices of Scrum. At the end of the evaluation, the analysis and conclusion are presented to see if with the Scrum agile methodology it is possible to meet the requirements of the Project Management level E of the Brazilian Software Process Improvement quality model.*

Resumo. *As empresas de software buscam nos dias atuais uma constante melhoria na produtividade e qualidade no processo de desenvolvimento de seus produtos a fim de obter maior competitividade no mercado e proporcionar uma melhor satisfação de seus clientes. Fornecer um produto com qualidade e entrega dentro do prazo definido com o cliente continua sendo um problema para as organizações. Modelos de qualidade e metodologias ágeis de desenvolvimento de software podem ser utilizadas pelas empresas como solução para tal problema. Enquanto os modelos de qualidade definem requisitos e normas para o processo, as*

metodologias ágeis de desenvolvimento definem valores e práticas que contribuem para o desenvolvimento de um produto de qualidade. Em virtude da importância dos modelos de qualidade e metodologias ágeis de desenvolvimento foi realizada esta pesquisa com o objetivo de verificar o relacionamento entre ambas. Neste caso, foi realizado mapeamento dos requisitos da Gerência de Projetos do nível E do modelo de qualidade Melhoria do Processo de Software Brasileiro aplicando-se as práticas da metodologia ágil de desenvolvimento Scrum. Para que fosse possível realizar o mapeamento foram definidos critérios de avaliação e posteriormente foram descritos e avaliados individualmente todos os requisitos da Gerência de projetos do nível E aplicando-se as práticas do Scrum. Ao final das avaliações é apresentada a análise das mesmas e a conclusão definindo se com a metodologia ágil Scrum é possível atender os requisitos da Gerência de Projetos do nível E do modelo de qualidade Melhoria do Processo de Software Brasileiro.

1. Introdução

Mesmo com a evolução das ferramentas e técnicas nos últimos anos, a produção de software confiável e correto entregue dentro do prazo estipulado ainda é muito difícil. Métodos ágeis para desenvolvimento de software são uma solução proposta para as metodologias de desenvolvimento de software tradicionais que são aplicados em situações em que os requisitos para a construção do sistema são estáveis e com evolução previsível. (SOARES, 2004).

Scrum é uma metodologia de desenvolvimento ágil cujo objetivo é fornecer um processo constituído por valores, práticas e técnicas para projeto e desenvolvimento software. A idéia principal do Scrum é que o desenvolvimento de softwares envolve muitos requisitos, recursos e tecnologia, que podem mudar durante o processo de desenvolvimento. Isto torna o processo de desenvolvimento imprevisível e complexo, requerendo flexibilidade para acompanhar as mudanças (SOARES, 2004).

As empresas têm como um de seus principais objetivos fornecer um produto com qualidade a seus clientes, a fim de obter competitividade no mercado. Para as empresas de software, alcançar esta competitividade pela qualidade implica tanto na melhoria da qualidade de seus produtos e serviços quanto nos métodos de produção e distribuição do software. A Melhoria de Processo de Software (MPS) é um modelo de melhoria a avaliação do processo de software destinado preferencialmente a pequenas e médias empresas. A Melhoria do Processo de Software Brasileiro (MPS.BR) é um projeto que visa a melhoria do processo de software do país e sua novidade está na estratégia de implementação, criada para a realidade brasileira. A base técnica utilizada para a construção deste modelo de qualidade foram as normas ISO/IEC 12207, ISO/IEC 15504 além do conteúdo de outros modelos de referência como Capability Maturity Model Integration (CMMI) (WEBER, 2005).

O modelo MPS possui três componentes: Modelo de referência para Melhoria do Processo de Software (MR-MPS), Método de avaliação para Melhoria do Processo de Software (MA-MPS) e Modelo de negócio para Melhoria do Processo de Software (MN-MPS). No MR-MPS estão definidos os sete níveis de maturidade representados por letras de A até G no qual A é o nível mais alto e G o mais baixo (SOFTEX, 2009).

2. Engenharia de Software

A engenharia de software pode ser conceituada como uma metodologia de desenvolvimento e manutenção de software, com as seguintes características: roteiro dinâmico de soluções tecnológicas; adequação aos requisitos funcionais do cliente e seus respectivos processos; efetivação de padrões de qualidade; produtividade; fundamentação na tecnologia da

Informação disponível, viável, oportuna e customizada; planejamento e gestão de atividades, recursos, custos e prazos (REZENDE, 2005).

