



**UNIVERSIDADE DO SUL DE SANTA CATARINA
CAMPUS DA GRANDE FLORIANÓPOLIS
CURSO DE ESPECIALIZAÇÃO EM
ENGENHARIA DE PROJETOS DE SOFTWARE
SANDRO JOSÉ LONGEN**

**DESENVOLVIMENTO DE UM PROCESSO DE VERIFICAÇÃO E VALIDAÇÃO
PARA PRODUTOS DE SOFTWARE**

Florianópolis

2013

SANDRO JOSÉ LONGEN

**DESENVOLVIMENTO DE UM PROCESSO DE VERIFICAÇÃO E VALIDAÇÃO
PARA PRODUTOS DE SOFTWARE**

Trabalho de Conclusão de Curso apresentado ao curso de Pós-graduação em Engenharia de Projetos de Software da Universidade do Sul de Santa Catarina, como requisito parcial para a obtenção do título de Especialista em Engenharia de Projetos de Software.

Orientadora: Prof(a). Vera Schuhmacker, MEng.

Florianópolis

2013

SANDRO JOSÉ LONGEN

**DESENVOLVIMENTO DE UM PROCESSO DE VERIFICAÇÃO E VALIDAÇÃO
PARA PRODUTOS DE SOFTWARE**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Especialista em Engenharia de Projetos e Softwares e aprovado em sua forma final pelo Curso de Especialização em Engenharia de Projetos e Softwares da Universidade do Sul de Santa Catarina.

Florianópolis, 13 de agosto de 2013.

Orientador Prof(a). Vera Schuhmacker, MEng.
Universidade do Sul de Santa Catarina

Avaliador Prof(a). Maria Inés Castiñeira, Dr.
Universidade do Sul de Santa Catarina

RESUMO

A gerência de projetos tem demandado atenção e um crescimento significativo por parte das organizações, especialmente em projetos de desenvolvimento de software, muitas vezes por motivos relacionados à complexidade e a fatores tecnológicos. Processos e modelos têm sido desenvolvidos para estabelecer as atividades envolvidas na gerência de testes em projetos. Como exemplos desses tipos de processos e modelos estão a Integração do Modelo de Maturidade e Capacidade Integrado para Desenvolvimento - CMMI-DEV, e o Programa de Melhoria de Processo do Software Brasileiro - MPS.BR. Os projetos de desenvolvimento de software estão apresentando atualmente custos além do planejado, faltando funcionalidades planejadas e na maioria, atrasos no cronograma. Esses problemas podem ser minimizados pelo gerenciamento do projeto. Para contribuir com as atividades relacionadas à verificação e validação, realizou-se uma análise dos modelos de qualidade relacionados às áreas de verificação e validação em projetos de software, como CMMI para desenvolvimento (CMMI for development) e MPS.BR. Nesta análise foi possível a identificação de um conjunto de práticas que estejam definidas nestes modelos e possam ser citadas a fim de incorporar um modelo específico para software.

Palavras-chave: Verificação e Validação. Projetos de Software. MPS.BR. CMMI.

ABSTRACT

The project management has demanded attention and significant growth for organizations, especially in software development projects, often for reasons related to the complexity and technological factors. Processes and models have been developed to establish the activities involved in the management of testing projects. Examples of these types of processes and models are Integration Capability Maturity Model Integrated Development - CMMI-DEV, and the Program for Improvement of Brazilian Software Process - MPS.BR. The software development projects are currently hosting costs beyond the planned features missing and most planned, schedule delays. These problems can be minimized by managing the project. To contribute to the activities related to verification and validation, conducted an analysis of the quality models related to the areas of verification and validation in software projects, such as CMMI for Development (CMMI for development) and MPS.BR. In this analysis it was possible to identify a set of practices which are defined in these models and can be cited in order to embed a specific model for software.

Keywords: Verification and Validation. Software projects. MPS.BR. CMMI.

LISTA DE ILUSTRAÇÕES

Ilustração 1: Conceito “V” de teste de software.....	10
Ilustração 2: Modelo de representação contínuo.	17
Ilustração 3: Modelo de representação por estágio.	18
Ilustração 4: Níveis de maturidade do CMMI.	18
Ilustração 5: Componentes do Modelo CMMI-DEV.	19
Ilustração 6: Estrutura do modelo de referência MPS.....	24
Ilustração 7: Processo de qualidade.....	36
Ilustração 8: Processo de referência.	40

LISTA DE QUADROS

Quadro 1: Práticas específicas do processo de verificação.	21
Quadro 2: Práticas específicas do processo de validação.....	22
Quadro 3: Níveis de maturidade do MR.MPS.....	23
Quadro 4: Processo de verificação no MR.MPS.	25
Quadro 5: Processo de validação no MR.MPS.	26
Quadro 6: Análise comparativa das práticas.	27
Quadro 7: Matriz de responsabilidades.	41

SUMÁRIO

1	INTRODUÇÃO	10
1.1	SITUAÇÃO PROBLEMA	11
1.2	JUSTIFICATIVA	11
1.3	OBJETIVOS	12
1.3.1	Objetivo Geral	12
1.3.2	Objetivos Específicos	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	ENGENHARIA DE SOFTWARE.....	13
2.2	QUALIDADE DE SOFTWARE	14
2.3	IMPORTÂNCIA DA QUALIDADE NA ENGENHARIA DE SOFTWARE	16
2.4	MODELOS E NORMAS DE REFERÊNCIA DE QUALIDADE DE PROCESSO	16
2.4.1	Capability Maturity Model Integration for Development (CMMI- DEV) 16	
2.4.1.1	Verificação e Validação no CMMI	20
2.4.2	Melhoria de Processo do Software Brasileiro (MPS.BR)	22
2.4.2.1	Verificação e Validação no MPS.BR	25
2.4.3	Análise comparativa das práticas	27
2.5	TESTES DE SOFTWARE.....	30
2.5.1	Categorias de Testes de Software	31
2.5.2	Testes na Verificação e Validação.....	35
2.6	REQUISITOS PARA UM PROCESSO	36
3	DESENVOLVIMENTO	39
3.1	ESTRUTURA DO PROCESSO	39
3.2	ATIVIDADES DO PROCESSO	40
3.2.1	Definição da Equipe de Trabalho	40
3.2.2	Contextualização: Identificação das informações sobre a organização/projeto	41
3.2.2.1	Realização das entrevistas de levantamento	41
3.2.3	Preparar a Verificação	43
3.2.4	Realizar Revisões Técnicas	44
3.2.5	Análise dos Resultados da Verificação	45

3.2.6	Preparar a Validação.....	46
3.2.7	Realizar a Validação.....	48
3.2.8	Análise dos Resultados da Validação.....	49
3.2.9	Documentação e Comunicação dos Resultados.....	50
3.2.10	Avaliação do processo.....	51
3.2.10.1	Análise dos Resultados.....	52
4	CONCLUSÃO	53
	REFERÊNCIAS	54

1 INTRODUÇÃO

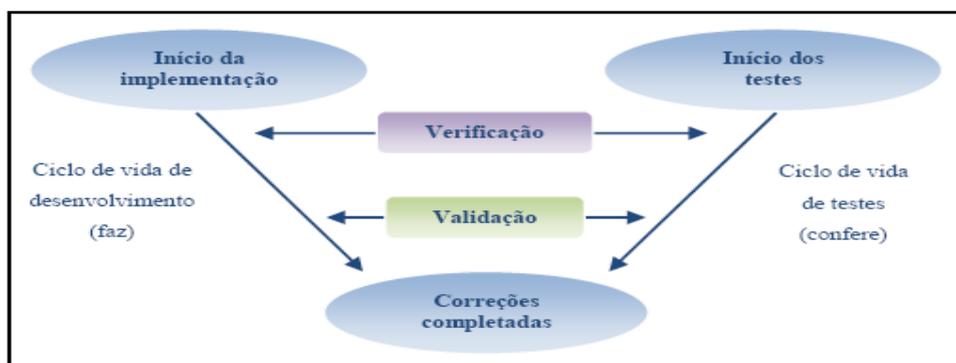
O tema proposto vem ao encontro das disciplinas oferecidas no curso de Especialização em Engenharia de Projetos de Software que tratam dos modelos Melhoria de Processo de Software Brasileiro - MPSBR e *Capability Maturity Model Integration* - CCMI-DEV e vai subsidiar a pesquisa. Assim a empresa onde será aplicado o estudo de caso teria também a oportunidade de apropriar este conhecimento no processo assim como de repercutir este novo conhecimento na equipe de qualidade e testes.

Conforme Corso (2008), existe a necessidade das organizações aperfeiçoarem o processo de teste e garantir a qualidade do software entregue, aumentando a quantidade de defeitos encontrados durante o processo de desenvolvimento e procurando identificar primeiro os defeitos que poderiam ter um custo mais elevado dentro do processo de desenvolvimento, ou impactar de forma mais significativa o negócio do cliente.

