



UNISUL

GUSTAVO GOULART COMELI

WILLIAN SPERANDIO

**MELHORIA DE PROCESSOS DE SOFTWARE BRASILEIRO:
RECOMENDAÇÕES PARA QUE SEJA ATINGIDO O NÍVEL G EM UMA
EMPRESA ESTUDO DE CASO**

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
da Universidade do Sul de Santa Catarina, como
requisito parcial à obtenção do título de Licenciado
em Ciência da Computação.

Orientador: Prof. Vera Rejane N. Schuhmacher Msc.

Palhoça

2010

GUSTAVO GOULART COMELI
WILLIAN SPERANDIO

MELHORIA DE PROCESSOS DE SOFTWARE BRASILEIRO
RECOMENDAÇÕES PARA QUE SEJA ATINGIDO O NÍVEL G EM UMA
EMPRESA ESTUDO DE CASO

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Licenciado em Ciências da Computação e aprovado em sua forma final pelo Curso de Graduação em Ciências da Computação da Universidade do Sul de Santa Catarina.

Palhoça, 25 de novembro de 2010.

Professor e Orientador: Vera Rejane N. Schuhmacher Msc.

Universidade do Sul de Santa Catarina

Prof. Maria Inés Castiñeira, Dra.
Universidade do Sul de Santa Catarina

Alexsandro de Farias, Analista de Sistemas.
ETS - Energia Transporte e Saneamento Ltda.

Dedicamos este projeto de conclusão de curso a todos que acreditam em nosso potencial, principalmente nossos pais, nossos maiores apoiadores.

AGRADECIMENTOS

Considerando este trabalho como resultado de uma longa jornada acadêmica, agradecer pode não ser tarefa fácil, nem justa. Para não correr o risco de cometer alguma injustiça, agradeço de antemão a todos que, de alguma forma, passaram pela minha vida e contribuíram para a construção de quem sou hoje. E agradeço, particularmente, a algumas pessoas pela contribuição direta ou indiretamente na construção deste.

Aos meus pais, Dilney Comeli e Maria Jaqueline Goulart Comeli, com quem pude contar sempre, em todas as circunstâncias da minha vida, e em especial neste último ano, o qual não poderia ser diferente, sempre ao meu lado, auxiliando, principalmente quando pensei em desistir, e me dando forças para continuar a caminhada. Muito obrigado por serem meus pais! Amo vocês!

A minha irmã Gabriela Goulart Comeli que sempre está disposta a colaborar de uma forma ou de outra, com quem também, sempre pude contar, principalmente neste último ano.

Aos amigos que conquistei durante a jornada acadêmica, especialmente aos que me acompanharam desde o início, Carol, Drica, Willian, Jonnes, Maluko, Lucas, Deh, Thiago e Giu. Foi muito gratificante tê-los como companheiros.

Ao meu companheiro de TCC, Willian que está nessa jornada comigo desde o começo do curso.

Aos amigos, Ely, Guiga, Bell e Ruy pelo incentivo, força e amizade que compartilhamos durante nossa caminhada. Com quem dividi a angústia das provas e a alegria das comemorações.

Em especial, ao meu compadre, grande amigo e companheiro Alexsandro de Farias. Pessoa que sempre me ajudou e incentivou a continuar na caminhada e não desistir do meu ideal, além de ser parceiros de projetos e companheiro de fórmula 1. Meu muito obrigado.

Aos professores dos Cursos de Ciência da Computação e Sistemas de Informação da Unisul – Universidade do Sul de Santa Catarina.

À professora Vera Rejane N. Schuhmacher, por sua disposição em ser minha orientadora na escolha do meu tema, por sua compreensão e dedicação e, pelas

contribuições teóricas que tornaram possível a conclusão deste Trabalho, além de uma grande amiga, uma pessoa com quem me identifiquei desde o primeiro dia de aula, uma grande pessoa de coração.

Aos professores Maria Inés Castineira e Ricardo Davalos, por todo o auxílio durante toda a faculdade.

A toda a equipe da ETS – Energia transporte e Saneamento Ltda., amigos que quero levar pela vida toda, principalmente Alexsandro, Viviani, Tatiana, Robson, Wagner, Aldori, Fabian, Luan, Vitor e Gildomar vulgo nego veio, tchê!!!

Gustavo Goulart Comeli

AGRADECIMENTOS

Eu tenho muito que agradecer, a começar pelo fato de estar vivo, e de poder compartilhar este trabalho com outros seres, iguais a mim, que gostam de se apoderarem dos conhecimentos herdados de nossos antepassados.

Aos meus PAIS Noir José Sperandio e Iete da Veiga Sperandio. Ambos serão responsáveis por cada sucesso obtido e cada degrau avançado pro resto da minha vida. Durante todos esses anos vocês foram pra mim um grande exemplo de força, de coragem, perseverança e energia infinita para nunca desistir diante do primeiro obstáculo encontrado. Vocês são e sempre serão meu maior porto seguro aqui embaixo, meu maior exemplo de vitória, meus heróis e simplesmente aqueles que mais amo. Obrigado por estarem sempre comigo. Obrigada simplesmente por participarem comigo durante essa caminhada, me ajudando a construir os alicerces de um futuro que começa agora, após cinco anos dedicados à uma paixão que surgiu na infância. Vocês me ensinaram direta e indiretamente lições pra toda uma vida.

Agradeço mais uma vez minha namorada por todos esses anos, de muito carinho, amizade, companheirismo e felicidade. Obrigado por tudo meu amor, você é demais!

Aos amigos que me acompanharam e me auxiliaram que nos deram um socorro sempre nas horas mais difíceis, se prontificando a nos ajudar e tirar as dúvidas, seja elas no trabalho na faculdade ou até mesmo dúvidas da “vida”.

À professora Vera Rejane N. Schumacher, por sua disposição em ser nossa orientadora, por sua compreensão, experiência e dedicação, estando presente sempre nas horas necessárias, sempre disposta a nos ajudar.

Aos professores que passaram por minha vida, que foram muitos, quero agradecer especialmente à, Prof^a. Maria Inés Castiñeira, com a qual sempre atenciosa e que nos acompanhou nesta longa jornada. Em fim a todos os professores que passamos por estes cinco anos, sem vocês seria impossível, obrigado!

Ao Gustavo, amigo, parceiro, irmão, que conheci na Universidade, e que desde o início desta caminhada estivemos juntos. Sempre, mas sempre, estive pronto a

ajudar sem esperar algo em troca. Uma pessoa que me ajudou, me ensinou e contribuiu em muito para a minha formação profissional, acadêmica e pessoal. Um verdadeiro amigo.

Quero muito agradecer também a todos os colegas de turma, especialmente aos que como eu chegaram até o fim, Carol, Drica, Gustavo, Jonnes, Deh, Thiago e Giu. Foi bom conviver com vocês, Muito obrigado!

E finalmente, agradeço a todos que me ajudaram direto ou indiretamente para o desenvolvimento deste projeto. Um MUITO OBRIGADO a todos vocês!

Willian Sperandio

“O sucesso é um professor perverso. Ele seduz as pessoas inteligentes e faz com que estas pessoas pensem que jamais vão cair.” (Bill Gattes).

RESUMO

O MPS.BR é um modelo de qualidade de processos que tem como principal vantagem o seu custo reduzido para certificações, sendo assim, ideal para micro, pequenas e médias empresas. Neste projeto, teve-se a intenção de aplicar aos processos de uma empresa estudo de caso o nível G do MPS.BR. A partir da definição de uma empresa estudo de caso foi analisado o setor de TI da empresa. Foram descritos os processos de cada setor, sendo mapeados os processos de gerência de projetos e gerência de requisitos existentes na empresa por meio da modelagem de negócios. A partir da análise dos processos foram realizados estudos para identificar possíveis recomendações que permitissem que o setor de TI da empresa alcançasse o nível G do MPS.BR. As recomendações foram validadas pelo gerente de TI da empresa estudo de caso.

Palavras-chave: MPS.BR. Qualidade de processos. Gerência de projetos. Análise de requisitos. Modelo de qualidade.

ABSTRACT

The MPS.BR is a model of quality of processes whose main advantage is low cost for certification, so ideal for micro, small and medium enterprises. In this project, was intended to apply to the processes of a business case study the level of G MPS.BR. From the definition of a business case study analyzed the information technology sector of the company. We describe the processes in each sector, and mapped the processes of project management and management of existing requirements in the enterprise through business modeling. From the analysis of cases studies were undertaken to identify possible recommendations that would enable the sector information technology company reached the level of G MPS.BR. The recommendations were validated by the manager of information technology business case study.

Keywords: MPS.BR, process quality. project management, requirements analysis, quality model.

LISTA DE SIGLAS

AP – Atributos do Processo
CMM - Capability Maturity Model
CMMI – Capability Maturity Model Integration
EAP – Estrutura Analítica de Projetos
ETM – Equipe técnica do modelo
ETS – Energia, Transporte e Saneamento
FATEC – Faculdade de Tecnologia de Sorocaba
FINEP – Financiadora de Estudos e Projetos
FCC – Fórum de Credenciamento e de Controle
GPR – Gerência de Projetos
GRE - Gerência de Requisitos
IEC – International Electrotechnical Commission
ISO – International Standardization Organization
MA-MPS – Modelo de Avaliação
MCT – Ministério da Ciência e Tecnologia
MN-MPS – Modelo de Negócio
MR-MPS – Modelo de Referência
MPS.BR – Melhoria do Processo de Software Brasileiro
PMBOK – Project Management Body Of Knowledge
RAP – Resultados esperados dos atributos do processo
RUP – Rational Unified Process
SIGERAR – Sistema de Gerenciamento de Requisitos
SOFTEX – Programa nacional de software para exportação
SPICE – Software Process Improvement and Capability determination
TI – Tecnologia da Informação
UNISUL – Universidade do Sul de Santa Catarina
WEB - World Wide Web (que em português significa, "Rede de alcance mundial"; também conhecida como Web e WWW)

LISTA DE QUADROS

QUADRO 1 – 22 áreas de processo do CMMi.

QUADRO 2 - Regras para avaliar o grau de implementação dos indicadores.

QUADRO 3 - Resultados esperados da gerência de projetos para o nível G

QUADRO 4 - Resultados esperados da gerência de requisitos para o nível G

QUADRO 5 - Resultados esperados da gerência de projetos para o nível G

QUADRO 6 – Resultados esperados da gerência de requisitos para o nível G

QUADRO 7 - Aproximação da realidade da empresa com os resultados esperados da gerência de projetos para o nível G

QUADRO 8 - Aproximação da realidade da empresa com os resultados esperados da gerência de requisitos para o nível G

SUMÁRIO

1 INTRODUÇÃO	15
1.1 OBJETIVO	16
1.2 OBJETIVOS ESPECÍFICOS	16
1.3 PROBLEMÁTICA	16
1.4 JUSTIFICATIVA	18
1.5 ESTRUTURA DA MONOGRAFIA	18
2 ENGENHARIA DE SOFTWARE	20
2.1 PRODUTOS	21
2.1.1 Ciclos de vida	22
2.1.2 Prazos e custos	23
2.2 QUALIDADE DE SOFTWARE	24
2.2.1 Dimensão da qualidade do processo	26
2.3 CMMI – CAPABILITY MATURITY MODEL INTEGRATION	27
2.4 MPS.BR – MELHORIA DO PROCESSO DE SOFTWARE BRASILEIRO	30
2.4.1 Método de Avaliação (MA-MPS)	32
2.4.2 Modelo de Referência (MR-MPS)	33
2.4.2.1 Capacidade do processo	34
2.4.2.2 Processos	36
2.4.2.3 O Estudo do Nível G	37
2.4.2.3.1 Gerência de Projetos	37
2.4.2.3.2 Gerência de Requisitos	41
2.4.3 Modelo de Negócio (MN-MPS)	44
2.5 CMMi x MPS.BR	44
2.6 BENEFÍCIOS DE UM PROCESSO DE QUALIDADE DE SOFTWARE	47
3 METODOLOGIA	49
3.1 TIPOS DE PESQUISA	49
3.2 METODOLOGIA DE DESENVOLVIMENTO DO PROJETO	50
3.3 PROPOSTA DA SOLUÇÃO	52
3.4 DELIMITAÇÕES	54
4 DESENVOLVIMENTO	55
4.1 A EMPRESA ESTUDO DE CASO	55
4.2 DIVISÃO DO SETOR DE TI	56
4.2.1 Gerente de TI	56
4.2.2 Analista de Sistemas	57
4.2.3 Analista Programador	57
4.2.4 Técnico em Informática	58
4.3 DESCRIÇÃO DOS PROCESSOS DA EMPRESA	58
4.3.1 Processos de gerência de projetos existentes na empresa	59
4.3.2 Gerência de Requisitos aplicada na empresa.	61
4.4 APROXIMAÇÃO DO NÍVEL G A REALIDADE DA EMPRESA	62
4.5 RECOMENDAÇÕES PARA O GRUPO DE PROCESSOS DE GERÊNCIA DE PROJETOS (GPR)	65
4.5.1 GPR2: As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados	65
4.5.2 GPR 3: O modelo e as fases do ciclo de vida do projeto são definidas	66
4.5.3 GPR 5: O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos	67

4.5.4 GPR 6: Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados	68
4.5.5 GPR 10: Planos para a execução do projeto são estabelecidos e reunidos no Plano do Projeto	69
4.5.6 GPR 13: O progresso do projeto é monitorado com relação ao estabelecido no Plano do Projeto e os resultados são documentados	71
4.5.7 GPR 15: Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento.....	72
4.5.8 GPR 16: Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas	72
4.6 RECOMENDAÇÕES PARA O GRUPO DE PROCESSOS DE GERÊNCIA DE REQUISITOS (GRE).....	73
4.6.1 GRE 3: A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida	74
4.6.2 GRE 4: Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;.....	75
4.7 VALIDAÇÃO DAS RECOMENDAÇÕES.....	77
5 CONCLUSÕES	79
REFERÊNCIAS.....	80

1 INTRODUÇÃO

Quando o assunto é desenvolvimento de software, a qualidade de software é um dos assuntos que mais vem chamando atenção nas últimas décadas. Segundo Koscianski e Soares (2006), este assunto já é bem antigo.

Como a avaliação de problemas de desenvolvimento de software e a qualidade de um produto final não são tão eficazes como é esperado, acabam-se gerando vários problemas, como retrabalho e acúmulo de serviços, o que pode influenciar no rendimento e insatisfação do desenvolvedor, e que refletem no resultado do produto final.

Mesmo com amplas ferramentas, sistemas que auxiliam na organização do trabalho, normas e padrões, não são obtidos os resultados esperados.

Segundo Silva (2003), antes que o projeto de desenvolvimento de software seja iniciado, é necessários conhecer seu escopo, sua necessidade de recursos, um cronograma de planejamento e muitas outras informações acerca do trabalho a ser realizado. A organização tem que situar estrategicamente o projeto, priorizá-lo e preparar-se para sua execução.

Infelizmente, nem sempre esta situação é executada corretamente, gerando atrasos e alterações inesperadas no sistema. A falta de planejamento pode acarretar futuros desenvolvimentos inesperados.

Nesta monografia, será apresentado um modelo de qualidade que auxilia o desenvolvimento de softwares. Este modelo se chama Melhoria do Processo do Software Brasileiro (MPS-BR) e é voltada para a área de qualidade de software para pequenas e médias empresas do Brasil. Uma das principais vantagens de seu modelo é seu custo benefício.

Tendo em vista a ampla área de desenvolvimento e os variados níveis de implantação do MPS-BR, esta monografia se aprofundará em uma camada que se identifica com o nível G, que enfatiza um padrão na área de gerência de requisitos e gerência de projetos.

1.1 OBJETIVO

Este projeto de conclusão de curso tem como objetivo principal, gerar recomendações para que possa alcançar o nível G do modelo de referência MPS.BR (Melhoria do Processo de Software Brasileiro) no setor de tecnologia da informação de uma empresa estudo de caso.