A engenharia de software abrange um conjunto de três elementos – métodos, ferramentas e procedimentos – que possibilita ao gerente formas para a construção de um software de alta qualidade e produtividade.

Pode-se concluir então que engenharia de software é uma metodologia ou um roteiro composto de diversas técnicas que são aplicadas em suas diversas fases de construção de um sistema. A seguir serão conceituadas e definidas as fases de desenvolvimento de software.

2.1. Fase de Definição

Esta fase focaliza o *o que*, ou seja, durante a definição o analista procura verificar quais informações deverão ser processadas e armazenadas pelo sistema, define a interface, quais são as restrições existentes no projeto e quais critérios de validação são exigidos para se definir um sistema bem-sucedido. Durante este processo, dependendo do padrão de definição aplicado, três etapas Análise do Sistema, Planejamento do Projeto de Software e Análise de Requisitos ocorrerão de alguma forma (PRESSMAN, 1995).

Conclui-se então que a fase de definição procura se aprofundar e planejar toda a infra-estrutura do sistema, realizando a análise do sistema e definindo os requisitos para que o mesmo atenda as necessidades dos usuários ou clientes.

2.2. Fase de Desenvolvimento

Nesta fase, o desenvolvedor focaliza-se em como a estrutura de dados e a arquitetura de software tem de ser projetados, como o projeto será traduzido numa linguagem de programação e como os testes têm de ser realizados. A fase de desenvolvimento ocorre em três passos: Projeto de software, Codificação e Realização de testes de software. (PRESSMAN, 1995).

2.3. Fase de Manutenção

Um software sofre manutenções, sejam estas por simples correções, alteração legal ou por estar gerando problemas para o cliente. Segundo Rezende (2005, p. 48) “a manutenção de software existente demanda mais de 70% de todo o esforço despendido por uma organização (ou departamento) de software...”.

As alterações nos sistemas são inevitáveis, e para isto deve-se desenvolver mecanismos e processos para avaliar, controlar e fazer modificações, afim de reduzir o esforço despendido em manutenção.

3. Projeto MPS.BR

Nesta seção será conceituado o projeto MPS.BR e o nível E. Também serão descritas suas bases de construção que são os modelos MPS e CMMI.

3.1. CMMI

O Capability Maturity Model for Software (CMM) é um modelo para melhoria dos processos de software criado em 1991, porém para as empresas existia-se a dificuldade da aplicação conjunta de diferentes modelos.

O Projeto CMM Integration (CMMI) foi criado para integrar diferentes modelos (DONEGAN et al, 2005) onde o objetivo do CMMI Product Team era a combinação de três modelos originais: Capability Maturity Model for Software (SW-CMM); Eletronic Industries Alliance Íterim Standard (EIA/IS); Integrated Product Development Capability Maturity Model (IPD-CMM) em um framework de melhoramento simplificado para empresas que perseguem ampla iniciativa de melhoria de processos (ATAÍDES, 2006).

Além do desenvolvimento de um framework comum para suportar a integração de outras disciplinas específicas do modelo CMMI, a equipe procurou garantir que todos os produtos desenvolvidos fossem consistentes e compatíveis com o ISO/IEC 15504 (ATAÍDES, 2006).

3.2. MPS

A Melhoria do Processo de Software (MPS) é um modelo de melhoria e avaliação de processo de software, voltado preferencialmente a pequenas e médias empresas, normalmente grupos de empresas em um Modelo de Negócio Cooperado (MNC), mas também vem sendo aplicado em organizações de grande porte no Modelo de Negócio Específico (MNE) (SOFTEX, 2009).

O modelo define também regras para sua implementação e avaliação. Como mostra a figura 2, o MPS possui três componentes: Modelo de Referência para Melhoria do Processo de Software (MR-MPS), Método de Avaliação para Melhoria de Processo de Software (MA-MPS) e Modelo de Negócio para Melhoria de Processo de Software (MN-MPS) (SOFTEX, 2009).