Para a SOFTEX (Associação para Promoção de Excelência do Software Brasileiro, 2009) o processo de validação consiste em confirmar que um produto atenderá os objetivos de seu uso, quando colocado no ambiente de produção e o processo de verificação confirma se o produto está atendendo aos requisitos especificados durante as atividades executadas no desenvolvimento do produto. Então, a importância de se fazer verificação e validação em teste de software estão em checar se o sistema esta de acordo com os requisitos especificados pelo cliente.

Bastos *et al.* (2007) citam o conceito “V” de teste, ou seja, o procedimento de fazer e conferir converge do início até o fim do projeto (vide Ilustração 1). A equipe que programa o sistema e a equipe que testa, executam procedimentos de testes visando minimizar ou eliminar defeitos.

Ilustração 1: Conceito “V” de teste de software.



Fonte: Bastos et al, 2007.

O teste, da maneira como é executado pela maioria das empresas, como uma etapa do processo de desenvolvimento e, em geral, executado no final, pelos próprios desenvolvedores ou até pelos usuários do sistema, serve apenas para garantir que as especificações ou os requisitos foram implementados.

Desta forma, espera-se que, a existência de um processo de Verificação e Validação, aderente às principais normas e modelos de processo auxilie na implementação de um processo sistemático de testes. Esta iniciativa, alinhado às melhores práticas dos modelos e normas de referência, tende a resultar em uma redução de defeitos encontrados num produto de software.

1.1 SITUAÇÃO PROBLEMA

Atualmente os processos de Verificação e Validação - V&V, sendo executados de maneira não sistemática e pelos próprios desenvolvedores e usuários do sistema, não têm sido suficientes para garantir que as especificações ou os requisitos sejam implementados de maneira correta. Além disso, como o processo de desenvolvimento tende a criar produtos com defeito, é necessário prevenir, descobrir e corrigir esses defeitos o quanto antes. Como o Desenvolvimento de um Processo de Verificação e Validação pode melhorar a qualidade das entregas de produtos de software?

1.2 JUSTIFICATIVA

As áreas de Tecnologia da Informação - TI demandam aprimoramento constante dos profissionais. O curso de Especialização em Engenharia de Projetos de Software que tem diversas disciplinas da área, como a Gerência da Qualidade de Software - CMMI propõe agregar valor e qualificação às atividades exercidas pelos profissionais de TI.

Assim, o desenvolvimento do trabalho estará mediando o conhecimento adquirido no curso com a prática empresarial, gerando um novo conhecimento adequado à realidade do mercado. A possibilidade de aperfeiçoamento de conhecimentos nessa área na busca da melhoria no desenvolvimento dos produtos privilegiando sua qualidade é fundamental para o mercado carente de profissionais com este perfil.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Descrever o desenvolvimento de um processo de Verificação e Validação para produtos de software.

1.3.2 Objetivos Específicos

- Pesquisar na literatura modelos e normas de referência que tratem das áreas de verificação e validação;
- Pesquisar ferramentas de software que ofereçam suporte para as áreas de verificação e validação;
- Descrever um processo de verificação e validação de produtos de software, visando aplicação e avaliação do mesmo.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados conceitos relacionados ao tema do trabalho.

2.1 ENGENHARIA DE SOFTWARE

Segundo Pressman (2006), uma primeira definição de Engenharia de Software foi proposta por Fritz Bauer (*apud* PRESSMAN, 2006): O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.

Conforme Pressman (2006) a Engenharia de Software abrange um conjunto de três elementos fundamentais que possibilita ao gerente o controle do processo de desenvolvimento do software e oferece ao profissional uma base para a construção de software de alta qualidade. Os elementos fundamentais citados abaixo compreendem um conjunto de etapas que envolvem métodos, ferramentas e procedimentos:

- Métodos: proporcionam os detalhes de “como fazer” para construir o software;
- Ferramentas: proporcionam apoio automatizado ou semi-automatizado aos métodos; e
- Procedimentos: constituem o elo que mantêm juntos os métodos e as ferramentas e possibilita o desenvolvimento racional e oportuno do software de computador.

Conforme Inthurn (2001), a Engenharia de Software permite que o gerente controle o processo de desenvolvimento, fornecendo ao engenheiro de software as bases para a construção de software de alta qualidade e de modo produtivo, onde o desenvolvimento engloba três elementos fundamentais: métodos que fornecem a técnica para a construção do software, as ferramentas que fornecem suporte automático ou semiautomático para os métodos, e os procedimentos que unem os métodos e as ferramentas, permitindo o desenvolvimento do software de forma racional e no prazo estipulado.

2.2 QUALIDADE DE SOFTWARE

Antes de definir qualidade de software, é preciso entender o que é o software. Pressman (2006) define o software como sendo um conjunto de instruções, que ao serem executadas fornecem características, funções e desempenho desejados. O autor define ainda que o software é uma estrutura de dados que permite aos programas manipularem adequadamente as informações, e por último, que estes são os documentos que descrevem a operação e o uso dos programas.

A gestão de qualidade de software já está incorporada às normas de qualidade do produto de software faz algum tempo, porém não existe uma cultura de se exigir tais padrões ao se adquirir um produto de software (BARTIÉ, 2006). Algumas normas e modelos aplicadas à qualidade do produto de software ou qualidade do processo são (INTHURN, 2001):

- Modelo CMM/CMMI: apesar de não ser uma norma, esse modelo é muito bem aceito no mercado mundial e tem como objetivo a qualidade do processo de software;
- MPS.BR: é baseado no CMMI, nas normas ISO/IEC 12207 e ISO/IEC 15504, o objetivo é o desenvolvimento para a melhoria e um modelo de qualidade de processo;
- Norma ISO/IEC 12119: estabelece os requisitos de qualidade para pacotes de software e instruções para teste, considerando-se esses requisitos; e
- Norma ISO/IEC 15504: tem como objetivo avaliar os processos de desenvolvimento de software.

Conforme Inthurn (2001) sabe-se que a definição de qualidade é um tema amplo e complexo. Para se atingir um resultado satisfatório, inúmeros fatores devem ser considerados e atendidos, porém o principal indicador de qualidade de software é a satisfação do cliente. O autor afirma ainda que a qualidade no desenvolvimento de software significa:

- alinhamento total entre as necessidades e expectativas dos usuários e as especificações geradas;

- alinhamento total entre as especificações aprovadas e o produto construído; e
- produto final com a menor quantidade de erros possível.

Por fim, o autor afirma que:

Qualidade de software é um conjunto de propriedades a serem satisfeitas de modo que o software atenda às necessidades de seus usuários. [...]. Como em qualquer ramo industrial, a qualidade do produto é um objetivo de projeto, sendo raras as ocasiões em que a qualidade poderá ser incorporada ao produto após este ter sido construído. Assim, ao desenvolvermos software, devemos sempre ter em mente o objetivo de qualidade previamente estabelecido. Sem conhecer este objetivo e sem trabalhar conscientemente para alcançá-lo, logo no início do desenvolvimento, torna-se impossível assegurar a qualidade do software no final de sua construção (INTHURN, 2001, p. 22-23).

Para Pressman (2006), a qualidade de software é definida como:

Conformidade com requisitos funcionais e de desempenho explicitamente declarados, normas de desenvolvimento explicitamente documentadas e características implícitas, que são esperadas em todo software desenvolvido profissionalmente (PRESSMAN, 2006, p. 724).

Conforme Sommerville (2003), o planejamento da qualidade deve ser iniciado juntamente com o processo de desenvolvimento do software, devendo definir as qualidades desejadas para o produto e como estas deverão ser avaliadas. O autor ainda apresenta uma lista de atributos de qualidade, que deverão ser observados durante o processo de planejamento da qualidade, definindo assim, os atributos mais significativos para o produto. Entre estes atributos, pode-se destacar: segurança, proteção, confiabilidade, capacidade de recuperação, robustez, facilidade de compreensão, testabilidade, facilidade de adaptação, modularidade, complexidade, portabilidade, facilidade de uso, facilidade de reuso, eficiência, facilidade de aprendizado entre outros.

Portanto, pode-se observar que a qualidade do produto de software é também determinada pela qualidade dos processos adotados para o desenvolvimento, sendo assim, a melhoria de qualidade do software pode ser obtida pela melhoria de qualidade dos processos. Inicialmente, a seguir são apresentadas algumas normas para a qualidade

de produto de software e na sequência normas e modelos de referência para qualidade de produto são também apresentados.

2.3 IMPORTÂNCIA DA QUALIDADE NA ENGENHARIA DE SOFTWARE

Conforme Molinari (2003), qualidade de software é algo que todos querem. Os gerentes sabem que eles precisam ter alta qualidade em seus trabalhos, os desenvolvedores sabem que desejam produzir um produto de alta qualidade, os usuários, insistem que o seu trabalho através do uso do software deve ser confiável e consistente.

Com o desenvolvimento da Engenharia de Software diversos esforços estão sendo empregados na tentativa de melhorar a qualidade do software desenvolvido. Neste sentido diversas normas e modelos de maturidade como CMMI (SEI, 2006), MPS-BR (SOFTEX, 2009), ISO 12207 (ISO, 2004) e 15504-5 (ISO, 2006) foram desenvolvidos com o objetivo de estabelecer processos e produtos de software de melhor qualidade.