1.2 OBJETIVOS ESPECÍFICOS

Este projeto de conclusão de curso tem como objetivos específicos, gerar recomendações para que a empresa melhore a gerência dos processos, aumente a produtividade, aumente a qualidade dos softwares desenvolvidos, otimize os controles organizacionais, melhore o processo de levantamento de dados, evite acúmulo de serviços, padronize suas linhas de código e *layouts* de telas e controle suas versões de software.

Este projeto tem também como objetivo o aprofundamento nos estudos do modelo MPS.BR e da área de engenharia de software.

1.3 PROBLEMÁTICA

A empresa, em questão, desenvolve sistemas WEB e Mobile para engenheiros, setor administrativo e setor operacional da própria empresa.

Com a vontade de melhorar seus softwares e os processos de desenvolvimento, surge a necessidade de implantar padrões de qualidade. O aumento na maturidade ajuda ainda mais para que os produtos sejam melhores e mais competitivos, englobando desde o processo de levantamento dos requisitos até

o momento em que serão definidas as atividades para o controle do andamento do projeto.

O setor de TI tem como principais dificuldades viabilizar a aplicação de ferramentas de gerenciamento de projetos, devido ao número de sistemas a serem desenvolvidos e a manutenção de outros sistemas. Com um número reduzido de profissionais, os sistemas são desenvolvidos através de reuniões com a equipe técnica da empresa estudo de caso, juntamente com documentos que a equipe técnica usa que são as únicas documentações de regras de negócio executadas. O setor de TI mantém regras estabelecidas no desenvolvimento, como nomenclatura de classes, variáveis e métodos, mas não utiliza nenhum tipo de documentação de códigos.

Outro problema está relacionado a versões de aplicativo, pois, as vezes, os programadores alteram um código cuja modificação já foi executada.

A empresa, também, tem sérias dificuldades quando o assunto é levantamento de dados. Esse problema, considerado gravíssimo, torna o processo de desenvolvimento de software penoso, pois os desenvolvedores, às vezes, desenvolvem algo já definido como regra de negócio e quando chega na fase de validação ou implementação do sistema, essas regras mudam e não são passadas ao setor de TI. Com isso há o retrabalho, a insatisfação, os prazos de entrega alterados constantemente e a elevação do custo.

A empresa necessita a aplicação de um modelo de referência que facilite e simplifique as atividades citadas acima, tendo em vista um aumento nos lucros, melhoria de organização dos processos, documentação de código desenvolvido e um ambiente de trabalho mais favorável, considerando a motivação de seus desenvolvedores em melhorar a qualidade de software e o resultado final do produto.

1.4 JUSTIFICATIVA

O desenvolvimento deste projeto vai otimizar o processo de desenvolvimento de software do setor de TI da empresa estudo de caso. Com isso, facilitará a comunicação entre os membros da equipe de TI e a compreensão do que cada um desenvolve.

Qualidade de software é um processo sistemático que focaliza todas as etapas e artefatos produzidos com o objetivo de garantir a conformidade de processos e produtos, prevenindo e eliminando defeitos. (BARTIÉ, 2002).

A qualidade de um produto tem um propósito: satisfazer o cliente. (KOSCIANSKI; SOARES, 2007).

Com a implantação do MPS.BR (Melhoria do processo de software brasileiro) na empresa, todo o processo de desenvolvimento terá um aumento na qualidade. As tarefas ficarão mais claras, melhorando, assim, o resultado final do produto, deixando o usuário satisfeito e fazendo uso de um produto que passa por um modelo de qualidade e suas etapas de desenvolvimento.

1.5 ESTRUTURA DA MONOGRAFIA

- Capítulo 1: Neste capítulo, é apresentada a introdução, justificativa, problemática, assim como o tema proposto na monografia.
- Capítulo 2: Este capítulo apresenta a revisão bibliográfica sobre o assunto tema do trabalho: o modelo MPS.BR, que referencia metodologias e técnicas para um melhor desenvolvimento de software.
- Capítulo 3: Neste capítulo, é apresentada a metodologia utilizada na presente monografia.
- Capítulo 4: Este capítulo apresenta a modelagem dos processos atuais da empresa estudo de caso, assim como as recomendações para que seja atingido o nível G do MPS.BR.

- Capítulo 5: Este capítulo apresenta as conclusões sobre a presente monografia.

2 ENGENHARIA DE SOFTWARE

Para Pressman (2006), apesar de vários autores terem desenvolvido definições pessoais de engenharia de software, uma definição proposta por Fritz Bauer na conferência pioneira sobre o assunto ainda serve de base. A engenharia de software é a criação e a utilização de alguns princípios sólidos da engenharia, com o intuito de obter softwares econômicos, que sejam confiáveis e que trabalhem eficientemente em máquinas reais.

Segundo Pressman (2006), o conceito de Fritz é um pouco limitado. Ele diz pouco sobre os aspectos técnicos da qualidade de software, não fala sobre a pontualidade, não fala diretamente da necessidade de satisfação do cliente, não diz sobre a importância das medidas e unidades e não fala sobre a importância de um processo amadurecido. Mesmo assim, o conceito dá aos engenheiros de software uma linha básica.

De acordo com Pressman (2006), a engenharia de software é uma tecnologia em camadas. Qualquer abordagem de engenharia (inclusive a engenharia de software) deve se apoiar num compromisso organizacional e com a qualidade. Isso está expresso na figura 1. O foco da engenharia de software está na qualidade.



Figura 1 – Engenharia de Software em camadas
Fonte: Pressman (2006, p. 17).

Conforme Pressman (2006, p.17):

O alicerce da engenharia de software é a camada de processo. O processo de engenharia de software é o adesivo que mantém unidas as camadas de tecnologia e permite o desenvolvimento racional e oportuno de softwares de computador. [...]. Os processos de softwares formam a base para o controle gerencial de projetos de software e estabelecem o contexto no qual os métodos técnicos são aplicados, os produtos de trabalho (modelos, documentos, dados, relatórios, documentos, etc.) são produzidos, os marcos são estabelecidos, a qualidade é assegurada e as modificações são adequadamente geridas.

Segundo Pressman (2006), os métodos de engenharia de software fornecem a técnica de “como fazer” para construir softwares, e as ferramentas de engenharia de software fornecem o apoio automatizado para os processos e para os métodos. Quanto as ferramentas, algumas delas são integradas de modo que a informação criada em uma ferramenta possa ser usada por outra ferramenta. Dessa forma, um sistema de apoio ao desenvolvimento de software é estabelecido.

2.1 PRODUTOS

Segundo Paula (2001), o software, que é tratado na área como um produto, é a parte programável de um sistema de informática. Ele é um elemento central que realiza estruturas complexas que trazem funções, valor e utilidade ao sistema. Porém, outros componentes são indispensáveis para o bom funcionamento do software, como o hardware, os recursos de rede e as bases de dados.

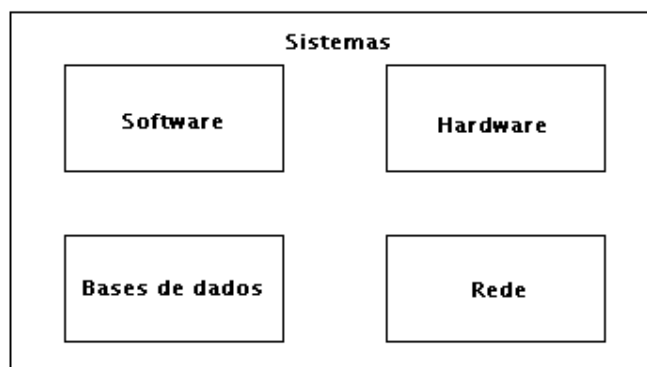


Figura 2 – Sistema de informática e suas partes.
Fonte: Paula (2001, p. 5).

2.1.1 Ciclos de vida

Para Leite (2006), o ciclo de vida de um software descreve as fases que o mesmo passa desde a sua concepção até ficar sem uso algum. Existem várias propostas para as fases do ciclo de vida de um software. Aqui mostraremos uma proposta que identifica quatro fases. Cada fase inclui um conjunto de atividades ou disciplinas que devem ser realizadas pelas partes envolvidas. As quatro fases são: Definição, desenvolvimento, operação e retirada.

- Definição: Segundo Leite (2006), nessa atividade, vários profissionais buscam o conhecimento da situação atual e a identificação de problemas para que possam elaborar propostas de solução de sistemas que resolvam tais problemas. Dentre as propostas apresentadas, deve-se fazer um estudo de viabilidade, incluindo análise de custo-benefício, para decidir qual solução deverá ser escolhida. O resultado dessa atividade deve incluir a decisão da aquisição ou desenvolvimento do sistema, indicando informações sobre hardware, software, pessoal, procedimentos, informação e documentação.
- Desenvolvimento: Segundo Leite (2006), a fase de desenvolvimento ou de produção do software inclui todas as atividades que têm por objetivo a construção do produto. Ela inclui principalmente o design, a implementação e a verificação e validação do software.
- Operação: A fase de operação envolve alguns diferentes tipos de atividades. (LEITE, 2006), que são:
 - distribuição e entrega: Pode ser feita diretamente pelo desenvolvedor ou em um pacote a ser vendido em prateleiras de lojas;
 - instalação e configuração: Normalmente, é feito com a ajuda do software de instalação, fornecido pelo desenvolvedor;
 - utilização: é o objeto do desenvolvimento do software. A qualidade da utilização é a usabilidade do software;

- manutenção: geralmente, ocorre de duas formas distintas: corretiva e evolutiva. A manutenção corretiva visa a resolução de problemas referentes à qualidade do software (falhas, baixo desempenho, usabilidade ruim, etc.). A manutenção evolutiva visa a produção de novas versões ou novos módulos a serem acionados ao software, de forma a atender a novos requisitos dos clientes;
- retirada: Segundo Leite (2006), essa é a fase mais difícil, pois, muitas vezes, os softwares estão funcionando perfeitamente na empresa, com perfeitos níveis de confiabilidade, porém eles precisam evoluir para novas plataformas operacionais e incorporação de novos requisitos.

2.1.2 Prazos e custos

Para PAULA (2001), não cumprir os cronogramas e orçamentos é parte da rotina da maioria dos profissionais de software. Isso faz com que clientes e gerentes de projeto se desesperem com os atrasos dos projetos de software. Estimar prazos e custos faz parte da rotina de qualquer ramo da engenharia. Para que um produto seja viável, não basta que ele apenas atinja os requisitos desejáveis, tem que ser construído dentro de certos parâmetros de prazo de entrega e custos. Se isso não foi possível, o produto pode não ser viável no ponto de vista do mercado, ou pode ser preferível que o cliente compre outro produto, mesmo que isso lhe custe o sacrifício de alguns requisitos. Portanto, ter estimativas de prazos e custos é mais do que importante para clientes e gerentes.

O problema, segundo PAULA (2001), é que existem desenvolvedores que não possuem os conhecimentos técnicos que permitam que eles estimem prazos e custos de desenvolvimento. Os profissionais que têm essa bagagem técnica, muitas vezes, não podem apresentar suas avaliações corretas por

trabalharem em empresas arcaicas, em que existe a política de “matar os mensageiros de notícias ruins”.

2.2 QUALIDADE DE SOFTWARE

Bartié (2002) afirma que qualquer decisão tomada durante o processo de desenvolvimento de software pode e deve comprometer a sua qualidade final. Logo, o produto final é exatamente o somatório de todas as decisões tomadas e as realizações geradas durante todo o ciclo de desenvolvimento. Se for desejado produzir software com alta qualidade, é necessário investir tempo e dinheiro em qualidade em todos os pontos do processo.

Bartié (2002) define qualidade de software como um processo sistemático que focaliza todas as etapas e artefatos produzidos com o objetivo de garantir a conformidade de processos e produtos, prevenindo e eliminando defeitos.

Para Pressman (2006), a qualidade de software pode ser definida como “concordância com os requisitos funcionais e de desempenho claramente colocados, padrões de desenvolvimento explicitamente documentados e características implícitas que são esperadas de todo software profissionalmente desenvolvido”.

Segundo Inthurn (2001), o desenvolvimento de software com qualidade é um assunto amplo, complexo e ainda muito discutido. São vários os fatores que precisam ser considerados para se obter um resultado satisfatório. Porém o principal indicador de qualidade de qualquer produto, incluindo nestes, o de software, é a satisfação do cliente, ou seja, o parâmetro que serve para exprimir o valor e as utilidades do sistema é o quanto ele atende as reais necessidades do usuário final.

Portanto, qualidade de software é um conjunto de propriedades a serem satisfeitas de modo que o software atenda as necessidades de seus usuários. (INTHURN 2001).

Sabe-se que não é suficiente somente desejar que o software a ser produzido atinja um nível adequado de qualidade. Para que esse nível seja alcançado, é necessário que esta qualidade esteja inserida em todos os ciclos de vida do desenvolvimento do software e que todos os profissionais envolvidos atuem conscientemente e objetivamente para atingi-los.

Para Bartié (2002), softwares mal testados podem provocar prejuízos enormes para as empresas. Um simples erro interno no projeto poderá encadear requisições de compras desnecessárias, solicitar manutenções de equipamentos antes do período ideal, produzir estatísticas falsas de produtividade, distribuir rendimentos de forma desproporcional, entre outros. Os problemas podem afetar o setor operacional, limitando uma tarefa específica, até a tomada de decisões de gerentes, diretores e acionistas de uma empresa. Estes são profissionais que estão se apoiando nas informações de sistemas informatizados para minimizar riscos, direcionar esforços, promover investimentos, visando tornar a empresa mais competitiva, eficiente e rentável. O sucesso das decisões tomadas por esses profissionais está ligado diretamente à qualidade das informações disponibilizadas pelos diversos sistemas usados pela empresa. É praticamente impossível obter um software de qualidade com processos de desenvolvimento frágeis e deficientes, logo, não é possível estabelecer um processo de garantia da qualidade que não enfoque simultaneamente o produto tecnológico e o processo de desenvolvimento desse software. Então, podem-se estabelecer duas dimensões fundamentais para atingir-se a qualidade do software: a dimensão da qualidade do processo e a dimensão da qualidade do produto.

Segundo Magalhães (2007), a qualidade de software é muito importante. O controle da qualidade de software aplicado em cada fase embute mecanismos que possibilitam identificar e tratar mais cedo os defeitos inseridos ao longo do ciclo de vida, bem como reduzir o número de defeitos amplificados, colaborando efetivamente para a obtenção de um produto com maior qualidade. Quanto mais cedo um defeito for encontrado, mais fácil será corrigi-lo.

2.2.1 Dimensão da qualidade do processo

Segundo Bartié (2002), quando se está diante do desafio de garantir qualidade de um software, na verdade, está-se estabelecendo uma cultura de não tolerância à falhas e erros. Os processos estão sendo estruturados de modo que os mesmos possuam mecanismos de inibição e impedimento de falhas, possibilitando, assim, que os diversos artefatos gerados durante o ciclo de desenvolvimento tenham procedimentos que avaliem sua qualidade, possibilitando, assim, a identificação prévia de defeitos nesse artefato.

Bartié ainda afirma que o termo “artefato” não deve ser entendido apenas como um produto tecnológico, e, sim, qualquer saída produzida por uma atividade do ciclo de desenvolvimento. Deve-se incluir aqui todos os documentos gerados durante o desenvolvimento do software, tais como requisitos levantados, modelos, especificações de negócios, arquitetura física, modelos de dados e classes, distribuições de componentes, análises de custos e as projeções financeiras. Todas as atividades em que o foco principal é garantir a qualidade de cada etapa do processo de engenharia de software devem ser consideradas dentro desta dimensão.

Bartié (2002) acredita que esse é um dos aspectos que mais é negligenciado dentro das empresas de desenvolvimento e fornecimento de softwares. Em grande parte dos casos, não existe um processo formal de desenvolvimento e, quando este processo existe, ele somente está no papel, não sendo respeitado pelas equipes de desenvolvimento. Logo, se a empresa deseja produzir software com qualidade, uma das primeiras providências a serem tomadas é estabelecer que esse processo seja respeitado pelas equipes.

Existem alguns modelos de qualidade e maturidade de processos que auxiliam no sucesso da qualidade dos processos. Nesta monografia, será dado ênfase ao modelo MPS.BR (Melhoria do Processo de Software Brasileiro) e ao modelo de maturidade CMMI (*Capability Maturity Model Integration*).