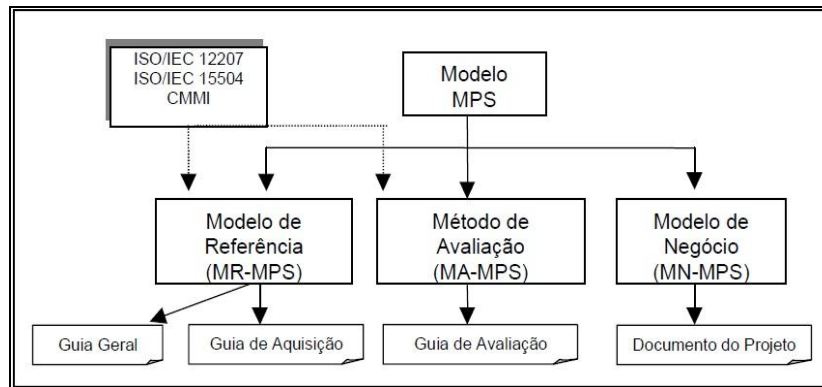


Figura 1. Documentos do modelo de qualidade MPS.BR

No Modelo de Referência para Melhoria de Processo de Software (MR-MPS) estão descritos os requisitos que as empresas precisam atender para estar em conformidade com o modelo MPS (SOFTEX, 2009). Este por sua vez é composto por sete níveis de maturidade (A, B, C, D, E, F e G) que são seqüenciais e cumulativos. Cada um destes níveis é composto por um conjunto de processos em um determinado nível de capacidade onde o progresso e o atendimento de cada nível de maturidade é obtido somente quando são atendidos todos os requisitos e atributos relacionados àquele nível (SANTANA, TIMÓTEO, VASCONCELOS, 2006).

3.2.1. Nível E

O nível E é o terceiro nível de maturidade do MPS.BR e tem como foco principal a padronização dos processos da organização por meio da definição de processos padrão que devem ser definidos através das melhores praticas e processos existentes na organização, que consiste no primeiro passo de uma avaliação contínua e melhoria dos processos.

No nível E, há a evolução dos requisitos descritos na GPR. Além da GPR há os processos: Definição do Processo Organizacional; Avaliação e Melhoria do Processo Organizacional e Medição. Estes, apesar da relação com a GPR não serão descritos, pois não condiz com o objetivo deste trabalho que foca na descrição e análise apenas dos requisitos da GPR (SOFTEX, 2011).

O nível E possui os seguintes requisitos: GPR4, GPR8, GPR20, GPR21 e GPR22.

4. Scrum

O Scrum foi criado em 1996 por Ken Schwaber e Jeff Sutherland e reúne atividades de monitoramento e feedback com reuniões rápidas e diárias com toda a equipe afim de identificar problemas e correções de impedimentos encontrados durante o processo de desenvolvimento (SOARES et al, 2007).

A abordagem do Scrum é oposta ao modelo cascata sendo o processo iterativo onde se inicia a análise com o levantamento de alguns requisitos disponíveis, assim que alguma análise tiver sido feita já é iniciado o processo de desenvolvimento. Sendo assim, pode-se dizer que o projeto trabalha em um pequeno pedaço de cada vez (MARTINS, 2007).

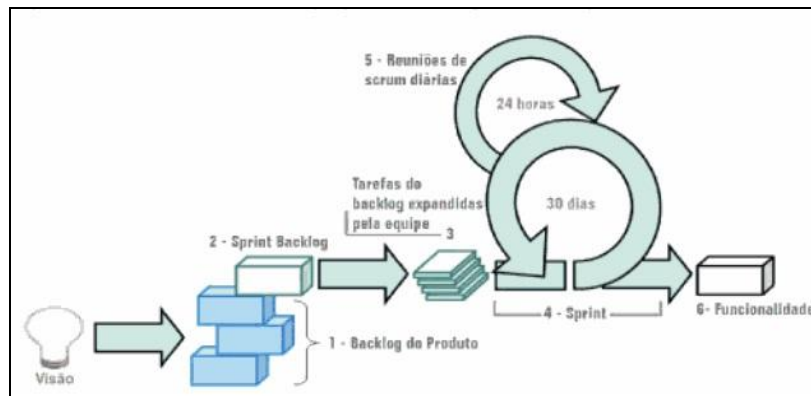


Figura 2. Processo do Scrum

No Scrum há o Product Backlog que é uma lista inicial de requisitos que pode ser ajustada pelo Dono do Produto (Product Owner) durante o projeto a fim de mudar a seqüência dos itens que serão trabalhados. Há também o Sprint Backlog que é um conjunto de atividades de desenvolvimento conduzidas em um período pré-definido de tempo chamado time box (caixa de tempo), que pode variar de uma a quatro semanas ou mais. Este intervalo de tempo é definido de acordo com os riscos, complexidade do produto e grau de volatilidade dos requisitos (MARTINS, 2007).