A qualidade de software é um tema tão importante que é encontrado em todas as outras áreas de conhecimento envolvidas em um projeto de desenvolvimento de software. Além disso, ele deixa claro que essa área, trata daqueles que não exigem a execução do software para avaliá-lo, em contraposição à área de conhecimento teste de software (IEEE, 1999).

2.4 MODELOS E NORMAS DE REFERÊNCIA DE QUALIDADE DE PROCESSO

Conforme descrito acima, a qualidade do produto de software é fortemente dependente da qualidade do processo que o produz. Desta forma, nesta seção são contextualizados os modelos de referência abordados nesse trabalho, que serão utilizados como base para a identificação e a fundamentação das atividades de Verificação e Validação.

2.4.1 Capability Maturity Model Integration for Development (CMMI-DEV)

Criado pelo *Software Engineering Institute* - SEI, o CMMI surgiu da necessidade de integrar os diversos modelos de maturidade disponíveis e compatibilizar

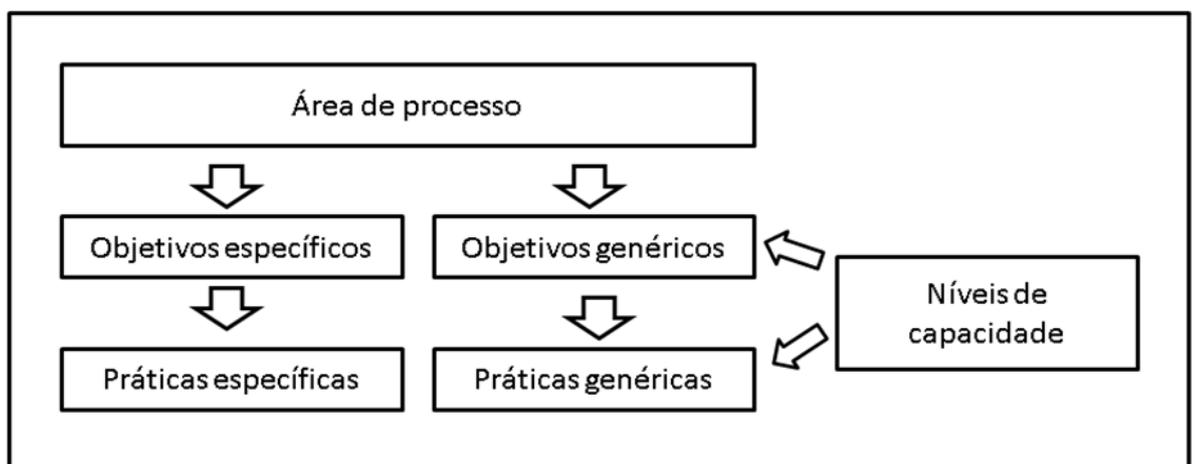
o SW-CMM com a norma ISO/IEC 15504. A utilização de diversos modelos de maturidade mostrou-se problemática nas organizações. O CMMI foi idealizado com o objetivo de resolver esse problema de integração.

Desde a primeira versão, o SEI vem trabalhando na estruturação de um framework de melhoria que possa ser aplicado também em outras áreas de interesse. Desta forma, nasceram as chamadas "constelações", nas quais componentes do CMMI são usados para construir modelos, materiais de treinamento e documentos de avaliações que são agrupados, gerando assim uma constelação. O modelo CMMI na sua versão 1.2 contempla a constelação *CMMI for Development*, que inclui definições quanto ao método de avaliação e materiais de treinamento. Também compõem a arquitetura de modelos: *CMMI for Services* e *CMMI for Acquisition* (SEI, 2006).

O CMMI-DEV é constituído de dois tipos de representação: contínuo (Ilustração 2) e por estágios (Ilustração 3).

Modelo de representação contínuo – Para uma única área de processos ou conjunto de áreas de processos. Avalia os processos de zero (0) a cinco (5) níveis de capacidade (sendo zero (0) incompleto, um (1) executado, dois (2) gerenciado, três (3) definido, quatro (4) gerenciado quantitativamente e cinco (5) otimizado).

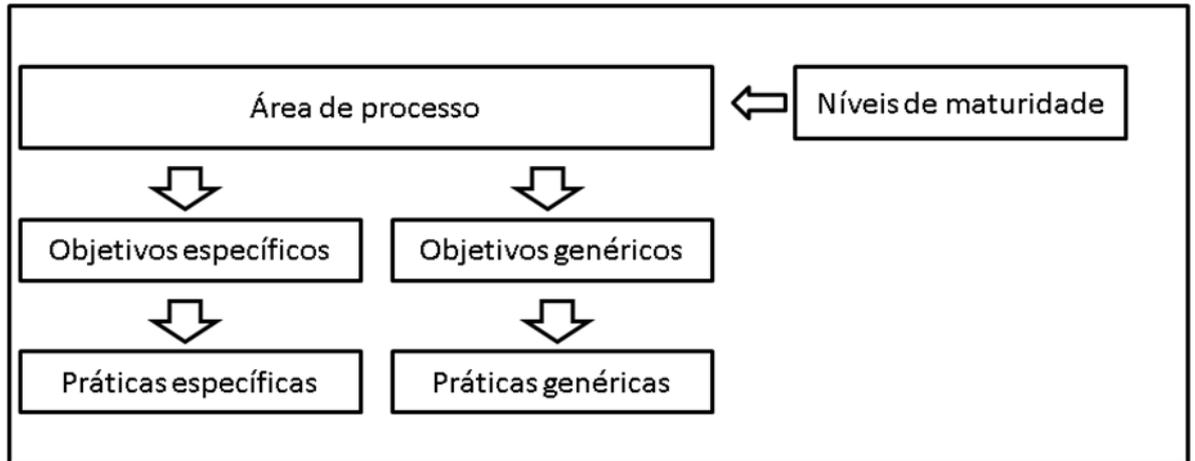
Ilustração 2: Modelo de representação contínuo.



Fonte: SEI (2006)

Modelo de representação por estágios – Para um conjunto estabelecido de áreas de processos pela organização. Avalia os processos de um (1) a cinco (5) níveis de maturidade (sendo um (1) inicial, dois (2) gerenciado, três (3) definido, quatro (4) gerenciado quantitativamente e cinco (5) otimizado).

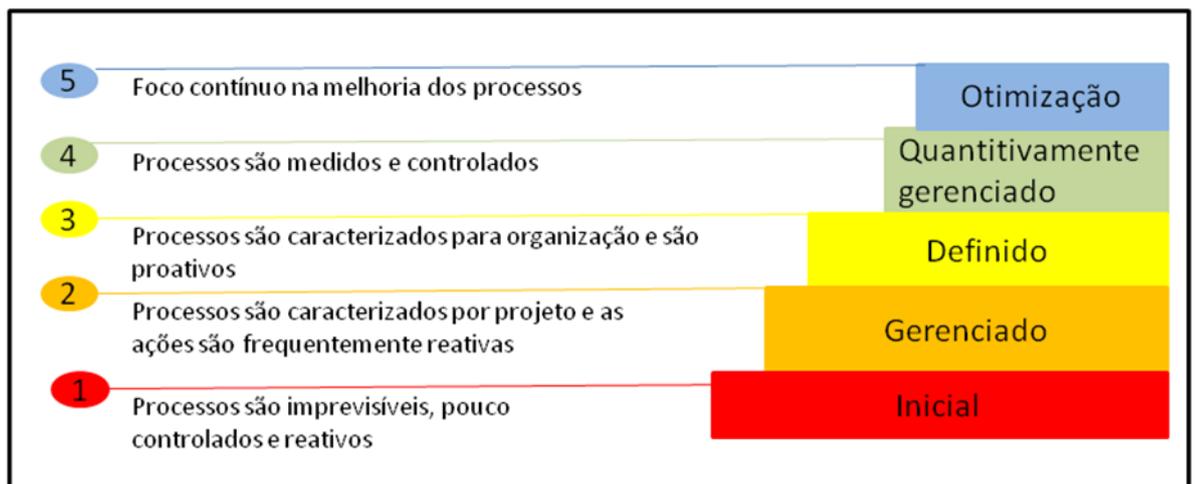
Ilustração 3: Modelo de representação por estágio.



Fonte: SEI (2006)

De acordo com a representação por estágios, o CMMI possui cinco níveis de maturidade (vide Ilustração 4) (SEI, 2006):

Ilustração 4: Níveis de maturidade do CMMI.