2.3 CMMI – CAPABILITY MATURITY MODEL INTEGRATION

Segundo Colenci (2008), o CMMi (*Capability Maturity Model Integrated*) é o modelo de qualidade para processo de desenvolvimento de software mais reconhecido no mundo. Ele é idealizado pelo instituto de engenharia de software da universidade de Carnegie Mellon, nos Estados Unidos, e teve a sua primeira versão lançada em 1991.

Segundo Humphrey (1989), citado por Colenci (2008), o modelo possui estágios de maturidade em que as empresas passam enquanto evoluem o seu ciclo de desenvolvimento. O CMMi possui 5 níveis de maturidade, conforme mostra a figura 3, abaixo:

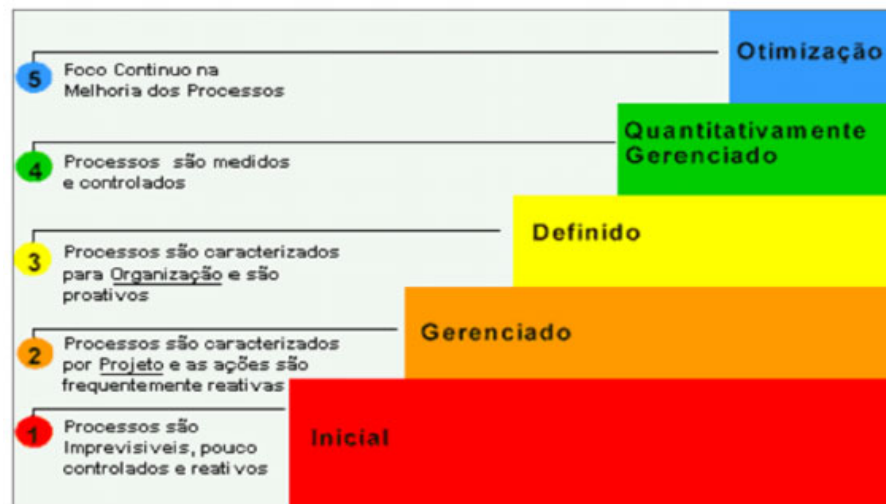


Figura 3 – Estágios do CMMi.
Fonte: Colenci (2008)

De acordo com Colenci (2008), o CMMi possui duas representações, que permitem à empresa utilizar diferentes caminhos para a melhoria de acordo com as suas necessidades. São Elas:

contínua: possibilita utilizar a ordem de melhoria que melhor atende os objetivos de negócio da empresa. É composto por níveis de capacidade;

por estágios: apresenta uma seqüência pré-determinada para melhoria baseada em estágios, que não deve ser desconsiderada. Essa seqüência é muito importante, pois cada estágio serve de base para o próximo estágio, sendo composto por níveis de maturidade.

A figura 4 apresenta, resumidamente, as características de cada nível do CMMi:

1 – Inicial	No nível 1 os processos são informais e caóticos. Existe uma completa falta de planejamento e controle dos processos. Os funcionários estão focados basicamente em atividades corretivas que surgem a todo instante.
2 – Gerenciado	Os processos básicos de gerenciamento de projetos para planejar e acompanhar custos, prazos e funcionalidades são estabelecidos. Compromissos são firmados e gerenciados. A disciplina de processo permite repetir sucessos de projetos anteriores em aplicações similares. Tipicamente, possui gerenciamento de projetos estabelecido; alguns procedimentos técnicos escritos; acompanhamento de qualidade; gerência de configuração inicial; atividades básicas de medição e análise. O sucesso depende basicamente do gerenciamento do projeto.
3 - Definido	Atividades de gerenciamento básico e de Engenharia de Software são documentadas, padronizadas e integradas em processos-padrão. Todos os projetos de desenvolvimento ou manutenção de softwares utilizam uma versão de um desses processos adaptada às características específicas de cada projeto. Possui processos gerenciais e técnicos bem definidos, possibilidade de avaliação do processo; ferramentas e metodologias padronizadas; medições iniciais de desempenho; inspeções e auditorias rotineiras; testes padronizados; gerência de configuração; evolução controlada dos processos técnicos e gerenciais.
4 – Gerenciado Quantitativamente	Métricas detalhadas do processo de software e da qualidade do produto são coletadas. Tanto o processo como o produto de software são quantitativamente compreendidos, avaliados e controlados. Relatórios estatísticos são gerados. Tipicamente, encontra-se estabelecido e em uso rotineiro um programa de medições. A qualidade é planejada por um grupo dedicado, sendo rotineiramente avaliada e aprimorada
5 - Otimizado	A melhoria contínua do processo é estabelecida por meio de sua avaliação quantitativa e da implantação planejada e controlada de tecnologias e idéias inovadoras. Projetos-piloto são realizados para a absorção e internalização de novas tecnologias. Tipicamente, um alto nível de qualidade e de satisfação dos clientes é alcançado rotineiramente, com grande foco na melhoria contínua.

Figura 4 – Resumo das características de cada estágio do CMMi.

Fonte: FIORINI (1998), citado por Colenci (2008, p.50).

De acordo com Colenci (2008), os 5 níveis de maturidade do CMMi abrangem 22 áreas de processos. Essas áreas de processo são divididas em 4 categorias:

- gerenciamento de projetos;
- gerenciamento de processos;
- engenharia;
- suporte.

O quadro 1 representa as 22 áreas de processo do CMMi, a sua categoria e o seu nível de maturidade:

Área de Processo	Categoria	Nível de Maturidade
Análise e Resolução de Causas	Suporte	5
Gerenciamento de Configuração	Suporte	2
Análise de Decisão e Resolução	Suporte	3
Gerência de Integração do Projeto	Gerenciamento de Projetos	3
Medição e Análise	Suporte	2
Inovação e Implantação Organizacional	Gerenciamento de Processos	5
Definição do Processo Organizacional	Gerenciamento de Processos	3
Foco no Processo Organizacional	Gerenciamento de Processos	3
Desempenho do Processo Organizacional	Gerenciamento de Processos	4
Treinamento Organizacional	Gerenciamento de Processos	3
Integração do Produto	Engenharia	3
Monitoração e Controle de Projeto	Gerenciamento de Projetos	2
Planejamento de Projeto	Gerenciamento de Projetos	2
Garantia da Qualidade do Processo e do Produto	Suporte	2
Gerência Qualitativa do Projeto	Gerenciamento de Projetos	4
Desenvolvimento de Requisitos	Engenharia	3
Gerência de Requisitos	Engenharia	2
Gerência de Risco	Gerenciamento de Projetos	3
Gerência de Acordo com Fornecedor	Gerenciamento de Projetos	2
Solução Técnica	Engenharia	3
Validação	Engenharia	3
Verificação	Engenharia	3

Quadro 1 – 22 áreas de processo do CMMi.

Fonte: CMMi-DEV (2006), citado por Colenci (2008, p.51).

Para Paulk (1998), citado por Colenci (2008), apesar do CMMi ser tão bem visto no mercado mundial, existe uma grande discussão a sua aplicação nas

pequenas e médias empresas. Paulk defende que o modelo em si, somente, diz o que deve ser cumprido em termos de processo, deixando a cargo das pequenas empresas a forma de trabalho, tendo, então, que buscar meios para isso.

Segundo o MCT (2007), até o final de 2006, eram poucas empresas que possuíam certificação CMMi no Brasil. Eram cerca de 21 empresas e, apenas, 6 delas então enquadradas no nível 5. Além disso, todas estas 21 empresas são de médio ou grande porte.

Para Colenci (2008), diante das dificuldades das pequenas empresas em se enquadrar no CMMi, existem algumas alternativas, entre elas está o MPS.BR.

2.4 MPS.BR – MELHORIA DO PROCESSO DE SOFTWARE BRASILEIRO

Segundo a SOFTEX (2007), o MPS.BR é um programa para Melhoria de Processo do Software Brasileiro coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), contando com apoio do Ministério da Ciência e Tecnologia (MCT), da Financiadora de Estudos e Projetos (FINEP) e do Banco Interamericano de Desenvolvimento (BID). Ele é baseado nas normas ISO / IEC 12207, ISO / IEC 15504 e na realidade do mercado de software brasileiro. Além disso, ele é compatível com o CMMI. Uma de suas principais vantagens é o seu custo reduzido para certificações, sendo assim, ideal para micro, pequenas e médias empresas. A principal motivação para a idealização desse projeto é que o Brasil é um país no qual o desenvolvimento de software está entre os maiores do mundo. É inevitável que com esse crescimento não aumente o nível de exigência dos clientes em relação à qualidade e à complexidade dos produtos. As empresas estão buscando cada vez mais a maturidade dos seus processos de software para atingir padronizações de qualidade e produtividade internacionais, que são essenciais para a sobrevivência de um mercado tão concorrido como o mercado da tecnologia da informação. Todavia, o custo de uma certificação pode chegar até 400 mil dólares, o que acaba sendo bastante inviável para empresas de pequeno e médio porte. Com esse intuito, em uma parceria entre a SOFTEX, o governo federal e as universidades, surgiu o projeto MPS.BR.

A coordenação do Programa MPS.BR conta com duas estruturas de apoio para o desenvolvimento de suas atividades: o Fórum de Credenciamento e de Controle (FCC) e a Equipe Técnica do Modelo (ETM) (MPS, 2007).

- Fórum de Credenciamento e Controle: tem como principais objetivos assegurar que as instituições implementadoras (II) e as instituições avaliadoras (IA) passem por um processo adequado de credenciamento e que suas atuações não se afastem dos limites de qualidade esperados, além de avaliar os resultados obtidos pelo MPS.BR.
- Equipe técnica do modelo: atua sobre os aspectos técnicos relacionados ao Modelo de Referência (MR-MPS) e Método de Avaliação (MA-MPS), tais como, a concepção e evolução do modelo, elaboração e atualização dos guias do MPS.BR, preparação de material e definição da forma de treinamento e de aplicação de provas, publicação de relatórios técnicos e interação com a comunidade visando a identificação e à aplicação de melhores práticas.

Segundo a SOFTEX (2007), por meio destas estruturas, o MPS.BR obtém a participação de representantes de universidades, instituições governamentais, centros de pesquisa e de organizações privadas, os quais contribuem com suas visões complementares que agregam qualidade ao empreendimento.

O MPS.BR está descrito por meio de documentos em formato de guias, que são listadas a seguir: (SOFTEX, 2007)

- guia geral: contém a descrição geral do MPS.BR e detalha o Modelo de Referência (MR-MPS), seus componentes e as definições comuns necessárias para seu entendimento e aplicação;
- guia de aquisição: descreve um processo de aquisição de software e serviços correlatos. É descrito como forma de apoiar as instituições que queiram adquirir produtos de software e serviços correlatos, apoiando-se no MR-MPS;
- guia de avaliação: descreve o processo e o método de avaliação MA-MPS, os requisitos para avaliadores líderes, avaliadores adjuntos e Instituições Avaliadoras (IA); e
- guia de implementação: composto de 7 partes, cada uma delas descrevendo como implementar um determinado nível do MR-MPS.

De acordo com a SOFTEX (2007), o MPS.BR baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de produtos de software. Assim, o MPS.BR se divide em três partes: Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS).

2.4.1 Método de Avaliação (MA-MPS)

De acordo com a SOFTEX (2007), esse método considera a aderência aos processos estabelecidos para cada nível e a adequação dos atributos de processo (AP) que implementam os processos. O grau de implementação dos resultados dos atributos de processo (RAP), relacionados a um processo e a um dos sete níveis estabelecidos, deve ser avaliado a partir de indicadores, que são:

- direto (D): documentos que comprovem que uma atividade foi realizada;
- indireto (I): artefatos intermediários resultantes de alguma atividade;
- afirmação (A): entrevista com os membros do projeto avaliado, na qual é questionado sobre a execução do projeto.

O grau de implementação dos indicadores deve ser avaliado de acordo com as quatro situações a seguir: totalmente implementada (T); largamente implementada (L); parcialmente implementada (P); e não implementada (N). (MPS 2007).

Grau de Implementação	Caracterização	Grau de Alcance
Totalmente Implementado	• O indicador direto está presente e julgado adequado. • Existe pelo menos um indicador indireto e/ou afirmação para confirmar a implementação. • Não foi notada nenhuma fraqueza substancial	>85% a 100%
Largamente Implementado	• O indicador direto está presente e julgado adequado. • Existe pelo menos um indicador indireto e/ou afirmação para confirmar a implementação. • Foi notada uma ou mais fraquezas	>50% a 85%
Parcialmente Implementado	• O indicador direto não está presente ou é julgado inadequado. • Artefatos ou afirmações sugerem que alguns aspectos da prática estão implementados. • Fraquezas foram documentadas.	>15% a 50%
Não Implementado	• Qualquer situação diferente das acima.	0 a 15%

Quadro 2 – Regras para avaliar o grau de implementação dos indicadores.
Fonte: SOFTEX (2007)

Segundo a SOFTEX (2007), a empresa pode definir a unidade organizacional que será avaliada. Uma unidade organizacional pode ser uma divisão, um setor ou toda a empresa. Para a realização da avaliação, devem ser submetidos dois projetos concluídos e dois projetos em andamento, que possuam representatividade dentro da unidade organizacional. O resultado de uma avaliação terá validade por dois anos.

2.4.2 Modelo de Referência (MR-MPS)

Segundo a SOFTEX (2007), este modelo define níveis de maturidade, que são uma combinação entre processos e sua capacidade. A definição dos processos segue os requisitos para um modelo de referência de processo, que estão especificados na ISO/IEC 15504-2, declarando o propósito e os resultados

esperados da sua execução. Isso permite avaliar e atribuir à efetividade na execução dos processos. As atividades e tarefas necessárias para atender ao propósito e aos resultados esperados devem ficar a cargo dos usuários do MR-MPS.

De acordo com a SOFTEX (2007), o MR-MPS é dividido em níveis de maturidade, que estabelecem patamares de evolução de processos, caracterizando estágios de melhoria da implementação de processos na empresa. Sabendo o nível de maturidade em que uma empresa se encontra, fica mais fácil prever o seu desempenho futuro ao executar um ou mais processos. O MR-MPS é dividido em 7 níveis de maturidade, que se inicia no nível G e vai progredindo até o nível A. São eles:

- nível A (em otimização);
- nível B (gerenciado quantitativamente);
- nível C (definido);
- nível D (largamente definido);
- nível E (parcialmente definido);
- nível F (gerenciado);
- nível G (parcialmente gerenciado).

Nesta monografia, dar-se-á foco ao nível G, que é o primeiro nível do MPS.BR.

De acordo com a SOFTEX (2007), para cada um destes sete níveis de maturidade, é atribuído um perfil de processos que indicam onde a organização deve colocar o esforço de melhoria. O progresso e o alcance de um determinado nível de maturidade do MR-MPS se obtêm quando são atendidos os propósitos e todos os resultados esperados dos respectivos processos e dos atributos de processo estabelecidos para aquele nível.

2.4.2.1 Capacidade do processo

Para Colenci (2008, p.62):

A capacidade do processo é um conjunto de atributos descrito em termos de resultados os quais proporcionam o cumprimento dos atributos de processo. A capacidade estabelece o grau de refinamento e institucionalização com que o processo é executado na empresa. À medida que se evolui nos níveis, um maior ganho de capacidade para desempenhar o processo é exigido pela empresa.

A capacidade do processo se caracteriza de acordo com 5 atributos de processo (AP), que são: (COLENCI, 2008).

AP 1.1 – O processo é executado;

AP 1.2 – O processo é gerenciado;

AP 2.1 – Os produtos de trabalho dos processos são gerenciados;

AP 3.1 – O processo é definido, e;

AP 3.2 – O processo é implementado.

A figura 5 exibe os níveis de maturidade do modelo, os seus processos e as capacidades necessárias para cada nível.