5. Mapeamento das Práticas do Scrum em relação aos Requisitos referente à Gerência de Projetos do Nível E do MPS.BR

Nesta seção serão definidos os critérios de avaliação de conformidade dos requisitos descritos na GPR do nível E do MPS.BR. Posteriormente serão definidos e avaliados individualmente os requisitos GPR4 (evolução do nível G), GPR8, GPR20, GPR21 e GPR22.

6.1. Critérios de Avaliação dos Requisitos

Foram definidos três critérios para classificação da metodologia ágil em relação aos requisitos da Gerência de Projetos do nível E do MPS.BR. Os critérios são: Satisfeito, Parcialmente Satisfeito e Não Satisfeito.

6.2. Definição da GPR4

Para atender a este requisito, a realização das estimativas e do planejamento do projeto é baseada no processo definido para o projeto, nos ativos de processo e no repositório de medidas da organização onde os ativos de processo e repositório de medidas da organização estão definidos no escopo do processo Definição do Processo Organizacional.

Para a realização das estimativas no nível E, ao contrário do nível G, é necessária a construção de bases históricas. Bases históricas consistentes são baseadas nos projetos já realizados pela organização e além de possuírem o objetivo de facilitar as estimativas também proporcionam um melhor entendimento do comportamento dos projetos da organização (SOFTEX, 2011).

6.2.1. Avaliação da GPR4

Assim como já descrito na definição do requisito em questão, diferente do nível G, no nível E é necessário o registro de bases históricas para que se possam construir as estimativas. Conforme Santos (2010) no nível G não é exigido o registro de bases históricas onde devido ao fato de o Scrum não mencionar formas para armazenar estimativas de esforço, o Scrum atende parcialmente o requisito.

O Scrum é um framework onde no mesmo podem-se empregar diversos processos e técnicas para o desenvolvimento de produtos complexos (SANTOS; SANTOS, 2010). Com isto para que se tenha possibilidade de satisfação desta condição, a organização pode encontrar uma forma própria ou buscar em outras fontes métodos adequados para armazenar as informações de custo e esforço dos Sprints ou iterações do Scrum.

Com a armazenagem destes dados abre-se a possibilidade da construção de uma base histórica para que se possa ser feito o planejamento e estimativa nas próximas iterações do processo. Conclui-se que, para o nível E, o Scrum não atende o requisito pelo mesmo fato de que o mesmo não menciona formas de armazenagem de informações de esforço fazendo com que para a construção de bases históricas sejam buscados outros métodos fora da metodologia ágil.

6.3. Definição da GPR8

Para este requisito têm-se a necessidade para planejar as tarefas, com base na EAP (Estrutura Analítica do Projeto) ou estrutura equivalente, assim como prover os recursos e o ambiente necessários incluindo equipamentos, ferramentas, serviços, componentes e requisitos do processo. A partir do nível E, o planejamento do ambiente de trabalho para o projeto deve estar compatível com o ambiente padrão de trabalho da organização onde este se encontra definido na DFP7 do processo de Definição do Processo Organizacional.

Todos os recursos precisam estar planejados, mesmo os já disponíveis ou que serão compartilhados com outros projetos, uma vez que se trata de sua alocação para uso. As adaptações ao planejamento do ambiente também precisam estar documentadas (SOFTEX, 2011).

6.3.1. Avaliação da GPR8

Muitos projetos possuem uma lista de requisitos como tipo de sistema, itens de planejamento, tipo de aplicação, ambiente de desenvolvimento, etc. No Scrum, estes requisitos são definidos no Backlog de Produto que nada mais é do que uma lista de requisitos definidos pelo Dono do Produto onde esta pode ser alterada ao longo do projeto (MARTINS, 2007).