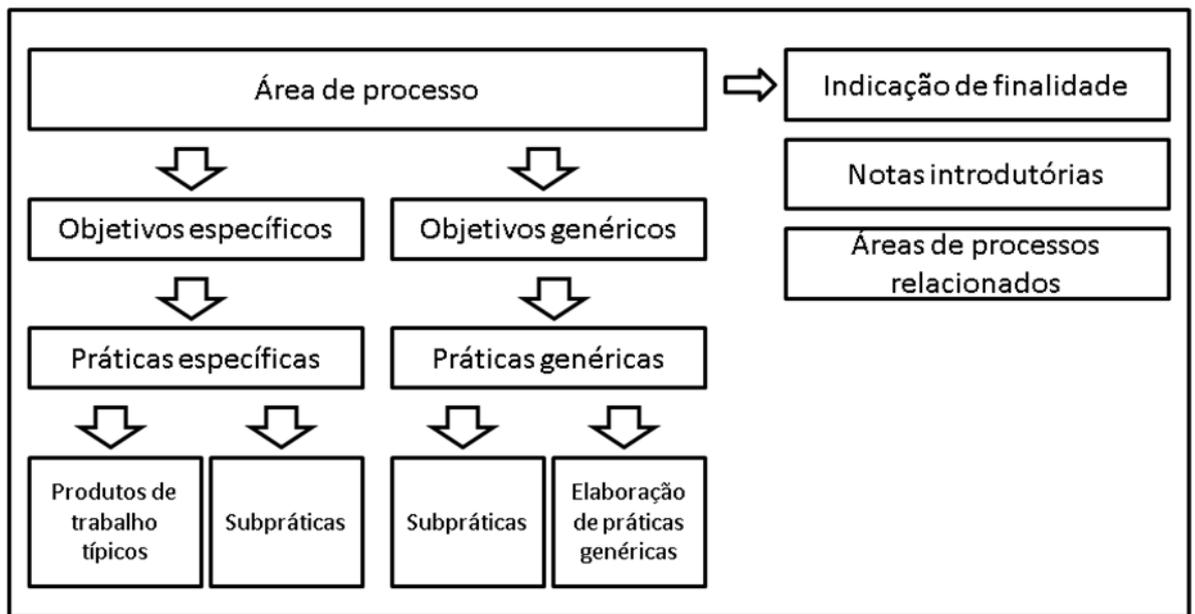


Fonte: SEI (2006)

Cada nível de maturidade é formado por áreas de processo, cada uma contemplando diversas práticas. Cada área de processo possui objetivos a serem alcançados e práticas que auxiliam na busca por esses objetivos.

Para melhor compreensão do CMMI, faz-se necessária a apresentação dos elementos que compõem o modelo, conforme a Ilustração 5.

Ilustração 5: Componentes do Modelo CMMI-DEV.



Fonte: SEI (2006)

Para cada uma das áreas de processos são indicados os objetivos genéricos e específicos, bem como as práticas genéricas e específicas e os produtos de trabalhos típicos. Os componentes do modelo são descritos na sequência, de acordo com SEI (2006).

Áreas de processos correspondem a um grupo de práticas relacionadas em uma área. Quando essas práticas são executadas de maneira coletiva, elas satisfazem as metas consideradas importantes para realizar melhorias significativas na área a que pertencem. Todas as áreas de processos do CMMI são as mesmas tanto para a representação contínua quanto para a representação por estágios.

Objetivos específicos se aplicam a uma área de processo e identificam características únicas que descrevem o que deve ser implementado para satisfazer a área de processo. Os objetivos específicos são utilizadas nas avaliações para auxiliar a determinar se uma área de processo foi satisfeita ou não.

Práticas específicas é uma atividade importante para satisfazer o objetivo específico associado. As práticas específicas descrevem as atividades que se espera que resultem no atendimento de objetivos específicos de uma área de processo.

Produtos de trabalho típicos são componentes informativos do modelo que oferecem exemplos de saídas de uma prática específica ou genérica. Esses exemplos são chamados “produtos de trabalho típicos” porque, existem outros produtos de trabalho que são tão eficientes quanto esses, mas que não estão listados.

Subpráticas são descrições detalhadas que fornecem um direcionamento para a interpretação de práticas específicas ou genéricas. As subpráticas podem ser expressas como se fossem exigidas, mas são, na verdade, componentes informativos dos modelos CMMI criados somente para fornecer ideias que podem ser úteis na melhoria dos processos.

Objetivos genéricos são chamados assim porque a mesma declaração de meta aparece em diversas áreas de processos. Na representação em estágios, cada área de processo tem somente um objetivo genérico. A satisfação de um objetivo genérico em uma área de processo significa um controle melhorado do planejamento e da implementação de processos associados com aquela área de processo, indicando, portanto, se esses processos parecem ser eficientes, repetíveis e duráveis. Os objetivos genéricos são utilizados em avaliações para determinar se uma área de processo está sendo satisfeita.

Práticas genéricas oferecem uma institucionalização que assegura que os processos associados com a área de processo serão eficientes, repetíveis e duráveis. As práticas genéricas são categorizadas pelos objetivos genéricos e características comuns.

Elaboraões de práticas genéricas são componentes informativos do modelo que aparecem em cada área de processo para fornecer instruções sobre como as práticas genéricas deverão ser aplicadas de forma única naquela área de processo.

2.4.1.1 Verificação e Validação no CMMI

A separação das áreas de processo verificação e validação no modelo CMMI tem como objetivo tornar possível uma análise exaustiva à correta execução de ambos os conceitos. A diferença entre ambos pode ser identificada através do âmbito da ação (testes, revisões, inspeções, entre outros) (GROSSO, 2006).

Verificação: Em relação a área de processo de verificação, esta insere-se no nível três de maturidade do modelo estagiado do CMMI. O seu objetivo é assegurar que os produtos de trabalho selecionados vão ao encontro dos requisitos especificados.

A área de processo de verificação envolve: a preparação da verificação, a execução da verificação e a identificação das ações corretivas. A verificação é um processo incremental, porque deve ocorrer durante todo o desenvolvimento do produto e dos produtos de trabalho, começando com a verificação dos requisitos, em seguida passando para a verificação dos produtos de trabalho em desenvolvimento e na verificação do produto completo (SEI, 2006).

As práticas específicas apresentadas pelo modelo são as seguintes (Quadro 1):

Quadro 1: Práticas específicas do processo de verificação.

Objetivos	Práticas	Descrição
O.E. 1 - Preparar Verificação: É feita a preparação da verificação.	P.E. 1.1 - Selecionar produtos de trabalho para a verificação	Selecionar os produtos de trabalho a serem verificados e os métodos de verificação a serem utilizados para cada um deles.
	P.E. 1.2 - Estabelecer o Ambiente de Verificação	Estabelecer e manter o ambiente necessário para suportar a verificação.
	P.E. 1.3 - Estabelecer procedimentos e critérios de verificação	Estabelecer e manter procedimentos e critérios de verificação para os produtos de trabalho selecionados.
O.E. 2 - Realizar Revisões Técnicas (<i>peer reviews</i>): São realizadas revisões técnicas aos produtos de trabalho selecionados.	P.E. 2.1 - Preparar as revisões técnicas	Preparar as revisões técnicas dos produtos de trabalho selecionados.
	P.E. 2.2 - Realizar revisões técnicas	Conduzir revisões técnicas aos produtos de trabalho selecionados e identificar ocorrências resultantes da revisão técnica.
	P.E. 2.3 - Analisar dados das revisões técnicas	Analisar os dados relativos à preparação, realização e resultados das revisões técnicas.
O.E. 3 - Verificar os Produtos de Trabalho Selecionados: Os produtos de trabalho são verificados face aos seus requisitos	P.E. 3.1 - Executar verificação	Executar a verificação dos produtos de trabalho selecionados.
	P.E. 3.2 - Analisar resultados das verificações e identificar ações corretivas	Analisar os resultados de todas as atividades de verificação e identificar ações corretivas.

Fonte: SEI (2006)

Validação: A área de processo validação, tal como a verificação, insere-se no nível três de maturidade do modelo. O seu objetivo é demonstrar que o produto ou componente do produto se encontram conforme a utilização pretendida quando colocado no ambiente pretendido. As práticas específicas apresentadas pelo modelo são as seguintes (Quadro 2):

As atividades de validação podem ser aplicadas a todos os aspectos do produto, em qualquer dos seus ambientes pretendidos, como operação, treinamento, manufatura, manutenção e serviços de suporte. Os métodos empregados para executar a validação podem ser aplicados a produtos de trabalho, bem como a produtos e componentes de produtos. Quando questões de validação são identificadas, elas são referenciadas aos processos associados como as áreas de processos de desenvolvimento de requisitos, soluções técnicas ou monitoramento e controle do projeto para a resolução (SEI, 2006).

Quadro 2: Práticas específicas do processo de validação.

Objetivos	Práticas	Descrição
O.E. 1 - Preparar Validação: É conduzida a preparação da validação	P.E. 1.1 - Selecionar os produtos para a validação	Selecionar os produtos, os componentes do produto a serem validados e quais os métodos de validação a serem utilizados para cada um.
	P.E. 1.2 - Estabelecer ambiente de validação	Estabelecer e manter o ambiente necessário para suportar a validação.
	P.E. 1.3 - Estabelecer procedimentos e critérios de validação	Estabelecer e manter procedimentos e critério de validação.
O.E. 2 - Validar Produto ou Componentes do Produto: O produto ou componentes do produto são validados,	P.E. 2.1 - Realizar validação	Realizar a validação dos produtos ou componentes do produto selecionados.
	P.E. 2.2 - Analisar resultados da validação	Analisar os resultados das atividades de validação e identificar ocorrências.