Nível	Processo	Capacidade
A	Inovação e Implementação na Organização Análise e Resolução de Causas	AP1.1, AP2.1, AP2.2, AP3.1 e AP3.2
B	Desempenho do Processo Organizacional Gerência Quantitativa do Projeto	AP1.1, AP2.1, AP2.2, AP3.1 e AP3.2
C	Análise de Decisão e Resolução Gerência de Risco	AP1.1, AP2.1, AP2.2, AP3.1 e AP3.2
D	Desenvolvimento de Requisitos Solução Técnica Integração do Produto Instalação do Produto Liberação do Produto Verificação Validação	AP1.1, AP2.1, AP2.2, AP3.1 e AP3.2
E	Treinamento Avaliação e Melhoria do Processo Organizacional Definição do Processo Organizacional Adaptação do Processo para Gerência de Projeto	AP1.1, AP2.1, AP2.2, AP3.1 e AP3.2
F	Medição Gerência de Configuração Aquisição Garantia da Qualidade	AP1.1, AP2.1 e AP2.2
G	Gerência de Requisitos Gerência de Projetos	AP1.1 e AP2.1

Figura 5 – Níveis de maturidade do MPS.BR.
Fonte: Colenci (2008, p. 58).

2.4.2.2 Processos

De acordo com Colenci (2008), processos consistem em um conjunto de operações realizadas segundo métodos próprios, desenvolvidos para o entendimento de necessidades estabelecidas ou implícitas. Eles respondem ao “como fazer”.

Os processos são descritos em termos de: (SOFTEX, 2007)

- propósito: descreve o objetivo principal a ser atingido durante a execução do processo e os prováveis resultados;
- resultados do processo: podem ser evidenciados por um artefato produzido, uma mudança significativa de estado e um atendimento as especificações;
- informações adicionais: consistem em referências que podem auxiliar na definição do processo pela empresa.

Segundo o SOFTEX (2007), os processos são agrupados de acordo com o seu objetivo principal no ciclo de vida do software. Este agrupamento consiste em 3 diferentes classes de processos, que são:

- fundamental: atendem o início e a execução do desenvolvimento, operação ou manutenção dos produtos;
- de apoio: auxiliam outro processo e ajuda no sucesso e qualidade do projeto de software;
- organizacional: uma empresa pode usar estes processos em nível corporativo para estabelecer, implementar e melhorar um processo do ciclo de vida.

A figura 6 demonstra as classes de processos fundamentais, organizacionais e de apoio do MPS.BR, bem como destaca os pontos a serem considerados no desenvolvimento de software.

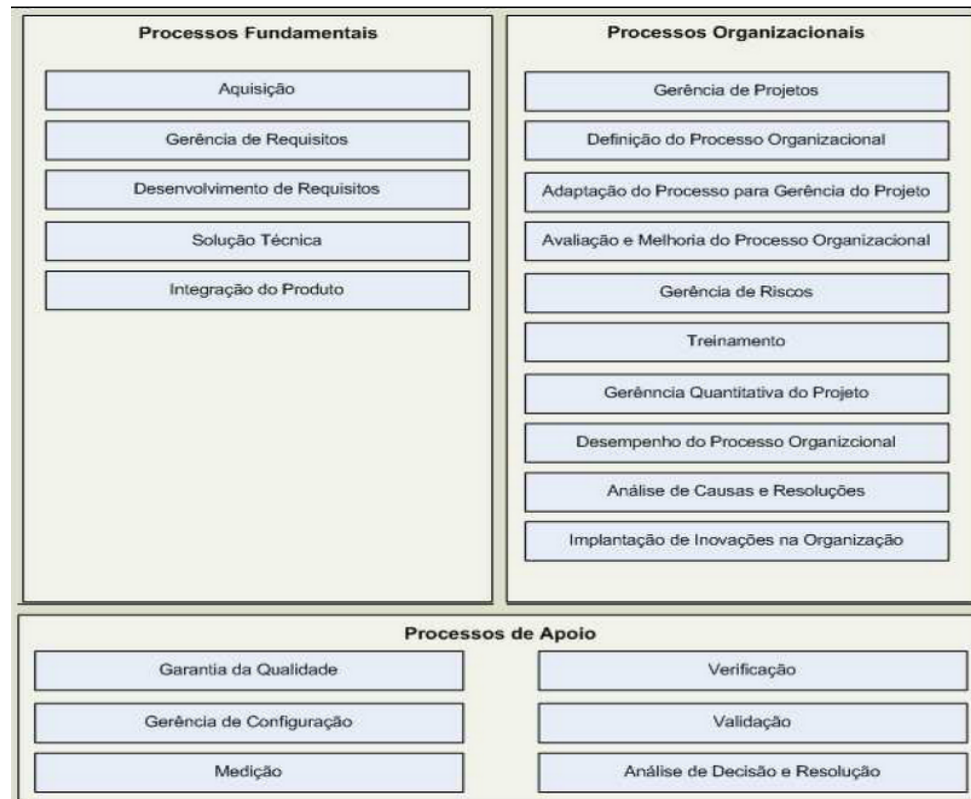


Figura 6 – Classes de processos do MPS.BR.
 Fonte: Colenci (2008, p. 61).

2.4.2.3 O Estudo do Nível G

Segundo a SOFTEX (2007), esse nível é composto pelos processos de Gerência de Projetos e Gerência de requisitos, satisfazendo, assim, os atributos de processos AP 1.1 e AP 1.2.

2.4.2.3.1 Gerência de Projetos

O propósito do processo Gerência de Projetos é identificar, estabelecer, coordenar e monitorar as atividades, tarefas e recursos envolvidos no projeto de um produto e/ou serviço. (KOSCIANSKI; SOARES, 2006). O processo de Gerência de

projetos está fundamentado em quatro ações básicas, identificar, estabelecer, coordenar e monitorar atividades, tarefas e recursos.

Segundo Koscianski e Soares (2006), ao passo que é feita a identificação dos assuntos pertinentes ao projeto, são analisadas e planejadas quais atividades e tarefas serão executadas para alcançar os objetivos, assim como os recursos que serão necessários para que o produto ou serviço seja obtido. Estabelecer implica atribuir responsabilidades pela execução de tarefas e atividades, assim como alocar os recursos necessários para que as mesmas sejam feitas corretamente. A ação de coordenar atividades, tarefas e recursos de um projeto envolve designar prioridades e integrar atividades e tarefas e desse modo distribuir recursos corretamente. Após os primeiros passos terem sido feitos, é preciso que haja constantemente um monitoramento do que está sendo realizado e utilizado, para que possíveis erros sejam identificados e tratados antes que causem dano ao plano estabelecido para execução do projeto.

Ao destacar habilidades que são importantes para o gerenciamento de projetos, devemos lembrar de um profissional, que é muito importante, o gerente de projetos. (MÁLAQUE, 2006).

Os gerentes de projeto são responsáveis por planejar e programar o desenvolvimento do projeto. Eles supervisionam o trabalho para assegurar que ele seja realizado em conformidade com os padrões requeridos e monitoram o progresso para verificar se o desenvolvimento está dentro do prazo e do orçamento. (SOMMERVILLE, 2003).

De acordo com Sommerville (2003), um bom gerente de projetos deve ter qualidades como: comunicação eficaz, influência sobre a organização, liderança, motivação, negociação e gerenciamento de conflitos e resolução de problemas. Um mau gerente de projeto leva o projeto em que é responsável ao fracasso.

Segundo Málaque (2006), para que o processo de gerência de projetos seja ágil e eficaz, o uso de ferramentas que facilitem o processo e as tarefas desenvolvidas é fundamental.

Diversas ferramentas automatizadas podem ser utilizadas para o cumprimento dessas tarefas, como gerenciadores de cronograma, simuladores e facilitadores de reuniões, que auxiliam, por exemplo, no gerenciamento de reuniões de *brainstorming*, (KOSCIANSKI; SOARES, 2006).

De acordo com Málaque (2006), as ferramentas para o auxílio do processo não precisam ser necessariamente automatizadas. Elas podem ser manuais como, por exemplo, *checklists* que auxiliem na execução de uma atividade. Um cuidado importante ao escolher uma ferramenta de auxílio é certificar-se que a ferramenta se adapta às necessidades do gerente de projeto e que os usuários saibam manuseá-la corretamente.

O propósito da gerência de projeto no MPS.BR consiste em estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, assim como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desvios significativos no desempenho do projeto. O propósito desse processo evolui à medida que a organização cresce em maturidade. (SOFTEX 2009).

O processo de gerência de projetos envolve várias atividades, como desenvolver um plano geral de controle do projeto, obter o comprometimento e mantê-lo ao longo de toda a execução do projeto; e conhecer o progresso do projeto, de maneira que ações corretivas possam ser tomadas, quando a execução do projeto desviar do planejado. (SOFTEX 2009).

Segundo o SOFTEX (2009), o desenvolvimento do plano do projeto inclui: identificar e estimar o escopo, os produtos de trabalho e as tarefas do projeto; estabelecer recursos necessários; identificar e analisar riscos do projeto; estabelecer compromissos, definir cronograma de execução baseado no ciclo de vida definido para o projeto. O plano do projeto estabelece a base de execução e controle para as atividades do projeto junto aos seus interessados (especialmente o cliente). Todos os interessados devem estar comprometidos com ele.

De acordo com o SOFTEX (2009), o progresso da execução do projeto é determinado pela comparação dos atributos reais de produtos de trabalho e tarefas, esforço, custo e cronograma com o que foi planejado nos marcos ou em pontos de controle predefinidos no planejamento do projeto.

Um marco é um ponto de revisão, por exemplo, o início ou o final de cada fase do projeto ou algumas atividades de fundamental importância para o seu sucesso. (SOFTEX 2009).

Segundo a SOFTEX (2009), a revisão de início de fase de projeto tem como objetivo verificar se as condições para que uma outra fase seja iniciada foram

atendidas. Já a revisão de fim de fase de projeto tem como objetivo verificar se todos os critérios de encerramento de fase foram cumpridos.

As revisões em marcos podem ter um caráter formal, com participação de gerências superiores, clientes e quaisquer outras partes que estejam interessadas no projeto. Sempre que necessário, deve-se realizar um replanejamento e uma nova análise de sua viabilidade, (MPS, 2009).

Para SOFTEX (2009, p.7):

Pontos de controle representam pontos entre um marco e outro nos quais revisões são realizadas para avaliar o andamento do projeto, porém, não estão no caminho crítico do projeto, ou seja, o projeto pode prosseguir mesmo que a revisão de um ponto de controle não tenha sido concluída. A visibilidade apropriada possibilita a tomada de ações corretivas quando o status do projeto se desvia significativamente do esperado. Tais ações podem exigir o replanejamento, para incluir a revisão do plano original, o estabelecimento de novos acordos ou atividades adicionais de mitigação de riscos no plano.

Os resultados esperados para a gerência de projetos são os seguintes:

GPR 1: o escopo do trabalho para o projeto é definido;
GPR 2: as tarefas e os produtos de trabalho do projeto são dimensionados, utilizando métodos apropriados;
GPR 3: o modelo e as fases do ciclo de vida do projeto são definidas;
GPR 4: (Até o nível F). O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas;
GPR 5: o orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos;
GPR 6: os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;
GPR 7: os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;
GPR 8: as tarefas, os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;
GPR 9: os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;
GPR 10: (Até o nível F). Planos para a execução do projeto são estabelecidos e reunidos no Plano do Projeto;
GPR 11: A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados.
GPR 12: o plano do projeto é revisado com todos os interessados e o compromisso com ele é obtido;
GPR 13: (Até o nível F). O progresso do projeto é monitorado com relação ao estabelecido no Plano do Projeto e os resultados são documentados;
GPR 14: o envolvimento das partes interessadas no projeto é gerenciado;
GPR 15: revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento;

GPR 16: registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas;
--

GPR 17: 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;
--

Quadro 3 – Resultados esperados da gerência de projetos para o nível G.

Fonte: SOFTEX (2009).

2.4.2.3.2 Gerência de Requisitos

Segundo Málaque (2006), um dos principais desafios no desenvolvimento de produtos é satisfazer as necessidades do cliente. Quando o produto é um software, as dificuldades aumentam. A dificuldade está relacionada à falta de preparo do profissional em levantar e gerenciar os requisitos do sistema a ser desenvolvido.

Para Blascheck (2006), citado em Málaque (2006), estão associados aos requisitos os principais problemas e dificuldades do desenvolvimento de software. Alguns desses problemas são requisitos que não refletem a real necessidade dos usuários, requisitos incompletos, mudanças em requisitos já definidos e a dificuldade para conseguir um entendimento comum entre usuários e desenvolvedores. Esses problemas e dificuldades provocam re-trabalho, atrasos no cronograma, custos ultrapassados e a insatisfação de cliente e de usuários do software.

Segundo Málaque (2006), um bom levantamento de requisitos pode dar uma idéia clara do que o sistema deve exatamente fazer e do seu comportamento.

Os requisitos de um software são as descrições sobre seu comportamento, restrições das operações que devem realizar e as especificações sobre suas propriedades ou atributos. Os requisitos compreendem as funcionalidades presentes no software, quando esse estiver pronto a ser executado. (KOSCIANSKI; SOARES, 2006).

Segundo Sommerville (2003), os requisitos podem ser divididos em requisitos funcionais e requisitos não funcionais.

- Requisitos Funcionais: Segundo Sommerville (2003), são declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas

específicas e como deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais podem também explicitamente declarar o que o sistema não deve fazer.

- Requisitos Não Funcionais: Segundo Paula (2001), esses requisitos quantificam determinados aspectos (qualidades) de comportamento do sistema.

Segundo o SOFTEX (2007), o propósito do processo de Gerência de Requisitos é gerenciar os requisitos dos produtos e componentes do produto do projeto e identificar inconsistências entre esses requisitos e os planos e produtos de trabalho do projeto. O processo de gerência de requisitos rastreia e gera os requisitos e os produtos destes, ou seja, através das especificações dos requisitos, podemos chegar aos produtos que foram pedidos e verificar se os mesmos atingiram os seus objetivos ou se inconsistências foram geradas. (MÁLAQUE 2006).

Segundo Blascheck, (2006) citado em Málaque (2006), para elaborar e manter uma especificação de requisitos é necessário que os desenvolvedores executem um conjunto estruturado de atividades destinadas a elicitar, a documentar, a analisar e a validar requisitos. Este aglomerado de atividades recebe o nome de Processo de Engenharia de Requisitos.

Elicitar requisitos é a atividade relacionada com a identificação dos requisitos do sistema, a partir de consulta aos representantes de cada grupo de usuários. A finalidade geral do processo de elicitação é identificar os fatos que compõem os requisitos do sistema, de forma a prover o mais correto entendimento do que dele é esperado (PAIM, 2003 citado por MALÁQUE, 2006).

Segundo Málaque, a primeira ação de elicitar seria levantar ou descobrir as funcionalidades que o cliente precisa. Após elicitar os requisitos, é necessário documentá-los. A documentação serve como o registro das especificações acordadas para o desenvolvimento do software. De acordo com Sommerville (2003), o documento de requisitos de software é importantíssimo, pois é a declaração oficial do que é exigido dos desenvolvedores do sistema. Para se ter uma boa documentação, deve-se adotar um modelo de documento (*template*) adequado para especificação de requisitos, identificar a fonte de todos os requisitos levantados, atribuir um nome para todos os requisitos e registrar as regras de negócio.

De acordo com Málaque (2006), outra atividade importante, na gerência de requisitos, é a análise de requisitos, pois é durante essa atividade que é verificada com maior atenção a viabilidade de cada requisito e a prioridade de cada um. Depois dessa etapa de análise, esses requisitos entram em outra etapa. Esta é a etapa de validação dos requisitos, na qual os requisitos devem ser validados, buscando identificar possíveis problemas antes da construção do sistema. A busca por erros, nessa fase, é muito importante, pois o impacto no projeto durante essa fase é menor. A falta de validação de requisitos pode implicar re-trabalho mais adiante e um custo adicional ao projeto. Para a validação dos requisitos, devem ser usadas técnicas ou ferramentas, como os *checklists* que auxiliem no trabalho.

De acordo com a SOFTEX (2009), o principal objetivo da Gerência de Requisitos no MPS.BR é controlar a evolução dos requisitos. O processo Gerência de Requisitos (GRE) gerencia todos os requisitos recebidos ou gerados pelo projeto, incluindo requisitos funcionais e não-funcionais, bem como os requisitos impostos ao projeto pela organização.