Ao longo do projeto o Scrum Master é o responsável por prover os recursos necessários conforme as necessidades e impedimentos que são reportados nas reuniões (SANTOS; SANTOS, 2010). Como no Scrum há uma definição de como os requisitos necessários ao ambiente de desenvolvimento devem ser documentados no planejamento do projeto, conclui-se que assim como no nível G o Scrum satisfaz este requisito para o nível E.

6.4. Definição da GPR20

Neste o resultado esperado é o estabelecimento das equipes de acordo com as regras para estruturação, formação e atuação das equipes na empresa. As regras, por sua vez, estão definidas na DFP8 do processo Definição do Processo Organizacional.

Na DFP8 do processo Definição do Processo Organizacional define que a estruturação das equipes está relacionada ao número, tipo de equipes e como elas se relacionam. Quanto à formação das equipes trata-se da definição de um organograma, definição de membros e líderes e a alocação de recursos necessários. Por fim, a atuação define como as equipes são formadas e como elas interagem para o alcance de seus objetivos incluindo a compreensão pelos participantes sobre a contribuição dos padrões de trabalho e participação na equipe (SOFTEX, 2011).

6.4.1. Avaliação da GPR20

Verificando o quesito estruturação das equipes, o Scrum não descreve número de equipes, tipo e nem relacionamento entre as mesmas. Porém, o Scrum possui seus papéis e funções bem definidas atendendo aos quesitos de formação e atuação das equipes, porém não descreve nada a respeito da estruturação das mesmas. Desta forma, entende-se que o Scrum atende parcialmente este requisito.

6.5. Definição da GPR21

Para este requisito é necessário que haja a identificação das experiências obtidas na execução do projeto que possam contribuir para o conjunto de ativos de processo organizacionais. As experiências obtidas incluem, mas não se limitam a, medidas coletadas, dados para refinamento de estimativas, lições aprendidas, boas práticas, etc.

Referente ao repositório de medidas, o mesmo deve ser constantemente atualizado seguindo os resultados esperados do processo de Medição. Com a implementação dos resultados esperados no nível E, os dados do repositório tendem a ficar mais organizados, onde ao seguirem um mesmo padrão as informações coletadas nos projetos devem ter uma estrutura padronizada (SOFTEX, 2011).

6.5.1. Avaliação da GPR21

No Scrum, o mesmo não menciona a respeito de armazenagem ou a criação e manutenção de um repositório de medidas. Quando a equipe consegue entregar tudo no prazo correto, isto é resultado do empenho e dedicação e não da precisão das estimativas. A imprecisão gera preocupações onde algumas empresas tentam melhorar a precisão das suas estimativas

construindo um banco de dados dos projetos com as estimativas e as horas reais utilizadas, e derivas as variações (MARTINS, 2007).

Com o que foi descrito anteriormente, justificando-se o fato de que o Scrum não define formas de criação e manutenção de um repositório de medidas conclui-se que o mesmo não contempla este requisito.

6.6. Definição da GPR22

Para este resultado, deve-se definir um processo a partir de um conjunto de processos padrão da empresa, onde o conjunto de processos padrão da organização define-se em um produto do processo Definição do Processo Organizacional, assim como as estratégias para adaptação deste processo e os modelos de ciclo de vida que devem ser utilizados nos projetos.

O processo definido para o projeto deve abranger todas as atividades para o desenvolvimento e manutenção do produto a partir do que é pedido pelo o cliente até a entrega do produto (SOFTEX, 2011).

6.6.1. Avaliação da GPR22

Como descrito na definição do resultado esperado, este por sua vez consiste na definição de um processo para o projeto que engloba todas as atividades para o desenvolvimento e manutenção do produto, para isto, pode-se adotar o Scrum como um processo padrão para o desenvolvimento de software organização. Conclui-se por sua vez que o Scrum atende a este requisito.

7. Análise das Avaliações

Anteriormente foram definidos e avaliados os requisitos da GPR do nível E com o Scrum. Nesta seção serão apresentados os gráficos evidenciando a conformidade dos requisitos para o nível E e o cruzamento destas informações com as avaliadas para o nível G do MPS.BR.