Fonte: SEI (2006)

Com a análise da execução das áreas de processo de verificação e validação, é avaliada uma parte fundamental do ciclo de vida de desenvolvimento do produto. Quanto mais ações de verificação e validação forem executadas, maior é a probabilidade do produto resultante sair com melhor índice de qualidade (GROSSO, 2006).

2.4.2 Melhoria de Processo do Software Brasileiro (MPS.BR)

O MPS.BR está em desenvolvimento desde 2003 e tem como objetivo definir um modelo de melhoria e avaliação de processo de software, preferencialmente para as micro, pequenas e médias empresas, de forma a atender às suas necessidades de negócio e a ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software (SOFTEX, 2009).

De acordo com SOFTEX (2009) o MR.MPS possui sete níveis de maturidade. Estes níveis estão apresentados na Quadro 3.

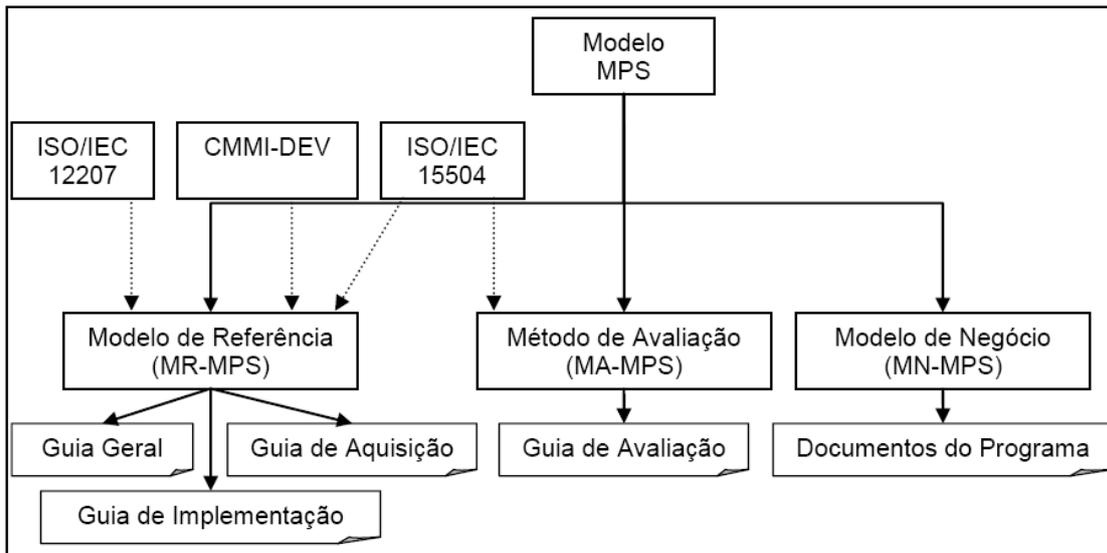
Quadro 3: Níveis de maturidade do MR.MPS.

Nível	Processos
A	
B	Gerência de Projetos
C	Gerência de Riscos
	Desenvolvimento para Reutilização
	Gerência de Decisão
D	Verificação
	Validação
	Projeto e Construção do Produto
	Integração do Produto
	Desenvolvimento de Requisitos
E	Gerência de Projetos
	Gerência de Reutilização
	Gerência de Recursos Humanos
	Definição do Processo Organizacional
	Avaliação e Melhoria do Processo Organizacional
F	Medição
	Garantia da Qualidade
	Gerência de Portfólio de Projetos
	Gerência de Configuração
	Aquisição
G	Gerência de Requisitos
	Gerência de Projetos

Fonte: SOFTEX (2009).

A base utilizada para a construção do MPS.BR é composta das normas NBR ISO/IEC 12207 - Processo de Ciclo de Vida de Software e ISO/IEC 15504 - Avaliação de Processo e seu Modelo de Avaliação de Processo de Software. O MPS.BR também cobre o conteúdo em relação aos processos da Norma NBR ISO/IEC 12207, conforme a Ilustração 6 (SOFTEX, 2009).

Ilustração 6: Estrutura do modelo de referência MPS.



Fonte: SOFTEX (2009)

- O **Modelo de Referência** de Melhoria de Processo de Software (MR-MPS) tem sete (7) níveis de maturidade, os quais possibilitam uma implantação mais gradual e adequada à micro, pequena e média empresa, além disto, as avaliações considerando mais níveis permitem maior visibilidade dos resultados de melhoria de processo, com prazos mais curtos e compatibilidade com CMMI.
- O **Método de Avaliação** - MA-MPS contém o processo de avaliação, os requisitos para os avaliadores e os requisitos para averiguação da conformidade ao modelo MR-MPS. Ele está descrito de forma detalhada no Guia de Avaliação e foi baseado na norma ISO/IEC 15504.
- O **Modelo de Negócio** - MN-MPS contém uma descrição das regras para a implementação do MR-MPS pelas empresas de consultoria, de software e de avaliação. O detalhamento dessas regras está disponível na página do SOFTEX (2009).

Os processos no MR-MPS são descritos em termos de propósito e resultados, sendo que propósito descreve o objetivo geral a ser atingido durante a execução do processo. Os resultados esperados do processo estabelecem os resultados a serem obtidos com a efetiva implementação do processo. A capacidade do processo é representada por um conjunto de atributos de processo descrito em termos de resultados

esperados. A capacidade do processo expressa o grau de refinamento e institucionalização com que o processo é executado na organização.

2.4.2.1 Verificação e Validação no MPS.BR

Os processos de verificação e validação estão definidos no nível D (Largamente definido) do guia geral (SOFTEX, 2009).

Verificação: O propósito do processo Verificação é confirmar que cada serviço e/ou produto de trabalho do processo ou do projeto reflete apropriadamente os requisitos especificados (SOFTEX, 2009).

O processo Verificação trata como avaliar produtos de trabalho e serviços, garantindo que os mesmos atendem a seus requisitos, através da identificação dos itens a serem verificados, do planejamento da verificação de cada um destes itens e da execução da verificação conforme planejado ao longo do desenvolvimento do produto. O objetivo da verificação é determinar se os produtos de software de uma atividade atendem completamente aos requisitos ou condições impostas a eles nas atividades anteriores. A norma também apresenta um processo de verificação que pode ser usado como referência auxiliar na implementação deste processo e na interpretação de seus resultados esperados, conforme Quadro 4.

Quadro 4: Processo de verificação no MR.MPS.

VER1	Produtos trabalho a serem verificados são identificados.	Devemos analisar os produtos que serão produzidos e identificar qual deverá ser verificado. Os produtos que devem ser verificados são selecionados levando em conta sua contribuição para se atingir os requisitos, e levando em conta os riscos do mesmo para o projeto.
VER2	Uma estratégia de verificação é desenvolvida e implementada estabelecendo cronograma, revisores envolvidos, métodos para verificação e qualquer material a ser utilizado na verificação.	Criar uma estratégia de verificação, descrevendo os procedimentos, a infra-estrutura e as responsabilidades.
VER3	Critérios para verificação dos produtos de trabalho a serem verificados são identificados e um ambiente para verificação é estabelecido.	Definir todos os critérios e procedimentos a serem utilizados na verificação, disponibilizando toda infra-estrutura necessária para a verificação de cada produto.
VER4	Atividades de verificação, incluindo testes e revisões por pares, são executadas.	Garantir que as atividades de verificação foram realizadas da forma planejada.
VER5	Defeitos são identificados, registrados e ações corretivas são realizadas.	Documentação e registro de todos os defeitos identificados durante as atividades de verificação. Classificar os defeitos, analisando-os e identificando os que devem ser removidos do sistema.
VER6	Resultados de atividades de verificação são analisados e disponibilizados para os interessados.	Analisar os resultados obtidos nas atividades de verificação e disponibilizá-las aos clientes. Analisando os laudos de avaliação e os relatórios de testes.

Fonte: SOFTEX (2009).

Validação: O propósito do Processo Validação é confirmar que um produto ou componente do produto atenderá a seu uso pretendido quando colocado no ambiente para o qual foi desenvolvido (SOFTEX, 2009). O processo de validação orienta como avaliar a qualidade de um produto ou componente de produto, garantindo que este atenda às necessidades de seus usuários, quando colocado em seu ambiente de uso. O objetivo da validação é garantir que o produto correto está sendo desenvolvido.

Quadro 5: Processo de validação no MR.MPS.