Segundo a SOFTEX (2009), para assegurar que o conjunto de requisitos acordados seja gerenciado e forneça apoio às necessidades de planejamento e execução do projeto, a empresa deve executar um conjunto de passos definidos e apropriados. Quando são levantados os requisitos pela pessoa responsável, estes devem ser revisados para resolver questões e prevenir o mau entendimento, antes que os requisitos sejam incorporados ao projeto. Assim que o cliente e a empresa chegam a um acordo, é obtido um compromisso das demais partes interessadas sobre os requisitos. Além disso, outras atribuições são destinadas ao processo Gerência de Requisitos, tais como documentar as mudanças nos requisitos e suas justificativas, manter a rastreabilidade bidirecional entre os requisitos e produtos de trabalho, em geral, e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto.

Os resultados da gerência de requisitos são os seguintes:

GRE 1 o entendimento dos requisitos é obtido junto aos fornecedores de requisitos;
GRE 2 os requisitos de software são aprovados utilizando critérios objetivos;
GRE 3 a rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
GRE 4 revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;
GRE 5 mudanças nos requisitos são gerenciadas ao longo do projeto.

Quadro 4 – Resultados esperados da gerência requisitos para o nível G.
Fonte: SOFTEX (2009).

2.4.3 Modelo de Negócio (MN-MPS)

Segundo a SOFTEX (2007), as instituições que se propõem a implantar os processos MPS.BR (Instituições Implementadoras) podem se credenciar através de um documento em que é apresentada a instituição proponente, contendo seus dados com ênfase na experiência em processos de software, estratégia de implementação do modelo, estratégia para seleção e treinamento de consultores para implementação do MR.MPS, estratégia para seleção e treinamento de avaliadores, lista de consultores de implementação treinados no modelo e aprovados em prova específica.

2.5 CMMI X MPS.BR

Depois de feitas as considerações individuais desses diferentes modelos, cabe-nos agora fazer um comparativo entre os dois, explicitando suas vantagens e limitações.

Segundo Couto (2007), citado em Colenci (2008), o MPS.BR possui uma série de vantagens sobre os outros modelos. Algumas dessas vantagens são:

- os sete níveis de maturidade do MPS.Br permitem uma implementação mais gradual, adequada à micro, pequena e média empresa, além de aumentar a visibilidade do processo de melhoria;
- o MPS.Br possui compatibilidade com CMMI e SPICE (*Software Process Improvement and Capability determination*), que são modelos reconhecidos internacionalmente;
- o modelo MPS.Br foi criado pensando na realidade da empresa brasileira, com foco na micro, pequena e média empresa de software;
- permite sua aplicação a custos acessíveis (em R\$) adequados a realidade brasileira;
- a qualificação da empresa que o aplica é mantida por avaliações periódicas (de dois em dois anos).

Segundo Colenci (2008), comparar o CMMi com o MPS.BR é muito interessante, pois o CMMi é um dos modelos de maturidade mais importantes e é reconhecido internacionalmente. O MPS.BR não tem como objetivo a criação de algo totalmente novo, mas, sim, uma adequação inovadora voltada para a realidade das empresas brasileiras. Com isso, pode-se dizer que a adequação de uma empresa ao MPS.BR permite uma futura adequação ao CMMi.

Segundo Colenci (2008), vale ressaltar que o CMMi possui cinco níveis de maturidade (1 a 5), diferente do MPS.BR que possui sete níveis (A a G). Os níveis de maturidade do CMMi possuem uma relação com os níveis de maturidade do MPS.BR, conforme apresentado na figura 7 abaixo:

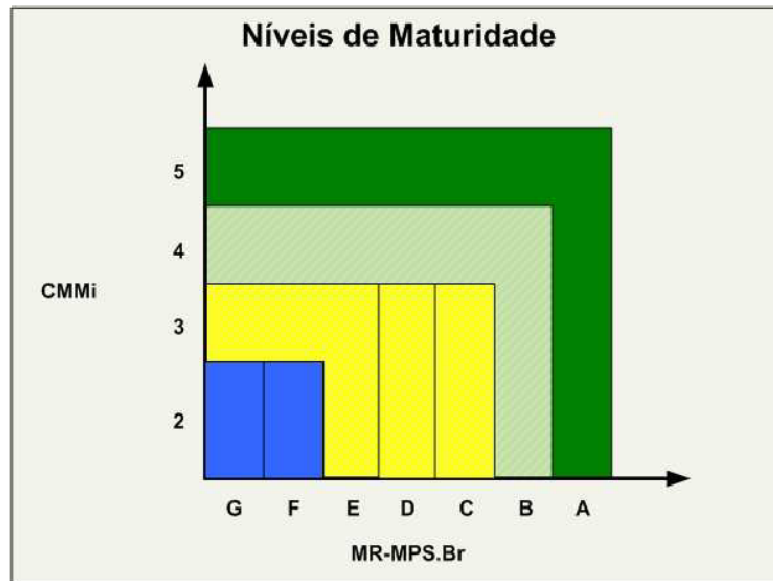


Figura 7 – Níveis de Maturidade: CMMi x MPS.BR.
 Fonte: PINTO (2006) citado por Colenci (2008, p.63).

No MR-MPS, a divisão dos níveis de maturidade permite uma implementação mais gradual e adequada às micros, pequenas e médias empresas brasileiras. A possibilidade de se realizarem avaliações, considerando um número maior de níveis permite uma visibilidade dos resultados de melhoria de processo, na empresa e no país, com prazos mais curtos. Couto (2007) citado em Colenci (2008, p.63).

MPS.BR		CMMI-SW
Nível	Processo	Áreas de Processo (Nível)
A	Inovação e implantação na organização	Inovação e implantação na organização (5)
	Análise de causas e resolução	Análise de causas e resolução (5)
B	Desempenho do processo organizacional	Desempenho do processo organizacional (4)
	Gerência quantitativa do projeto	Gerência quantitativa do projeto (4)
C	Análise de decisão e resolução	Análise de decisão e resolução (3)
	Gerência de riscos	Gerência de riscos (3)
D	Desenvolvimento de requisitos	Desenvolvimento de requisitos (3)
	Solução técnica	Solução técnica (3)
	Integração do produto	Integração do produto (3)
	Instalação do produto	
	Liberação do produto	
	Verificação	Verificação (3)
	Validação	Validação (3)
E	Treinamento	Treinamento organizacional (3)
	Avaliação e melhoria do processo organizacional	Foco no processo organizacional (3)
	Definição do processo organizacional	Definição do processo organizacional (3)
	Adaptação do processo para gerência de projeto	Gerência de projeto integrada (3)
F	Medição	Medição e análise (2)
	Gerência de configuração	Gerência de configuração (2)
	Aquisição	Gerência de acordos com fornecedores (2)
	Garantia da qualidade	Garantia da qualidade do processo e do produto (2)
G	Gerência de requisitos	Gerência de requisitos (2)
	Gerência de projetos	Planejamento do projeto (2) Gerência e controle do projeto (2)

Figura 8 – Comparação CMMi x MPS.BR por estágios.
Fonte: KOSCIANSKI; SOARES (2006, p.155).

2.6 BENEFÍCIOS DE UM PROCESSO DE QUALIDADE DE SOFTWARE

Qualquer tipo de erro gera custos financeiros à empresa. Enquanto o software não é implantado no cliente, os erros encontrados ficam restritos a retrabalho, remodelagem, recodificação e novos testes. Porém, quando esse software já está implantado, os custos começam a interferir nos resultados financeiros da empresa do cliente. Abaixo estão listados alguns dos benefícios de um processo de qualidade de software: (BARTIÉ, 2002):

- torna o ciclo de desenvolvimento confiável;
- garante ações corretivas no ciclo de desenvolvimento;
- evita a ingerência do projeto de software;
- amplia as chances de sucesso do projeto de software;

- amplia a produtividade do desenvolvimento;
- evita a propagação de erros;
- permite a automatização de testes, reduzindo os custos do projeto.

3 METODOLOGIA

O presente capítulo apresentará a metodologia usada nesta monografia.

3.1 TIPOS DE PESQUISA

Através da metodologia é que adquirimos novos conhecimentos para um bom trabalho. Barro e Lehfeld (2000, p.2) dizem que “a metodologia é o estudo da melhor maneira de abordar determinados problemas no estado atual de nossos conhecimentos. A metodologia não procura soluções, mas escolhe as maneiras de encontrá-las, integrando os conhecimentos a respeito dos métodos e vigor nas diferentes disciplinas científicas ou filosóficas.

De acordo com Cervo e Bervian (2002), a pesquisa bibliográfica procura explicar um problema a partir de referências teóricas publicadas em documentos, pode ser realizada independentemente ou como parte da pesquisa descritiva ou experimental. Em ambos os casos, busca conhecer e analisar as contribuições culturais científicas existentes sobre um determinado assunto.

Segundo Fachin (2003), a pesquisa bibliográfica diz respeito ao conjunto de conhecimento humano reunido em obras. Tem como finalidade fundamental conduzir o leitor a determinado assunto e proporcionar a produção, coleção, armazenamento e comunicação das informações coletadas para o desempenho de pesquisa.

Para Fachin (2003), esse método é caracterizado por ser um estudo intensivo. No método do estudo de caso, leva-se em consideração, principalmente, a compreensão, como um todo, do assunto investigado.

O projeto tem como principal função a explicação sistemática dos processos integrais na área de TI de uma empresa.

O estudo de caso é a pesquisa de um determinado indivíduo, assunto e tem como principal função entender “como” e “porque” funcionam as “coisas”.

A pesquisa abrangeu diversos tipos de publicações secundárias: livros, catálogos, resumos, teses, artigos, revistas, entre outros. Com essas fontes, foi encontrado algo substancial para o desenvolvimento do projeto.

Este projeto pode, portanto, ser classificado como um estudo de caso, com natureza exploratória, baseada na pesquisa bibliográfica e aplicada.

3.2 METODOLOGIA DE DESENVOLVIMENTO DO PROJETO

A metodologia de pesquisa apresenta as etapas de desenvolvimento do projeto, sendo caracterizada pela figura 9.



Figura 9 – Etapas do desenvolvimento do projeto.
Fonte: Os autores.

As etapas metodológicas definidas iniciam-se com a definição da empresa estudo de caso. A seguir, será realizado um estudo sobre a empresa e seus colaboradores. A definição dos processos e suas características serão definidas posteriormente.

Após a análise dos processos da empresa, serão identificados os processos existentes no nível G do MPS.BR.

De acordo com as pesquisas realizadas para identificar os processos do nível G, será feita a comparação entre os processos desse nível e os processos da empresa.

Finalizada a análise comparativa, será iniciado o processo de recomendações para adequação ao MPS.BR nível G. Posteriormente, serão realizadas as validações das recomendações com o modelo e a equipe da empresa.

3.3 PROPOSTA DA SOLUÇÃO

Levando-se em consideração o problema apresentado, é possível descrever a proposta de um modelo de solução para a empresa, conforme a figura, a seguir.

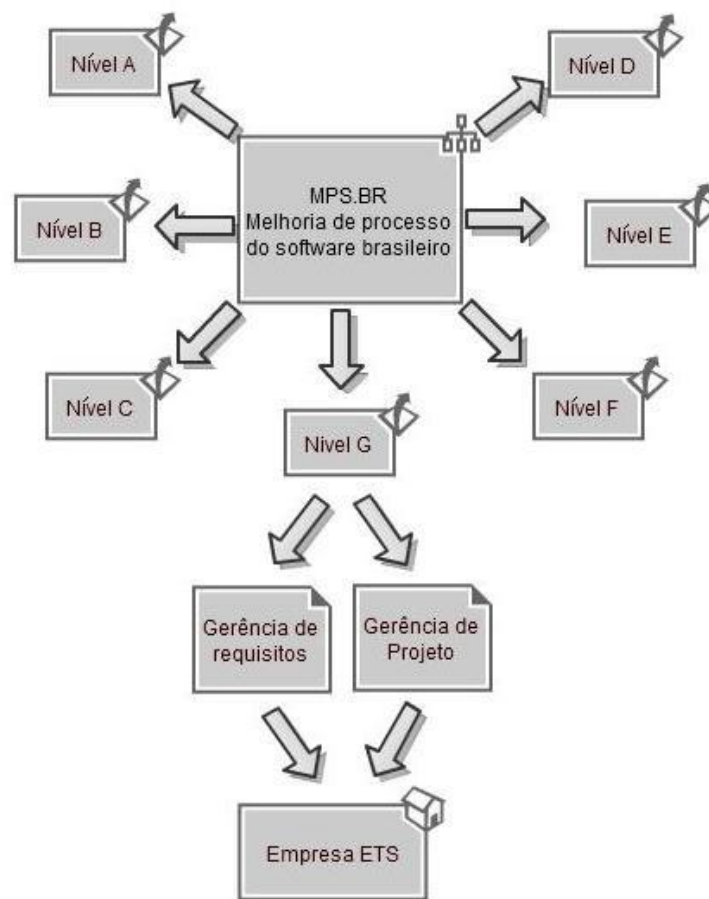


Figura 10 – Modelo de Solução.
Fonte: Os autores.

O MPS.BR possui sete níveis de maturidade:

- nível A (Em otimização);
- nível B (Gerenciado Quantitativamente);
- nível C (Definido);
- nível D (Largamente Definido);
- nível E (Parcialmente Definido).;
- nível F (Gerenciado);
- nível G (Parcialmente Gerenciado).

A escala de maturidade inicia-se no nível G e progride até o nível A. Cada nível descrito apresenta uma característica, tendo em vista a organização e objetivos para melhorar a forma de atender os objetivos do negócio.

Dentre o nível G temos descritos os seguintes processos:

- o processo de Gerência de Requisitos;
- o processo de Gerência de Projetos.

O foco do projeto é propor recomendações e artefatos que permitam a adequação da empresa ao nível G do modelo MPS.BR.

Desta forma, torna-se possível promover um melhor entendimento por parte da empresa e dos programadores sobre como demonstrar os resultados exigidos para a certificação do nível G do MPS.BR.

3.4 DELIMITAÇÕES

Este trabalho tem como objetivo principal colocar em prática os conhecimentos adquiridos através de estudos bibliográficos e documentos preliminares sobre o MPS.BR em busca de recomendações para a empresa estudo de caso.

São delimitações do projeto:

- considerando as restrições e os recursos disponíveis, não será prevista a implantação do projeto na empresa;
- dentre as técnicas identificadas na empresa, não é previsto a criação de novas técnicas ou modelos, serão assumidos modelos e técnicas já existentes customizados à realidade da empresa.

4 DESENVOLVIMENTO

Neste capítulo será apresentado o desenvolvimento a partir da metodologia apresentada, que inclui a descrição da empresa estudo de caso e a descrição dos seus processos.

4.1 A EMPRESA ESTUDO DE CASO

A empresa ETS – Energia, Transporte e Saneamento Ltda., fundada em abril de 1997, vem, ao longo destes 13 (treze) anos, destacando-se em todo o território brasileiro por sua atuação no gerenciamento e implantação de estudos e projetos na área ambiental, de engenharia, arquitetura e urbanismo, na viabilidade social e patrimonial de grandes empreendimentos do setor de geração e transmissão de energia elétrica (ETS, 2008).

Tem como missão desenvolver metodologias e instrumentos de gestão, que garantam a viabilidade dos empreendimentos, comprometendo-se com a preservação ambiental e com a justiça social.

Atualmente, a empresa conta com uma equipe composta por aproximadamente 100 funcionários: engenheiros civis, agrônomos, biólogos, sociólogos, geógrafos, geólogos, arquitetos, projetistas, programadores, técnicos em informática e comunicação social, dentre outros profissionais, focados no desenvolvimento das atividades propostas e no comprometimento dos valores corporativos.

4.2 DIVISÃO DO SETOR DE TI

O papel da tecnologia da informação (TI) nas organizações pode variar de um simples suporte administrativo até uma situação estratégica, em que ocupa posição hierárquica superior em organizações que disputam mercados em crescente competição.