No mapeamento dos requisitos do nível G chegou-se a uma conclusão de que o Scrum satisfaz 52,94% dos requisitos solicitados pela GPR. A figura 8 demonstra esta avaliação (SANTOS; SANTOS, 2010).

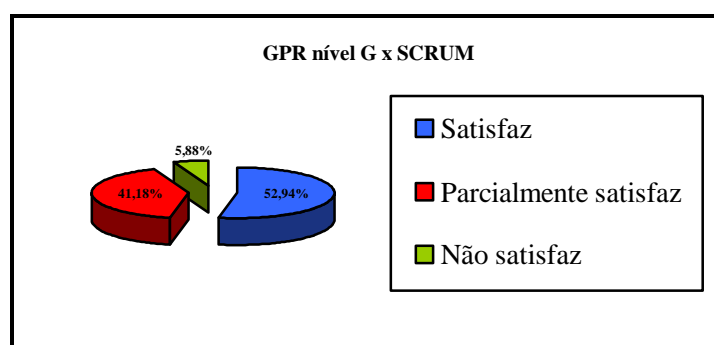


Figura 3. Resultado da análise do mapeamento dos requisitos do nível G

Analisando somente os requisitos do nível E percebe-se que o mesmo satisfaz apenas 40% dos requisitos, onde do restante atende parcialmente 20% e não atende 40%.

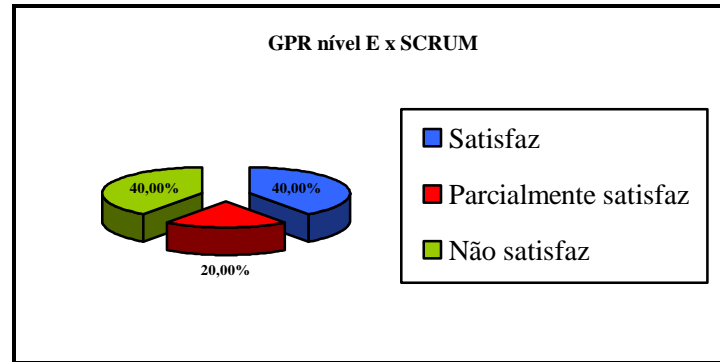


Figura 4. Resultado da análise do mapeamento dos requisitos do nível E

Cruzando as informações das avaliações da GPR dos níveis G e E verificamos que o Scrum contempla apenas 50% dos requisitos. O Scrum atende ainda parcialmente 36,37% dos requisitos e não atende 13,63% dos requisitos.

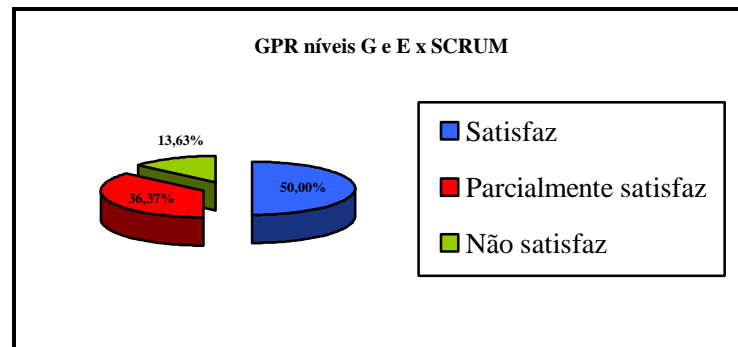


Figura 5. Cruzamento dos resultados das análises níveis G e E do MPS.BR

As informações utilizadas para o cruzamento das análises foram obtidas no trabalho feito por SANTOS (2010) que avalia os requisitos da GPR do nível G. Conforme verificado nesta seção foram apresentadas as análise dos mapeamentos dos requisitos da GPR do nível E do MPS.BR.

7. Conclusão

Neste artigo foram apresentados os mapeamentos ou análises de satisfação dos requisitos descritos no guia de implementação do MPS.BR entre o Scrum e a GPR do nível E. Deve-se levar em consideração que as análises feitas são totalmente passíveis de diferentes interpretações, pois as metodologias ágeis podem sofrer alterações e possuir interpretações, que podem afetar na compreensão e análise dos requisitos.