VAL1	Produtos de trabalho a serem validados são identificados.	Identificação dos produtos a serem validados durante o projeto. A escolha de tais componentes devem levar em conta as necessidades do cliente, assim como os riscos de cada componente analisado.
VAL2	Uma estratégia de validação é desenvolvida e implementada estabelecendo cronograma, participantes envolvidos, métodos para validação e qualquer material a ser utilizado na validação.	Garantia de que as atividades de validação sejam planejadas. Devem ser identificados também os métodos a serem utilizados nesta atividade.
VAL3	Critérios para validação dos produtos de trabalho a serem validados são identificados e um ambiente para validação é estabelecido.	Garantir que os critérios e procedimentos na fase de validação foram identificados, e existe um ambiente para os mesmos.
VAL4	Atividades de validação são executadas para garantir que os produtos de software estão prontos para uso no ambiente operacional pretendido.	Ter a certeza de que as atividades de validação foram realizadas seguindo o planejamento, e não ter dúvidas de que o produto esta pronto para ser utilizado no ambiente operacional.
VAL5	Problemas são identificados, registrados e ações corretivas são realizadas.	Garantir que todos os problemas detectados na validação foram identificado e documentados, deve-se também definir quais serão tratados.
VAL6	Resultados de atividades de validação são analisados e disponibilizados para as partes interessadas.	Depois de verificar que o produto satisfaz os requisitos descritos devemos realizar testes em seu ambiente de produção.
VAL7	Evidências de que os produtos de software desenvolvidos estão prontos para o uso pretendido são fornecidas.	Analisar todos os resultados obtidos com a validação do projeto disponibilizado ao cliente. É feita por meio de laudos resultantes do processo de avaliação e relatórios dos testes.

Fonte: SOFTEX (2009).

2.4.3 Análise comparativa das práticas

Pode-se destacar que os dois modelos comparados possuem processos de Verificação e Validação (V&V).

Quadro 6: Análise comparativa das práticas.

Atividades	Processos	
	CMMI-DEV	MR.MPS
Verificação		
Preparar verificação	P.E. 1.1 Selecionar produtos de trabalho para a verificação	VER1 Produtos de trabalho a serem verificados são identificados
	P.E. 1.2 Estabelecer o Ambiente de Verificação	VER2 Uma estratégia de verificação é desenvolvida e implementada
	P.E. 1.3 Estabelecer procedimentos e critérios de verificação	VER3 Critérios para verificação dos produtos de trabalho a serem verificados são identificados e um ambiente para verificação é
Revisões	P.E. 2.1 Preparar as revisões técnicas	VER4 Atividades de verificação, incluindo testes e revisões por pares, são executadas
	P.E. 2.2 Realizar revisões técnicas	
	P.E. 2.3 Analisar dados das revisões técnicas	
Verificar os produtos de trabalho	P.E. 3.1 Executar verificação	VER5 Defeitos são identificados, registrados e ações corretivas são realizadas
	P.E. 3.2 Analisar resultados das verificações e identificar ações corretivas	VER6 Resultados de atividades de verificação são analisados e disponibilizados para os interessados
Validação		
Preparar validação	P.E. 1.1 Selecionar os produtos para a validação	VAL1 Produtos de trabalho a serem validados são identificados
	P.E. 1.2 Estabelecer ambiente de validação	VAL2 Uma estratégia de validação é desenvolvida e implementada
	P.E. 1.3 Estabelecer procedimentos e critérios de validação	VAL3 Critérios para validação dos produtos de trabalho a serem validados são identificados e um ambiente para validação é estabelecido
Validar produtos ou componentes do produto	P.E. 2.1 Realizar validação	VAL4 Atividades de validação são executadas
		VAL5 Problemas são identificados, registrados e ações corretivas são realizadas
	P.E. 2.2 Analisar resultados da validação	VAL6 Resultados de atividades de validação são analisados e disponibilizados para as partes interessadas
		VAL7 Evidências de que os produtos de software desenvolvidos estão prontos para o uso

Fonte: Elaborado pelo autor.

As atividades “Preparar verificação e Preparar validação” aparecem em todos os processos estudados, sua principal finalidade é a identificação e seleção dos produtos de trabalho a serem verificados e validados.

2.5 INSPEÇÕES DE SOFTWARE

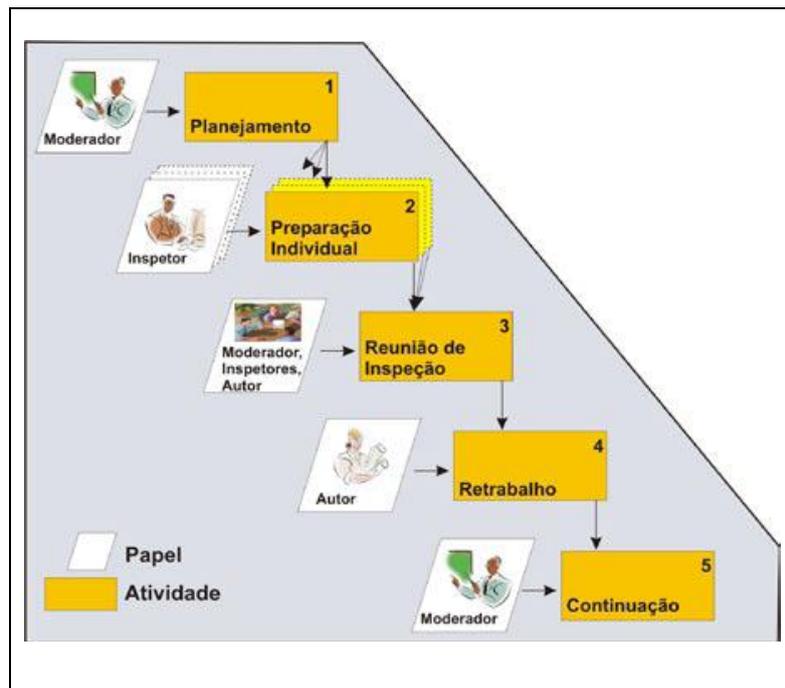
Inspeção de software é um tipo particular de revisão que pode ser aplicado a todos os artefatos de software e possui um processo de detecção de defeitos rigoroso e bem definido (KALINOWSKI & SPÍNOLA, 2007).

As inspeções de software não requerem que o programa seja executado, assim pode ser utilizadas como uma técnica de verificação antes que os programas sejam implementados. Durante uma inspeção, examina-se a representação original de um sistema, que pode ser um modelo de sistema, uma especificação ou um código de linguagem de alto nível (SOMMERVILLE, 2003).

As inspeções provaram ser uma técnica eficaz de detecção de erros. Os erros podem ser encontrados de um modo mais barato, por meio da inspeção, do que com extensivos programas de testes. Segundo Sommerville (2003), isso foi demonstrado em um experimento feito por Basili e Selby, que empiricamente compararam a eficácia das inspeções e dos testes. Eles constataram que a revisão estática de código era mais eficaz e menos dispendiosa que os testes de defeitos para descobrir defeitos em um programa.

O processo tradicional de inspeção de software, uma forma detalhada de se realizar uma revisão. Neste processo, existem seis atividades principais, conforme Ilustração 7.

Ilustração 7: Processo de inspeção de software.



Fonte: Kalinowski e Spínola (2007)

- **Planejamento:** Um usuário, desempenhando o papel de moderador da inspeção, define a descrição da inspeção, técnica a ser utilizada na detecção de defeitos, documento a ser inspecionado, entre outros.
- **Apresentação:** Os autores dos artefatos a serem inspecionados apresentam as características dos artefatos. Esta fase pode ser excluída se os inspetores possuem conhecimento sobre o projeto e os artefatos que devem ser inspecionados, por ser opcional, não está representada na figura.
- **Preparação:** Os inspetores analisam os artefatos individualmente, e fazem anotações produzindo uma lista de discrepâncias.
- **Reunião:** Uma reunião em equipe, envolvendo o moderador, os inspetores e os autores do documento. Discrepâncias são discutidas, e classificadas como defeito. A decisão final sobre a classificação de uma discrepância sendo discutida é do moderador. A solução dos defeitos não é discutida durante a reunião.
- **Retrabalho:** O autor corrige os defeitos encontrados pelos inspetores e confirmados pelo moderador.

- **Continuação:** O material corrigido pelos autores é repassado para o moderador, que faz uma análise da inspeção como um todo e re-avalia a qualidade do artefato inspecionado. Ele tem a liberdade de decidir se uma nova inspeção deve ocorrer ou não.

O objetivo de inspeções de software é melhorar a qualidade de artefatos de software através de sua análise, detectando e removendo defeitos antes que o artefato seja passado para a próxima fase do processo de desenvolvimento de software.

2.6 TESTES DE SOFTWARE

Antes de iniciar uma discussão sobre teste de software precisam-se esclarecer alguns conceitos relacionados a essa atividade.

Defeito é um ato inconsistente cometido por um indivíduo ao tentar entender uma determinada informação, resolver um problema ou utilizar um método ou uma ferramenta. Por exemplo, uma instrução ou comando incorreto (DIAS NETO, 2007).

Erro é uma manifestação concreta de um defeito num artefato de software. Diferença entre o valor obtido e o valor esperado, ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução de um programa constitui um erro (DIAS NETO, 2007).

Falha é o comportamento operacional do software diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos erros e alguns erros podem nunca causar uma falha (DIAS NETO, 2007).