O setor de tecnologia da informação da ETS – Energia, Transporte e Saneamento Ltda é composta por 3 pessoas que desempenham 4 papéis: gerente de TI, analista de sistemas, analista programador e técnico em informática. A empresa, conta também, com serviços terceirizados, como serviços de rede e de hardware.

4.2.1 Gerente de TI

Na ETS – Energia, Transporte e Saneamento Ltda, quem desempenha este papel é um profissional graduado em Sistemas de Informação pela UNISUL e técnico em processamento de dados na academia do comercio de Santa Catarina.

As principais funções deste cargo são:

- gerenciamento das atividades relacionadas ao setor de TI;
- exercer atividades de planejamento e assessoramento técnico e de supervisão nos assuntos relacionados às políticas e diretrizes de TI;
- pesquisa de Novas tecnologias;
- exercer atividades de Administração de Redes e Servidores;
- elaborar, orientar e participar de programas de treinamento na área, ministrando cursos;
- dar suporte aos analistas de sistemas e analistas programadores;
- coordenar a equipe de desenvolvimento de sistemas;
- orientar e controlar as atividades referentes a análise e projetos de sistemas;

- definir e/ou participar na elaboração de planos e projetos com vistas à implantação de sistemas de informação;
- prestar suporte na tomadas de decisão no que se refere a tecnologias de novos contratos.

4.2.2 Analista de Sistemas

Na ETS – Energia, Transporte e Saneamento Ltda, quem desempenha este papel é o mesmo profissional que desempenha a atividade de gerente de TI.

As principais funções desse cargo são:

- análise e gerenciamento do projeto em conjunto com o gerente de TI;
- atualizar-se com novas tecnologias que podem ser utilizadas no desenvolvimento dos projetos;
- levantamento de requisitos;
- repassar os requisitos e as regras de negócio ao analista programador.

4.2.3 Analista Programador

Os profissionais que desempenham esse papel são graduados em Sistemas de Informação e Ciência da Computação pela UNISUL.

As principais funções desse cargo são:

- desenvolver novos sistemas e adaptar os já existentes de acordo com a necessidade das áreas da empresa;
- dar apoio suporte a sistemas de informação desenvolvidos pelo setor de TI;
- ministrar treinamentos a usuários de tecnologias desenvolvidos pelo setor de TI;

- dar apoio na análise e desenvolvimento de tecnologias desenvolvidos pelo setor de TI;
- pesquisar novas tecnologias.

4.2.4 Técnico em Informática

Na ETS – Energia, Transporte e Saneamento Ltda, quem desempenha este papel é um profissional com capacitação técnica na área de informática.

As principais funções deste cargo são:

- dar apoio técnico e logístico na manutenção de equipamentos;
- Instalar softwares;
- dar treinamentos em sistemas de informação relacionados a programas socioambientais utilizados pela empresa e sobre conhecimentos em rotinas de backup;
- prestar assistência ao usuário da matriz e de filiais;
- analisar e comprar equipamentos e suprimentos de informática;
- prestar manutenção de redes em nível de suporte.

4.3 DESCRIÇÃO DOS PROCESSOS DA EMPRESA

Nesta seção, serão descritos os processos de gerência de projetos e gerência de requisitos na área de TI da empresa. Vale ressaltar que esses processos não são totalmente aplicados atualmente, mas tendem a ser melhorados.

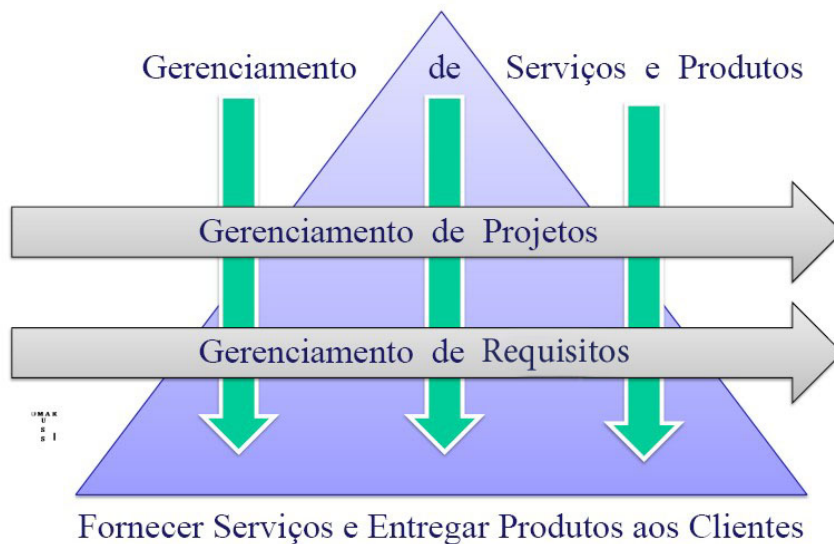


Figura 11 – Processos a serem melhorados com aplicação do nível G.
 Fonte: Webinsider UOL (2008), adaptado pelos autores.

4.3.1 Processos de gerência de projetos existentes na empresa.

Uma solicitação para o projeto é feita através dos engenheiros para o gerente de TI. Este repassa a solicitação para o analista de sistemas. Os envolvidos no projeto se reúnem com os analistas para fazer a definição do projeto. Com o projeto já definido, são levantadas as ferramentas e os recursos possivelmente usados no projeto. Com este levantamento são definidos os profissionais que estarão envolvidos até a conclusão do projeto e também uma estimativa de custos para a realização do mesmo de acordo com os históricos de outros projetos desenvolvidos pelo setor de TI da empresa. O projeto pode ser terceirizado e é cobrado certa documentação da empresa contratada.

Após os levantamentos de profissionais, recursos, ferramentas e custos, são realizadas as definições de uma proposta por parte dos analistas. Esta proposta é documentada e será validada pelos engenheiros. Caso a proposta seja válida, parte-se para a gerência de requisitos. Feita toda a gerência de requisitos, as tarefas começam a ser designadas para os analistas programadores através de uma ferramenta de gerência de projetos desenvolvida pelo próprio setor de TI da empresa, o "gerenciador operacional", que muitas vezes é descontinuado e essa

designação de tarefas passa ser feita de forma verbal. Caso a proposta não seja válida, revisões são realizadas na definição do projeto para corrigir possíveis erros.

Não existe um controle de tempo (cronograma) previsto por tarefa desempenhada, os programadores apenas começam a desenvolver e tentam terminar o mais rápido possível.

Na figura abaixo temos o diagrama de projetos representando o sistema atual da empresa no setor de TI.

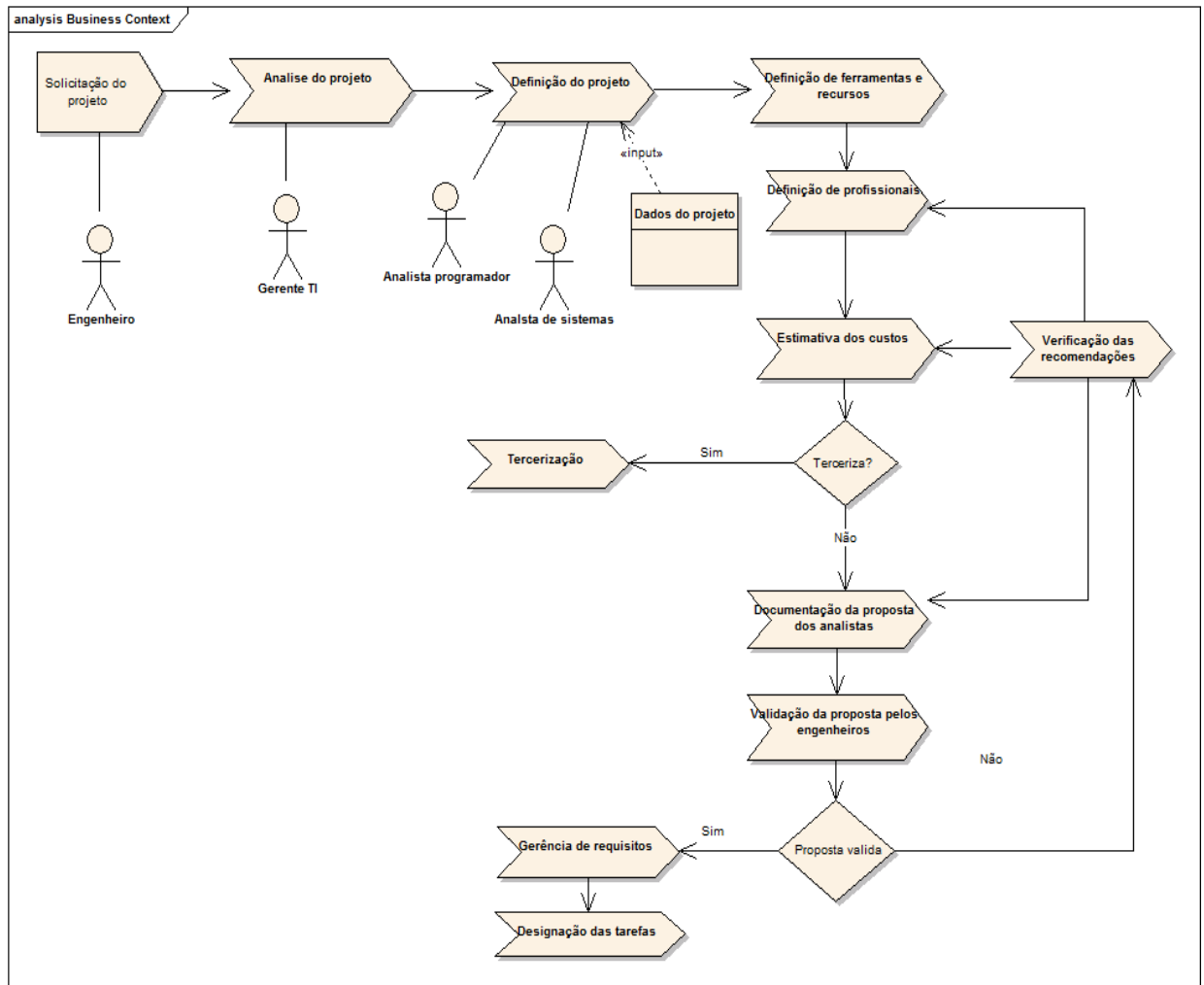


Figura 12 – Diagrama de negócio - Gerência de projetos.
Fonte: Os autores.

4.3.2 Gerência de Requisitos aplicada na empresa.

O processo de gerência de requisitos na área de TI da empresa também é relativamente simples. Os requisitos são levantados pelo analista de sistemas com a presença do analista programador e os engenheiros civis, engenheiros agrônomos, engenheiros ambientais, geólogos e biólogos, ou seja, os atores dos possíveis sistemas que são desenvolvidos

Com os requisitos levantados, são definidas as regras de negócios e ambos são documentados. Estes documentos são validados com os atores envolvidos. Uma vez aprovado, é feita a definição das telas, caso não aprovado volta-se ao levantamento de requisitos para uma nova análise.

Logo após, o analista programador faz as telas, cria a base de dados e codifica o sistema. Após a codificação o analista programador faz testes de software, caso encontre algum erro o mesmo os corrige. Após essas correções, o programador valida as funcionalidades e telas com os atores do sistema. Caso tenha alguma alteração a ser feita, o programador modifica e valida novamente. Após o sistema estar dentro das pretensões dos engenheiros, o mesmo é implantado e é dado o treinamento necessário para o uso correto.

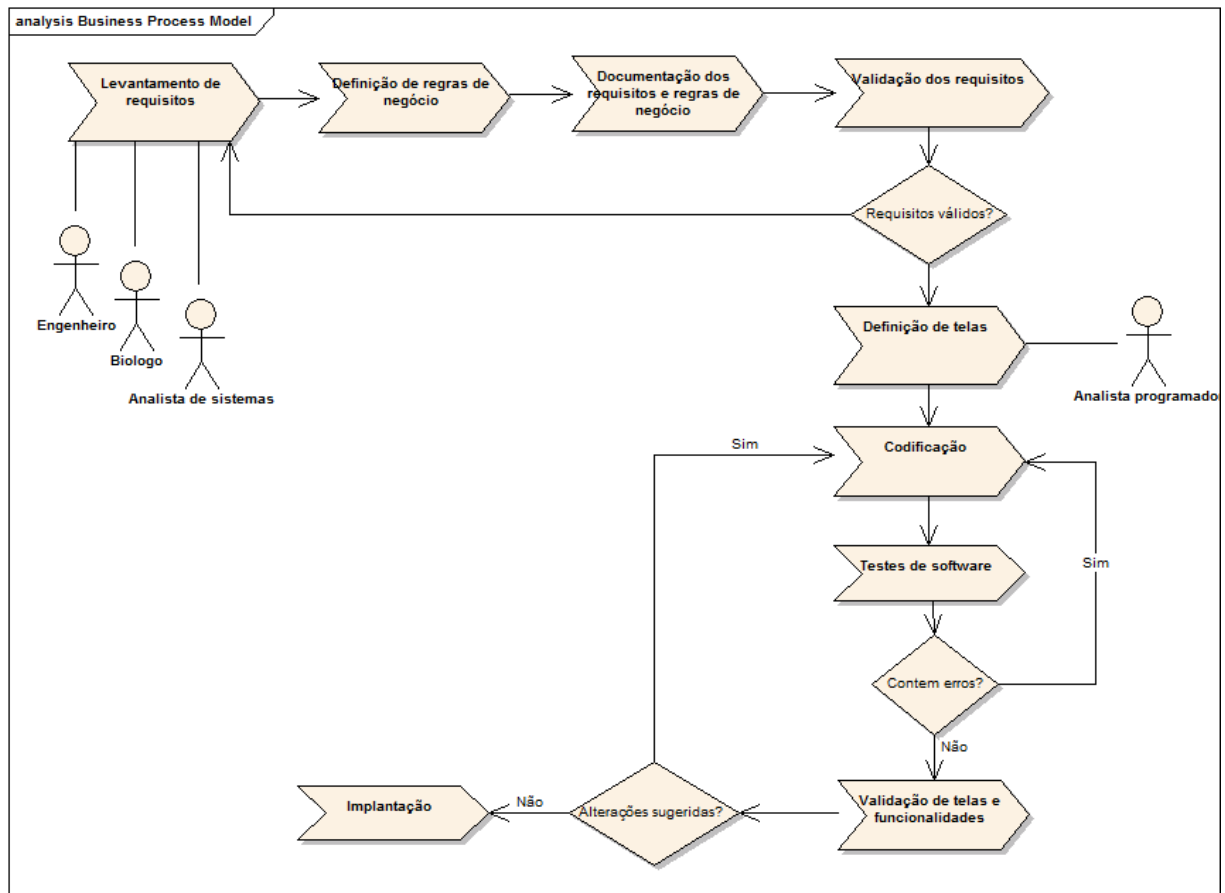


Figura 13 – Diagrama de negócio - Gerência de requisitos.

Fonte: Os autores.

4.4 APROXIMAÇÃO DO NÍVEL G A REALIDADE DA EMPRESA

Após avaliados os processos de gerência de projetos e gerência de requisitos foi possível saber quais os grupos de resultados estão sendo aplicados, quais estão sendo parcialmente aplicados e quais não estão sendo aplicados.

Nos quadros 5 e 6 estaremos relembando os resultados esperados do nível G no MPS.BR para a gerência de projetos e para a gerência de requisitos. Nos quadros 7 e 8 será apresentada a realidade da empresa de acordo com os resultados esperados pelo MPS.BR.

GPR 1: O escopo do trabalho para o projeto é definido;
GPR 2: As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;
GPR 3: O modelo e as fases do ciclo de vida do projeto são definidas;
GPR 4: (Até o nível F). O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas;
GPR 5: O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos;
GPR 6: Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;
GPR 7: Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;
GPR 8: As tarefas, os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;
GPR 9: Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;
GPR 10: (Até o nível F). Planos para a execução do projeto são estabelecidos e reunidos no Plano do Projeto;
GPR 11: A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados.
GPR 12: O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido;
GPR 13: (Até o nível F). O progresso do projeto é monitorado com relação ao estabelecido no Plano do Projeto e os resultados são documentados;
GPR 14: O envolvimento das partes interessadas no projeto é gerenciado;
GPR 15: Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento;
GPR 16: Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas;
GPR 17: 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;

Quadro 5 – Resultados esperados da gerência de projetos para o nível G.
Fonte: SOFTEX (2009).