Pode-se verificar também que ao cruzar as informações das análises do nível G com o nível E o Scrum atende a apenas 50% do total de vinte e dois requisitos. O Scrum não satisfaz a apenas 13,63% dos requisitos.

Para os requisitos com satisfação parcial, deve-se levar em consideração o fato de que o Scrum por ser uma metodologia ágil dispensa a parte de documentação e tem foco na execução e planejamento das tarefas de forma dinâmica com o objetivo de agilizar o desenvolvimento de software. Considera-se também, como exemplo, na GPR20 o Scrum define os papéis dos integrantes das equipes, porém não menciona informações a respeito de número de equipes e nem de relacionamento entre ambas. Fatores como falta de documentação e incompletude de informações contribuíram para que alguns requisitos fossem classificados como parcialmente satisfeitos.

Sugere-se, que como as metodologias ágeis são dinâmicas e de fácil adaptação, as organizações procurem meios próprios ou externos para solução dos problemas de não conformidade ou conformidade parcial de requisitos, já que as mesmas dão possibilidades de customizações a seus adeptos.

Referências

ATAÍDES, Adriana da Costa. **UM MÉTODO PARA ACOMPANHAMENTO E CONTROLE DA IMPLANTAÇÃO DO CMMI**. 2006. 125 f. Dissertação (Mestrado) - Universidade de Brasília, Brasília, 2006. Disponível em: <<http://repositorio.bce.unb.br/bitstream/10482/6733/1/ADRIANA%20DA%20COSTA%20ATA%20C3%8DDES.pdf>>. Acesso em: 10 out. 2011.

MARTINS, José Carlos Cordeiro. **Técnicas para Gerenciamento de Projetos de Software**. Rio de Janeiro: Brasport, 2007. 427 p.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995. 1056 p. ISBN 8534602379

Número de Chamada: 005.1 P935e 1995

REZENDE, Denis Alcides. **Engenharia de Software e Sistemas de Informação**. 3° Rio de Janeiro: Brasport, 2005. 313 p.

SANTANA, Célio A.; TIMÓTEO, Aline L.; VASCONCELOS, Alexandre 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: <<http://www.lbd.dcc.ufmg.br/colecoes/sbqs/2006/009.pdf>>. Acesso em: 15 out. 2011.

SANTOS, Claudio A. B.; SANTOS, José Amancio Macedo. **Mapeamento das práticas do Scrum em relação aos requisitos referente a Gerência de Projetos do nível G do MPS.BR**. Feira de Santana - Ba: 2010. Disponível em: <<http://www.cesmac.com.br/erbase2010/papers/wticg/65120.pdf>>. Acesso em: 05 abr. 2012.

SOARES, Felipe S. Furtado et al. **Adoção de SCRUM em uma Fábrica de Desenvolvimento Distribuído de Software**. Recife: 2007. Disponível em: <<http://200.17.137.110:8080/schisto/publicacoes/i-wdds/adocao-de-scrum-em-uma-fabrica-de-desenvolvimento.pdf>>. Acesso em: 20 out. 2011.

SOFTEX (Org.). MPS.BR - Melhoria de Processo do Software Brasileiro: **Guia de implementação - Parte 1: Fundamentação para Implementação do Nível G do MR-MPS**. Campinas, 2009. 31 p. Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte_1_2009.pdf>. Acesso em: 23 jun. 2011.

SOFTEX (Org.). MPS.BR - Melhoria de Processo do Software Brasileiro: **Guia de implementação - Parte 1: Fundamentação para Implementação do Nível E do MR-MPS**. Campinas, 2011. 67 p. Disponível em:

<http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Implementacao_Parte_3_2011.pdf>. Acesso em: 04 jun. 2012.

WEBER, Kival Chaves et al. **Modelo de Referência e Método de Avaliação para Melhoria do Processo de Software - versão 1.0 (MR-MPS e MA-MPS)**. In: **IV, 4., 2005, Florianópolis**. Modelo de Referência e Método de Avaliação para Melhoria do Processo de Software - versão 1.0 (MR-MPS e MA-MPS). Florianópolis: Softex, 2005. p. 1 - 14. Disponível em: <http://golden.softex.br/portal/softexweb/uploadDocuments/MR_MA-MPS.pdf>. Acesso em: 26 jun. 2011.