Defeitos fazem parte do universo físico e são causados por pessoas, por exemplo, através do mal uso de uma tecnologia. Defeitos podem ocasionar a manifestação de erros em um produto, ou seja, a construção de um software de forma diferente ao que foi especificado. Por fim, os erros geram falhas, que são comportamentos inesperados em um software que afetam diretamente o usuário final da aplicação e pode inviabilizar a utilização de um software.

Dessa forma, o teste de software revela simplesmente falhas em um produto. A atividade de teste é composta por alguns elementos essenciais que auxiliam na formalização desta atividade. Esses elementos são os seguintes (DIAS NETO, 2007):

- Caso de Teste descreve uma condição particular a ser testada e é composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado (DIAS NETO, 2007).
- Critério de Teste serve para selecionar e avaliar casos de teste de forma a aumentar as possibilidades de provocar falhas ou, quando isso não ocorre, estabelecer um nível elevado de confiança na correção do produto (ROCHA, MALDONADO, WEBER, 2001).

Durante o processo de desenvolvimento de um software existem atividades que procuram garantir a qualidade do produto final, entretanto, apesar dos métodos, técnicas e ferramentas utilizadas, falhas no produto ainda podem ocorrer. Assim, a etapa de teste, a qual representa uma das atividades de garantia de qualidade, é de grande importância para a identificação e eliminação de falhas, representando assim o último passo do desenvolvimento de software (INTHURN, 2001).

Os defeitos existentes nos software, na maior parte das vezes, constituem-se em riscos tanto para o negócio quanto para a imagem da empresa. O objetivo de um processo de teste é minimizar os riscos causados por defeitos provenientes do processo de desenvolvimento.

2.6.1 Categorias de Testes de Software

Sabe-se que o objetivo de todo teste é detectar os erros existentes no produto de software, porém a realização de todas as categorias de testes descrita a seguir não reflete a realidade vivenciada pelas equipes de teste. A realização de todas estas categorias demandaria um esforço muito grande, então se faz necessário selecionar as categorias a utilizar conforme o perfil do produto (BARTIÉ, 2002).

a) Testes de Funcionalidade

Para Bartié (2002), o direcionamento dos testes funcionais deve ser realizado com base nos documentos de especificação funcional. Este documento descreve o comportamento da aplicação nos diversos cenários existentes para cada requisito de negócio. A principal característica desta categoria de testes é o grande foco nos negócios, com isso, garante-se que não exista diferença entre os requisitos funcionais e o comportamento do software construído. Esta categoria cobre as seguintes situações: pré-condições de uma transação; o fluxo de dados de uma transação de negócios; o

cenário primário de uma transação de negócios; os cenários alternativos de uma transação de negócios; os cenários de exceção de uma transação de negócios; e as pós-condições de uma transação de negócios.

b) Testes de Usabilidade

Para Rios e Moreira (2003), os testes de usabilidade têm como meta simular as condições de uso da aplicação sob a perspectiva do usuário. Sendo assim, este tipo de teste tem como foco verificar a facilidade de navegação entre as telas da aplicação, a clareza dos textos e das mensagens apresentadas ao usuário, o volume reduzido de interações para realizar uma determinada função, o padrão visual e etc. Esta categoria de testes pode ser realizada da seguinte forma: avaliar a facilidade de navegação entre as telas da aplicação; realizar operações e desfazê-las; realizar operações inválidas e avaliar as mensagens de alerta; avaliar quantos passos são necessários para executar as principais operações; verificar a existência de ajuda em todas as telas; e verificar se as informações presentes nos manuais de ajuda são coerentes.

c) Testes de Carga

Conforme Bastos (2006), os testes de carga servem basicamente para avaliar como uma aplicação e toda sua infraestrutura se comportam quando se aplica um volume de transações superiores aos volumes máximos previstos para esta aplicação e quando se aplica variações bruscas de transações. Este tipo de teste pode ser executado da seguinte forma: aumentando e diminuindo sucessivamente a quantidade de transações simultâneas; aumentando e diminuindo o tráfego de rede; aumentando a quantidade de usuários simultâneos; e realizando uma combinação de todos estes itens anteriores.

d) Testes de Volume

Segundo Bartié (2002), os testes de volume são caracterizados por determinar os limites de processamento e a carga da aplicação e de toda a sua infra-estrutura. Este tipo de teste é conduzido de forma contínua, aumentando o volume das operações realizadas com a aplicação, até que se atinja o limite máximo. O objetivo deste teste é conhecer os limites da solução e avaliar se estão de acordo com as especificações. Para a realização deste teste pode-se: aumentar constantemente o volume de transações; aumentar

constantemente o volume de consultas; e aumentar constantemente o tamanho dos arquivos a serem processados.

e) Testes de Configuração

Para Molinari (2006), esta categoria de testes tem por finalidade executar o software sobre inúmeras configurações de software e de hardware. A meta deste teste é garantir que a solução funcione corretamente sobre os diversos ambientes previstos nas fases de levantamento de requisitos. A execução deste teste pode ser da seguinte forma: variando os sistemas operacionais; variando os hardwares; e combinando as variações citadas anteriormente.

f) Testes de Compatibilidade

A respeito dos testes de compatibilidade, Bartié (2002) afirma que esta categoria tem por finalidade executar a aplicação interagindo com as versões anteriores de outras aplicações ou dispositivos físicos. O objetivo deste teste é garantir que as novas versões de um software estão suportando antigas versões. Pode ser testado do seguinte modo: realizando a importação de dados gerados pela versão anterior; e realizando a comunicação com todas as versões de layout.

g) Testes de Segurança

Segundo Rios e Moreira Filho (2003), este tipo de teste tem como objetivo encontrar as falhas de segurança que podem comprometer a confidencialidade e a integridade das informações, provocar perdas de dados ou interrupções de processamento. Este tipo de teste pode ser executado da seguinte maneira: validar todos os requisitos de segurança identificados; tentar acessar funcionalidades que requerem determinados perfis; tentar invadir/derrubar o servidor de dados; e tentar extrair *backups* de informações e etc.

h) Testes de Desempenho

Conforme Bastos *et al* (2006), esta categoria de testes tem por finalidade determinar se o desempenho da aplicação está coerente com os requisitos definidos, quando se obtém situações em que o pico máximo de acesso e concorrência é atingido. Para se executar este tipo de teste, pode-se: validar todos os requisitos de desempenho

identificados; simular vários usuários acessando a mesma informação de forma simultânea; simular vários usuários processando a mesma transação de forma simultânea; simular percentagem de tráfego de rede; e combinar todos os itens apresentados anteriormente.

i) Testes de Confiabilidade e Disponibilidade

Para Rios e Moreira Filho (2003), a principal característica deste teste é monitorar uma aplicação por um determinado tempo e avaliar o nível de confiabilidade da arquitetura da solução. Este tipo de teste pode ser executado da seguinte maneira: monitorando a execução da aplicação; identificando todas as interrupções do ambiente; e identificando o tempo de interrupção do ambiente.

j) Testes de Recuperação

Os testes de recuperação, segundo Bartié (2002), tem por finalidade avaliar o comportamento da aplicação após a ocorrência de um erro e verificar sua recuperação. Esses testes podem ser realizados da seguinte forma: interrompendo o acesso à rede por alguns instantes e por um longo período; interrompendo o processamento, desligando o micro e o servidor; e gerando arquivos e posteriormente cancelando o processamento.

k) Testes de Instalação

O teste de instalação tem como objetivo validar o procedimento de instalação da aplicação, de forma que esta atenda a todas as necessidades apresentadas nos requisitos. É recomendado que este teste seja realizado pelo próprio usuário (BASTOS *et al*, 2006).

l) Teste de Regressão

Conforme BSTQB (2007), o teste de regressão é o teste repetido de um programa que sofreu alguma mudança após uma etapa de testes, tem como objetivo descobrir a existência de algum defeito introduzido ou não identificado originalmente como resultado da mudança, sendo realizado quando o software ou o ambiente são modificados. BSTQB (2007) ainda define que este tipo de teste pode ser realizado em todos os níveis de teste, podendo ser aplicado aos testes funcionais, não funcionais e estruturais. Os testes de regressão são executados muitas vezes, por esta razão deve ser automatizado.

De acordo com Pressman (2006), o teste de regressão é uma atividade que ajuda a garantir que as modificações no produto de software não introduzam um comportamento indesejável ou erros.