GRE 1 O entendimento dos requisitos é obtido junto aos fornecedores de requisitos;
GRE 2 Os requisitos de software são aprovados utilizando critérios objetivos;
GRE 3 A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
GRE 4 Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;
GRE 5 Mudanças nos requisitos são gerenciadas ao longo do projeto.

Quadro 6 – Resultados esperados da gerência de requisitos para o nível G.
Fonte: SOFTEX (2009).

Após ter avaliado os processos dentro da empresa e comparados com os esperados no nível G do MPSBR, foi criado um quadro com os resultados da análise dos requisitos. Os quadros 7 e 8 apresentam o levantamento das regras que a empresa aplica na sua organização ou até mesmo não atendem por partes ou até

mesmo por completo, como por exemplo o processo GPR1 que é aplicado na empresa, entretanto temos o GPR2 que atende parte do solicitado pelo processo.

Processo	Aplica	Aplica parcialmente	Não aplica
GPR 1	x		
GPR 2		x	
GPR 3			x
GPR 4	x		
GPR 5		x	
GPR 6			x
GPR 7	x		
GPR 8	x		
GPR 9	x		
GPR 10		x	
GPR 11	x		
GPR 12	x		
GPR 13			x
GPR 14			x
GPR 15		x	
GPR 16		x	
GPR 17	x		

Quadro 7 – Aproximação da realidade da empresa com os resultados esperados da gerência de projetos para o nível G.
Fonte: os autores (2010).

Processo	Aplica	Aplica parcialmente	Não aplica
GRE 1	x		
GRE 2	x		
GRE 3			x
GRE 4		x	
GRE 5	x		

Quadro 8 – Aproximação da realidade da empresa com os resultados esperados da gerência de requisitos para o nível G.
Fonte: os autores (2010).

Após a análise dos processos aplicados na empresa foi verificado que no setor de TI não é aplicado o processo GRE3 e o GRE4 é aplicado parcialmente, tendo um contraponto entre o executado e o esperado pelo processo.

4.5 RECOMENDAÇÕES PARA O GRUPO DE PROCESSOS DE GERÊNCIA DE PROJETOS (GPR)

Após o estudo bibliográfico e a análise das possibilidades, é sugerido para o ajuste dos processos de gerência de projetos ao nível G do MPS.BR:

4.5.1 GPR2: As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados

Para que o GPR2 seja atingido, é sugerido o uso de uma Estrutura Analítica de Projetos, mais conhecido como EAP e também o uso de dados históricos.

De acordo com o PMBOK (2009), um EAP é uma decomposição hierárquica orientada às entregas do trabalho a ser executado pela equipe para atingir os objetivos do projeto e criar as entregas requisitadas, sendo que cada nível descendente da EAP representa uma definição gradualmente mais detalhada da definição do trabalho do projeto. A EAP organiza e define o escopo total e representa o trabalho especificado na atual declaração do escopo do projeto que foi aprovado.

O EAP é uma ferramenta de decomposição do trabalho do projeto em partes manejáveis.

O trabalho que não está na EAP está fora do projeto. A EAP é usada freqüentemente para elaborar ou confirmar o escopo do projeto.

Em relação ao uso de dados históricos, a empresa irá manter informações de atividades anteriores. Serão guardadas informações do projeto de maneira bem especificada que servirá para auxiliar o desenvolvimento das estimativas das atividades. Para armazenar essas informações, será criado um documento baseado na lista de problemas do RUP. Esta lista fornecerá ao gerente de projetos uma maneira de registrar e acompanhar problemas durante o andamento do projeto. Seu

formato abrangerá dados como a descrição do problema, sua importância, datas relevantes do projeto e impactos no cronograma.

Código	Descrição do Problema	Importância	Data	Alterações no Cronograma
LP001	Inconsistências dos requisitos	Alta	25/08/2010	Acréscimo de 8 dias
LP002	Validação tabelas do banco	Alta	26/10/2010	Acréscimo de 4 dias
LP003	Vazamento de memória no banco de dados	Média	03/11/2010	Acréscimo de 1 dia

Figura 14 – Modelo de lista de problemas.
Fonte: Os autores (2010).

4.5.2 GPR 3: O modelo e as fases do ciclo de vida do projeto são definidas

Para que o GPR3 seja atingido, sugere-se seguir o ciclo de vida do Scrum.

Para Leite (2006), o ciclo de vida de um software descreve as fases que o mesmo passa desde a sua concepção até ficar sem uso algum. Existem várias propostas para as fases do ciclo de vida de um software. Neste caso usaremos o ciclo de vida do Scrum, que é dividido em 4 fases: Planejamento, preparação, desenvolvimento e entrega.

A seguir será apresentada a descrição de cada fase do ciclo de vida do Scrum:

- Planejamento: estabelecer a visão do projeto e expectativas garantindo recursos para a sua execução. Nesta fase são criadas as versões iniciais do *Product Backlog* e o plano de release, arquitetura de negócio e técnica em alto nível.
- Preparação: avaliar as várias dimensões do projeto criando itens adicionais ao *Product Backlog* relacionados com o tipo do sistema, time, ambiente de desenvolvimento, tipo de aplicação. Nesta fase os

times são formados e são construídos os mecanismos de comunicação e coordenação entre eles.

- Desenvolvimento: consiste de múltiplas Sprints para o desenvolvimento dos incrementos de funcionalidade do produto.
- Entrega: realizar a entrega do produto ao cliente.

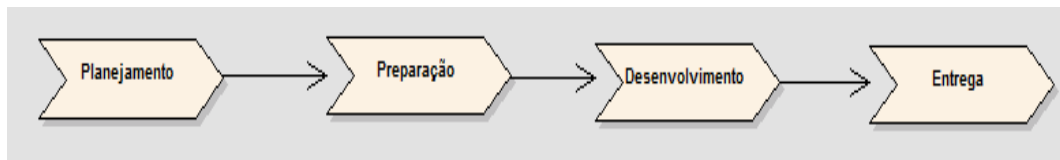


Figura 15 – Fases do ciclo de vida do Scrum
Fonte: Os autores (2010).



Figura 16 – Fases do ciclo de vida do Scrum
Fonte: IMasters UOL, (2008).

4.5.3 GPR 5: O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos

Para que o GPR5 seja totalmente atingido, deve-se criar um cronograma a partir do *Product Backlog* e a produtividade da equipe. Este cronograma deve conter os primeiros Sprints. O EAP deverá ser um auxílio na forma de definir o cronograma, assim como as estimativas de esforços e custos. É muito importante que se tenha o cuidado de manter a coerência entre o ciclo de vida, EAP, cronogramas e estimativas.

A figura abaixo mostra um exemplo do cronograma a ser utilizado.



Figura 17 – Exemplo de cronograma a ser usado.
 Fonte: Blog Carlos Santos, (2010).

4.5.4 GPR 6: Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados

Para que o GPR6 seja atingido, é sugerido que seja promovida uma interação para que sejam encontrados os riscos no projeto.

Essa interação deve contar com a presença e participação de toda a equipe do projeto, equipe de gerência de riscos, especialistas no tema de outras partes da empresa, clientes (engenheiros), usuários (operacional) e possíveis especialistas de fora da empresa (PMBOK, 2004). Todos os membros devem comparecer. Caso algum membro não possa comparecer por qualquer motivo, o mesmo deve participar via telefone ou vídeo-conferência.

De acordo com Schwaber (2004), esta interação deverá ter características da metodologia ágil, como reuniões diárias de 15 a 30 minutos para discutir as atividades, atualizar status e verificar quais os problemas e impeditivos estão atrapalhando o projeto. Caso algum membro da equipe esteja com algum problema, os membros da equipe se reúnem para discutir logo após a reunião diária.

As reuniões diárias devem ser sempre no mesmo horário e local, geralmente pela manhã, pois a primeira coisa em que os membros do time discutirão é sobre o que fizeram no dia anterior.

Deverá ser criado um modelo de documento para arquivar os riscos. Sugere-se utilizar o artefato do RUP, a lista de riscos. De acordo com o RUP (2004), a lista de riscos foi projetada para capturar os riscos aparentes para o sucesso do projeto.

Ela identifica, em ordem decrescente de prioridade, os eventos que poderiam levar a um resultado negativo significativo. Isso serve como ponto focal para atividades do projeto e é a base para a organização das iterações.

Esta lista deverá armazenar o código, a descrição e a resolução do risco como é apresentado na figura abaixo.

[illegible]

Figura 18 – Modelo de lista de riscos.
Fonte: Os autores (2010).

4.5.5 GPR 10: Planos para a execução do projeto são estabelecidos e reunidos no Plano do Projeto

Para que o GPR10 seja atingido, é sugerido que sejam usadas algumas atividades do SCRUM, como o documento de visão e o *Product Backlog*. Estes representam o plano em alto nível do projeto, com os requisitos funcionais e não funcionais e o porquê do desenvolvimento do projeto. Estas atividades irão existir durante todo o ciclo e vida do projeto e serão realizadas durante as reuniões de planejamento, chamadas de *Sprint Planning Meeting*.

O *Sprint Planning Meeting* é uma reunião na qual estão presentes o *Product Owner*, o *Scrum Master* e todo o *Scrum Team*, bem como qualquer pessoa interessada que esteja representando a gerência ou o cliente (Scrum, 2004).

O *Product Owner* tem como função representar os *stakeholders* e o negócio. Ele é quem define os requisitos do *backlog*, responsável pelo retorno sobre o investimento e as necessidades do cliente.

O *Scrum Master* tem como função manter os processos. Tem a função de remover qualquer impedimento a habilidade de uma equipe de entregar o objetivo do *Sprint*. O *Scrum Master* não é o líder da equipe, mas atua como um mediador entre a equipe e qualquer influencia que possa desestabilizar o projeto. Outra função do *Scrum Master* é de assegurar que a equipe esteja utilizando corretamente as práticas de Scrum.

O *Scrum Team* é um grupo multifuncional que faz a análise, projeto, implementação, testes e etc, ou seja, é a equipe de desenvolvimento.

Durante o *Sprint Planning Meeting*, o *Product Owner* descreve as funcionalidades de maior prioridade para a equipe. A equipe faz perguntas durante a reunião de modo que seja capaz de quebrar as funcionalidades em tarefas técnicas, após a reunião. Essas tarefas irão dar origem ao *Sprint Backlog*. O *Product Owner* não precisa descrever todos os itens que estão no *Product Backlog*. Dependendo do tamanho do *Product Backlog* e da velocidade da equipe, pode ser suficiente descrever apenas os itens de maior prioridade, deixando a discussão dos itens de menor prioridade para o próximo *Sprint Planning Meeting*.

Coletivamente, o *Scrum Team* e o *Product Owner* definem um objetivo para o *Sprint*, que é uma breve descrição daquilo que se tentará alcançar no *Sprint*. O sucesso do *Sprint* será avaliado mais adiante no *Sprint Review Meeting* em relação ao objetivo traçado para o *Sprint*.

Depois do *Sprint Planning Meeting*, a equipe Scrum se encontra separadamente para conversar sobre o que eles escutaram e decidir quanto eles podem se comprometer a fazer no *Sprint* que será iniciado. Em alguns casos, haverá negociação com o *Product Owner*, mas será sempre responsabilidade da equipe determinar o quanto ela será capaz de se comprometer a fazer (Scrum, 2004).

4.5.6 GPR 13: O progresso do projeto é monitorado com relação ao estabelecido no Plano do Projeto e os resultados são documentados

Para que o GPR13 seja atingido, sugere-se utilizar algumas atividades do SCRUM: as reuniões diárias, o *Sprint Burndown* e *Product Burndown*.

As reuniões diárias (*Daily Meeting*), devem ser gravadas (voz ou vídeo), para melhor evidenciar o que foi discutido na mesma. Essa atividade ajuda a monitorar o andamento do projeto e verificar os possíveis desvios no plano. O *Sprint Burndown* também ajuda neste quesito, pois ele mostra a velocidade da equipe e se as funcionalidades serão entregues até o final do período. (SCRUM, 2004).

O *Product Burndown*, de acordo com o SCRUM (2004), mostra o trabalho ainda restante em relação ao tempo necessário para a sua realização. O trabalho a ser realizado é representado pelo eixo Y do plano cartesiano e o tempo necessário para a realização deste trabalho é representado pelo eixo X.

Uma linha deve ser traçada com a representação da execução do trabalho; Esta linha representa o esforço já realizado na execução das tarefas. É esperado que a execução das atividades comece em Y e vá ao encontro de X. Este encontro representa o término das execuções das tarefas.

A figura 19 mostra o gráfico do *Product Burndown*.

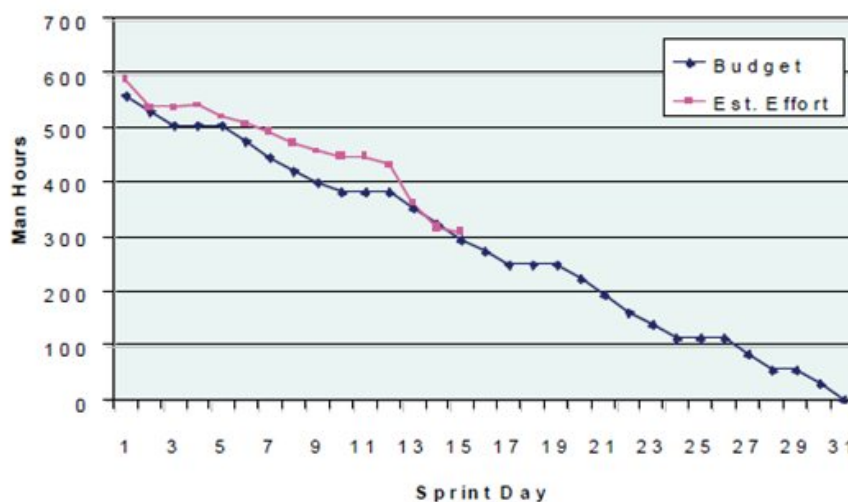


Figura 19 – Gráfico do Product Burndown
Fonte: Blog EdgardDavidson, (2009).

4.5.7 GPR 15: Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento

Para atender este requisito podemos adotar o Scrum.

Atividades de extrema importância ou início e fim de cada projeto deve ter o planejamento revisado para verificar se podem atender os requisitos exigidos pela próxima fase ou verificar se todos os critérios foram atendidos, ao final de cada sprint o progresso do projeto e o comprimento das atividades são revisadas durante o *Sprint Review* (Scrum, 2004).

No Scrum ao final de cada sprint, a equipe apresenta as funcionalidades implementadas em uma *Sprint Review Meeting*. Durante esta reunião, o *Scrum Team* mostra o que foi alcançado durante o *Sprint*. Tipicamente, isso tem o formato de um demo das novas funcionalidades.

Tendo em vistas as funcionalidades o *Sprint Review* pode ser considerado um marco do projeto, feito após cada sprint, e com os devidos registros sobre as revisões, um relatório preparado pelo *Scrum Master* para a reunião de *Sprint Review*, mostrando todos os itens do *Backlog* e que será usado para a validação do *Sprint* (Scrum, 2004).

Um *Sprint* pode ter duração de duas a quatro semanas, ficando a critério do *Scrum Master* avaliar e definir a duração de cada *sprint*, representando os pontos para as revisões do projeto.

4.5.8 GPR 16: Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas

Para que seja totalmente aplicado na empresa sugere-se que seja aplicado o modelo do Scrum, tendo assim a verificação e o controle do projeto de acordo com reuniões feitas diariamente, tendo em vista a análise e o controle dos devidos impedimentos no decorrer do projeto.

A identificação e análise de problemas junto às partes interessadas acontecem a todo o momento no Scrum, por exemplo: nas reuniões de planejamento de sprints, reuniões diárias, reuniões de revisão de sprints, reuniões de retrospectiva de cada sprint. Em todas estas reuniões há a presença do *product owner* (cliente) e equipe.

O Scrum registra todos os problemas encontrados num artefato chamado de *impediment backlog* ou na lista de problemas, onde a manutenção e ações dirigidas a estes problemas são de responsabilidade do *Scrum Master* (Schwaber, 2004).

[...] As atividades de revisão em marcos (GPR15) e de monitoramento (GPR13) do projeto possibilitam a identificação de problemas que estejam ocorrendo nos projetos. Estes problemas devem ser analisados e registrados, por exemplo, por meio de ferramentas específicas, planilhas ou outros tipos de mecanismos de gerenciamento de problemas [SOFTEX, 2009 p.17].

Durante o andamento do projeto pode ser encontrado diversos imprevistos, os problemas encontrados devem se analisados e registrados, sugere-se que seja analisado até a sua resolução, com base no plano de ação, estes problemas devem ser gravados num quadro ou na lista problemas.

4.6 RECOMENDAÇÕES PARA O GRUPO DE PROCESSOS DE GERÊNCIA DE REQUISITOS (GRE)

Após o estudo bibliográfico e a análise das possibilidades, é sugerido para o ajuste dos processos de gerência de requisitos ao nível G do MPS.BR:

4.6.1 GRE 3: A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida

Para o GR3 (grupo de processos de requisitos) ser totalmente atendido é necessário que a rastreabilidade dos requisitos seja feita desde o início.

De acordo com o RUP (2002), a rastreabilidade é a capacidade de rastrear um elemento do projeto a outros elementos correspondentes. Os elementos do projeto que estão envolvidos em rastreabilidade são chamados de itens de rastreabilidade. Os itens típicos de rastreabilidade incluem diferentes tipos de requisitos, elementos de modelo de design e de análise, artefatos de testes, material de treinamento e documentação de suporte a usuários finais.

Ao invés de fazer a documentação dos atributos de rastreabilidade em um plano de requisitos, sugere-se a escolha de uma ferramenta específica que auxilie nesta tarefa.

A sugestão é o uso de uma ferramenta web gratuita desenvolvida pela FATEC (Faculdade de Tecnologia de Sorocaba) que permite a gerência dos requisitos, o SIGERAR.

Esta ferramenta faz a coleta, armazena e mantém os requisitos, gerenciando as mudanças e promovendo a rastreabilidade entre os requisitos. (GRANDE; MARTINS, 2006).

Na figura abaixo é apresentado a tela do sistema SIGERAR em que é possível selecionar as dependências dos requisitos de um projeto.

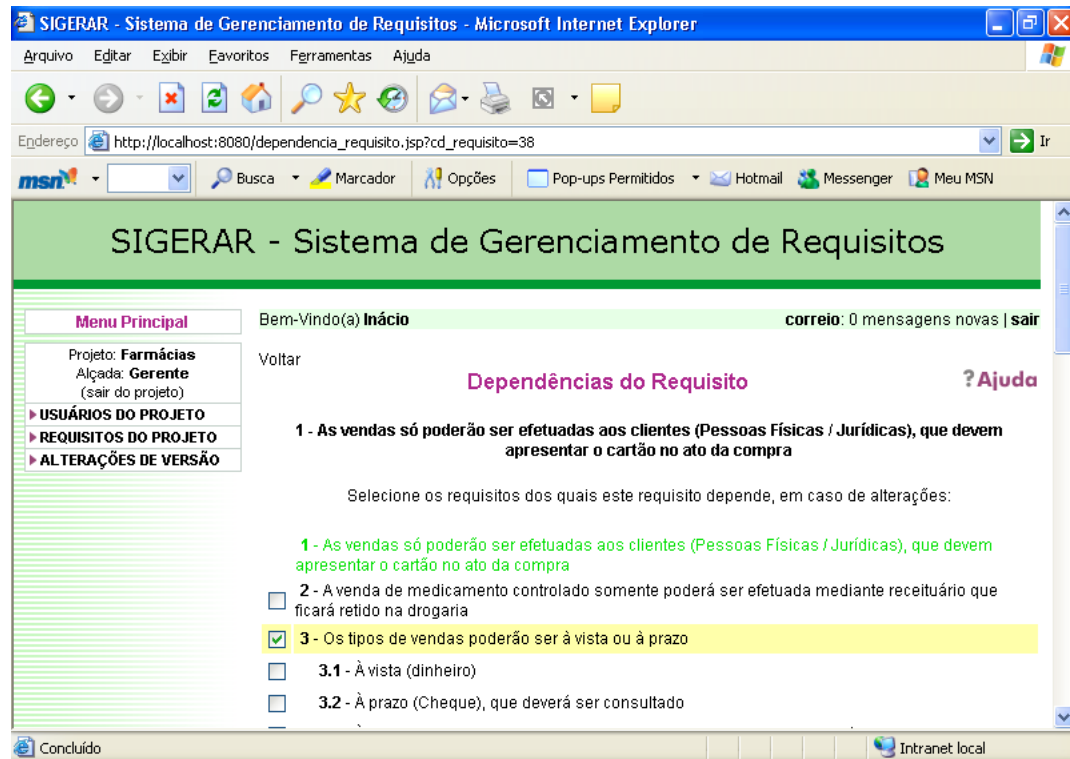


Figura 20 – Tela de dependências do requisitos do Sigerar.
Fonte: Sigerar.org, (2009).

4.6.2 GRE 4: Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;

Para que o GRE4 seja atingido totalmente, é necessário que sejam feitas revisões nos planos de produtos de trabalho freqüentemente para identificar possíveis inconsistências nos requisitos.

Para isso, é sugerido que seja aplicada a metodologia SCRUM. Nesta metodologia, os softwares são desenvolvidos em ciclos (*sprints*). Durante o ciclo, serão realizadas reuniões diárias onde são discutidos os objetivos alcançados e a alcançar. Ao final de cada *sprint*, serão realizadas reuniões para debater se os objetivos gerais estão sendo alcançados e se algum requisito precisa ser modificado. Se houver necessidade de modificação, o requisito é adicionado nos itens do *Product Backlog*, que nada mais é que a documentação dos requisitos. O objetivo do

uso desta metodologia é corrigir a inconsistência dos requisitos. Na figura abaixo é apresentado o processo do modelo.

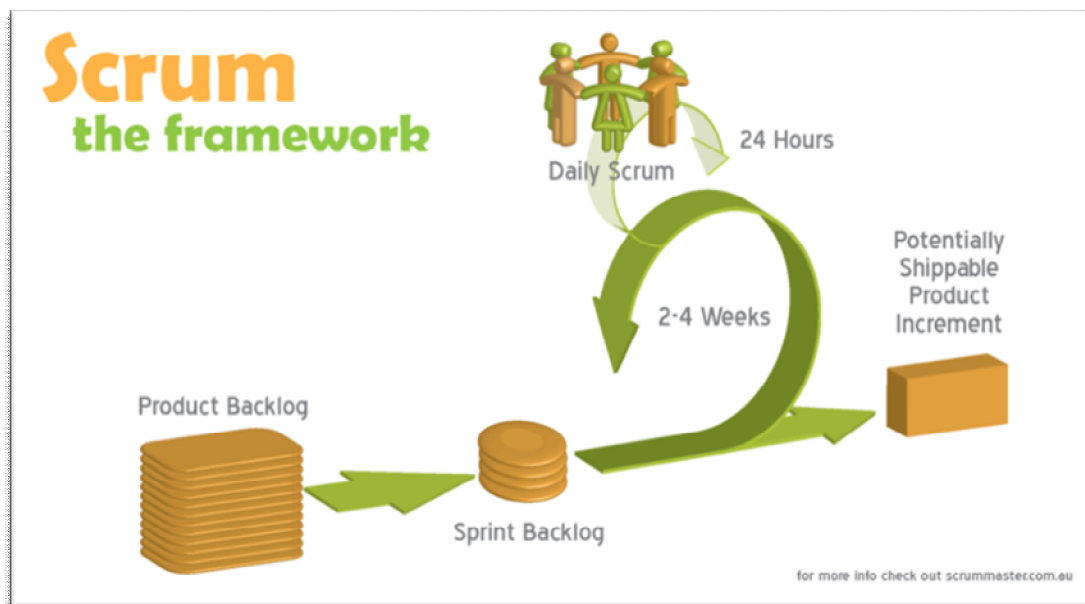


Figura 21 - Funcionamento da metodologia SCRUM.

Fonte: ScrumMaster, (2008).

A figura abaixo representa um exemplo de Product Backlog.

PRODUCT BACKLOG (exemplo)					
ID	Nome	Imp	Est	Como demonstrar	Notas
1	Depósito	30	5	Logar-se, abrir a página de depósito, depositar R\$ 10,00, ir para a página do meu saldo e verificar que este aumentou em R\$ 10,00.	Precisa de uma diagrama UML de sequência. Não é necessário se preocupar com criptografia por enquanto.
2	Verificar seu próprio histórico de transações	10	8	Logar-se, clicar em “transações”. Fazer um depósito. Voltar para transações, verificar se o novo depósito é listado.	Usar paginação para evitar consultas muito grandes ao banco de dados. Projetar de forma similar à página de visualização de usuários.

Figura 22 – Exemplo de Product Backlog.
 Fonte: Grupo Web Info Bits – GWIB, (2009).

4.7 VALIDAÇÃO DAS RECOMENDAÇÕES

A etapa de validação das recomendações apresentadas foi realizada pelo gerente de TI/Gerente de Projeto da empresa estudo de caso.

Foi realizada uma apresentação das sugestões. Em seguida, foi solicitada ao responsável pela validação, uma visão crítica sobre as recomendações apresentadas.

Para este profissional, as sugestões apresentadas podem ser aplicadas na empresa. Espera-se que a partir de artefatos, processos e técnicas do Scrum, RUP e conceitos do PMBOK, a ETS alcance todas as atividades relacionadas ao nível G do MPS.BR.

Todavia, algumas das recomendações fogem da realidade da empresa, pelo fato de exigirem que uma pessoa seja alocada única e exclusivamente para realizar

uma atividade específica, como é o caso das recomendações do GPR10, onde sugeriu-se que houvesse o *Product Owner* e o *Scrum Master* apenas desempenhando essas funções.

5 CONCLUSÕES

Com o avanço da tecnologia, o mercado está cada vez mais exigente, suprir as necessidades das empresas tem-se tornado um exercício de qualidade e controle de processos e produtos.

Essa nova realidade exige a atualização da empresa e de seus profissionais focando a qualidade de seus processos.

As empresas que não estão preparadas para desenvolver e entregar os produtos com qualidade estão perdendo cada vez mais espaço no mercado, pois este, é um mercado muito competitivo, empresas brasileiras acabam concorrendo com outras empresas brasileiras e até mesmo empresas internacionais tornando muitas vezes a disputa desleal.

Buscando reduzir custos e manter a qualidade dos softwares e seus processos, foi criado o MPS.BR, direcionado para pequenas e médias empresas

A sugestão de enquadramento da ETS no modelo MPS.BR (nível G) ajuda a empresa a aumentar ainda mais a sua competitividade e qualidade dos seus produtos e processos.

No estudo apresentado verificou-se que é possível adequar-se os processos da empresa ao MPS.BR, utilizando-se muito das técnicas e artefatos de metodologias ágeis como o SCRUM e o modelo RUP.

Entender os processos de gerência de requisitos e gerência de projetos da empresa foi fundamental para que fosse compreendido o próprio funcionamento do setor de TI da mesma, e para que fosse possível a geração das recomendações.

Como trabalhos futuros pretendem-se a aplicação das recomendações sugeridas e a análise de seus resultados na prática.

REFERÊNCIAS

BARROS, Aidil Jesus da Silveira; LEHFELD, Neide Aparecida de Souza. **Fundamentos de metodologia científica** – Um guia para a iniciação científica. São Paulo, Makron Books, 2000.

BARTIÉ, Alexandre. **Garantia da Qualidade de Software** – As melhores práticas de engenharia de software aplicadas a sua empresa. Rio de Janeiro: Editora Campos, 2002.

BLOG CARLOS SANTOS (2010). Disponível em: < <http://c-santos.blog.com/?p=5205533>>. Acesso em 25 out. 2010.

BLOG EDGARD DAVUDSON (2010). Disponível em: < <http://edgarddavidson.com/wp-content/uploads/2009/12/BurnDown.jpg>>. Acesso em 25 out. 2010.

BRASIL. MINISTÉRIO DA CIÊNCIA E TECNOLOGIA (MCT). **Qualificação CMM e CMMI no Brasil**. Disponível em: <www.mct.gov.br/upd_blob/0009/9238.pdf> Acessado em: 13 de Maio de 2010.

CERVO, A.R. BERVIAN, P. A. **Metodologia Científica**. 5 ed. São Paulo: Prentice Hall, 2002.

COLENCI, Alfredo,. **Proposta de um modelo de referência para desenvolvimento de software com foco na certificação MPS.BR**. Tese de Doutorado – Escola de engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2008.

FACHIN, Odília. Fundamentos de metodologia. 4 ed. São Paulo : Saraiva, 2003.

GRANDE, José Inácio de; MARTINS, Luiz Eduardo. **SIGERAR** – Uma ferramenta para o gerenciamento de requisitos. Tese de Mestrado – UNIMEP – Universidade Metodista de Piracicaba, Piracicaba, 2006.

GRUPO WEB INFO BITS (2009). Disponível em: < <http://gwibbers.wordpress.com/2009/10/14/um-pouco-sobre-historias-de-usuario/>>. Acesso em 28 out. 2010.

INTHURN, Cândida. **Qualidade & Teste de Software**, Florianópolis: Visual Books, 2001.

KOSCIANSKI, André; SOARES, Michel dos Santos. **Qualidade de Software** – Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2. Ed. São Paulo: Novatec Editora, 2007.

LEITE, Jair C. (2006), **Engenharia de Software - Ciclos de Vida**, Universidade Federal do Rio Grande do Norte, disponível em: <<http://www.dimap.ufrn.br/~jair/ES/slides/CiclodeVida.pdf>>. Acessado em: 21/04/2010.

MAGALHAES, Ana. (2007), **A Importância do Controle da Qualidade na Melhoria de Processos de Software**, SWQuality Consultoria e Sistemas, disponível em: <http://www.softex.br/portal/softexweb/uploadDocuments/_mpsbr/W6-1-%20Artigo_Controlde_Qualidade_Final.pdf>. Acessado em : 07/05/2010.

MÁLAQUE, Brahim José (2006), **MPS-BR Implantação dos processos do nível G**, Relatório de Estágio, Universidade Estadual de Londrina, 2006. Disponível em: <<http://www2.dc.uel.br/nourau/document/?down=356>>. Acessado em: 17/05/2010.

PAULA, Wilson de Pádua. **Engenharia de Software – Fundamentos, Métodos e Padrões**. Rio de Janeiro: Editora LTC, 2001.

PRESSMAN, Roger. **Engenharia de Software**, São Paulo: McGraw-Hill, 2001.

RUP v.2002. Disponível em: <http://www.wthreeex.com/rup/process/workflow/requirem/co_trace.htm>. Acesso em 06 out. 2010.

SCHWABER, K. BEEDLE, M. **Agile Software Development with Scrum**. Prentice Hall. 2004.

SCRUM MASTER (2008). Disponível em: <<http://www.scrummaster.com.au/Content/images/ScrumCycle.png>>. Acesso em 05 nov. 2010.

SILVA, Alex de Araujo; GOMIDE, Carlos Francisco; PETRILLO, Fabio. **Metodologia e Projeto de Software – Orientados a objetos**. São Paulo: Editora Érica, 2003.

SOFTEX – **Melhoria de Processo do Software Brasileiro**: Guia Geral. Versão 1.2. Campinas: SOFTEX, 2007.

SOFTEX – **Melhoria de Processo do Software Brasileiro**: Guia de Implementação: Nível G, parte 1. Campinas: SOFTEX, 2009.

SOMMERVILLE, Ian. **Engenharia de Software – São Paulo**: Addison Wesley, 2003.

WEB INSIDER UOL (2008). Disponível em: <<http://webinsider.uol.com.br/2008/09/16/o-que-e-automacao-de-processos-de-governanca-de-ti/>>. Acesso em 06 out. 2010.

IMASTERS UOL (2008). Disponível em: <http://imasters.com.br/artigo/8630/gerencia/sdlc_software_development_life_cycle>. Acesso em 08 out. 2010.