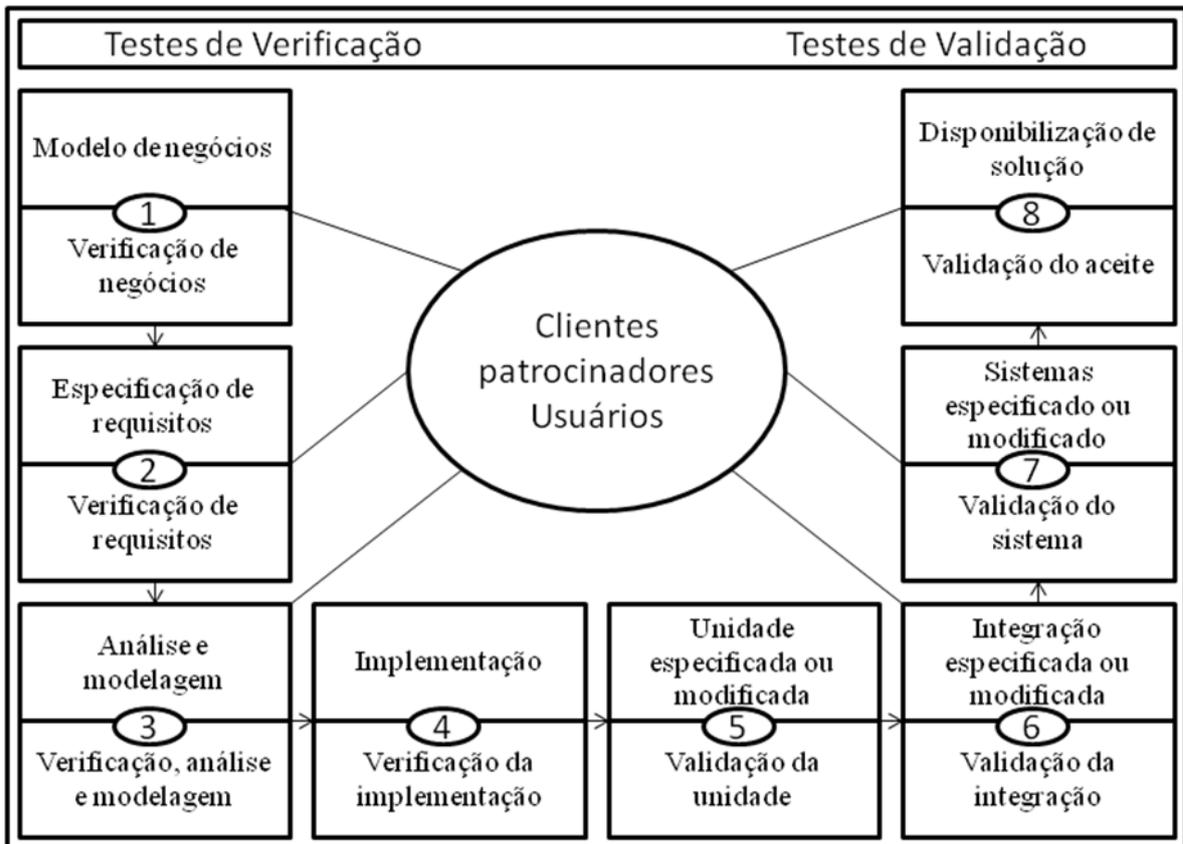
2.6.2 Testes na Verificação e Validação

O processo de qualidade de software está decomposto em fases. Cada fase tem uma sequência de execução dos testes a serem aplicados. Os testes de verificação visam garantir o processo de engenharia de software, enquanto os testes de validação estão focados na garantia da qualidade do produto de software, conforme Ilustração 7 (BARTIÉ, 2002).

Bartié (2002) cita:

Testes de verificação e validação são complementares. Em hipótese alguma, estes deverão ser encarados como atividades redundantes. Tanto um quanto o outro possui naturezas e objetivos distintos, fortalecendo o processo de detecção de erros e aumentando a qualidade final do produto. Um bom processo de qualidade de software deverá potencializar essas duas formas de testes de modo que os esforços sejam minimizados e os resultados sejam os mais positivos possíveis (p. 37).

Ilustração 8: Processo de qualidade



Fonte: Bartié (2002)

Os testes de verificação podem ser entendidos como um processo de auditoria das atividades e avaliação de documentos gerados durante todas as fases do processo de engenharia de software. Conforme Bartié (2002) as verificações devem ser aplicadas a todos os produtos (documentos, gráficos, manuais, código-fonte) que são produzidas em cada etapa do processo, evitando que dúvidas passem para a próxima fase.

Os testes de validação podem ser entendidos como um processo formal de avaliação de produtos tecnológicos que podem ser aplicados em componentes isolados, módulos existentes ou mesmo nos sistemas como um todo. Seu objetivo é avaliar a conformidade do software com os requisitos e especificações das etapas iniciais do projeto.

2.7 REQUISITOS PARA UM PROCESSO

Para conseguir o nível de qualidade adequado aos processos e produtos de software a literatura apresenta normas e modelos de processos voltados à V&V. São

apresentados também os requisitos considerados mínimos, que um processo de referência deve possuir, para ser utilizado nas organizações de desenvolvimento de software. Estes são apresentados na sequência.

Segundo Hauck, Wangenheim e Thiry (2007) um processo de Verificação e Validação deve ser alinhado aos principais modelos de referência, considerando os processos:

- Áreas de Processo de Verificação e Validação do CMMI-DEV V1.2; e
- Verificação e Validação do MPS.BR 2009.

Um processo, para que possa ser seguido com sucesso por uma organização de software deve apresentar um conjunto mínimo de atributos. Estes são apresentados em (HAUCK, 2007):

- **Custo:** o processo deve ser livre para utilização e adaptação sem custos. Ele deve, preferencialmente, indicar ferramentas e técnicas livres;
- **Simplicidade:** as indicações descritas no processo (técnicas, processos, etc.) devem ser simples, de forma a reduzir a complexidade da implantação;
- **Facilidade de compreensão:** o processo deve ser fácil de entender, não exigindo conhecimento profundo na área de gerência de projetos;
- **Facilidade de implantação:** o processo deve indicar oportunidades de automatizar o processo de monitoração e controle para reduzir o esforço da implantação;
- **Detalhamento:** o processo deve fornecer descrições em um nível de detalhe suficiente, que torne possível a execução das atividades;
- **Escopo:** o processo deve fornecer suporte para todo o processo de monitoração e controle, incluindo vários artefatos que auxiliem a execução do processo;
- **Adaptabilidade:** o processo deve facilitar sua adaptação aos diversos tipos de projetos e organizações.

Não foram encontrados modelos, normas ou guias que contemplem completamente os processos de V&V, estejam alinhados aos principais modelos e normas de referência (CMMI e MPS.BR) e atendam aos requisitos especificados para um processo. Desta forma propõe-se um novo processo de V&V que atenda a estes requisitos, ele é apresentado no capítulo seguinte.

3 DESENVOLVIMENTO

Realizou-se uma análise de dois dos principais modelos e normas de referência para qualidade de processos: o CMMI-DEV (SEI, 2006) e o MPS. BR (SOFTEX, 2009), propondo a descrição de um novo processo de V&V que esteja alinhado à estas normas e modelos.

Desta forma, espera-se que a existência de um processo de V&V, aderente às principais normas e modelos de processo auxilie na implementação de um processo sistemático de testes, colaborando para a redução de defeitos encontrados em um produto de software.

3.1 ESTRUTURA DO PROCESSO

Processos de Software podem ser definidos em níveis de detalhe, instanciados a partir de processo, padrão organizacional e modelos de referência ou normas. Conforme Hauck *et al* (2008) um processo de referência apresenta ao menos os seguintes itens:

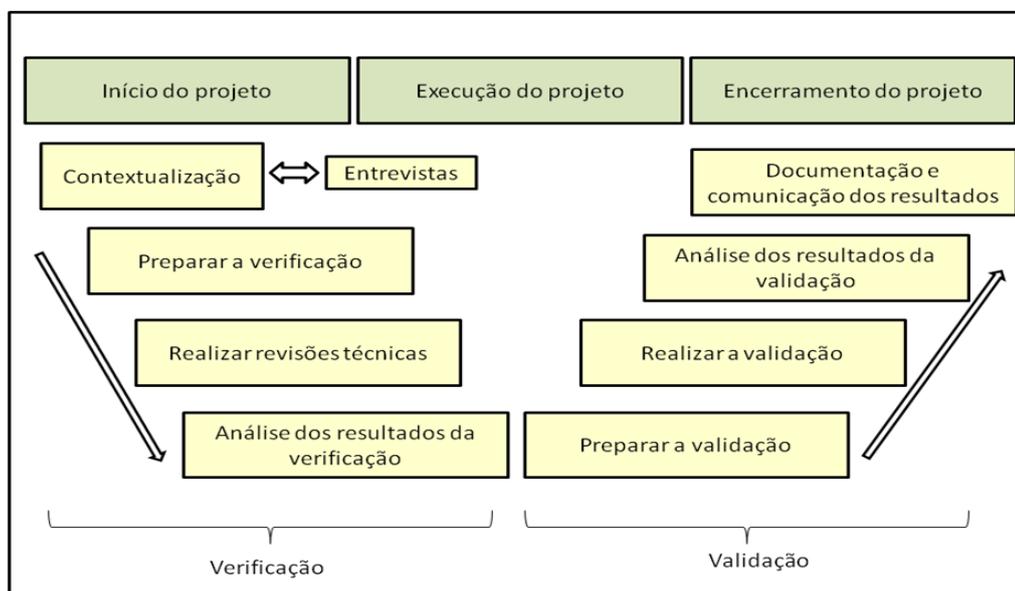
- **Apresentação:** é a parte inicial do processo, onde é exibido um diagrama que delimita o processo de monitoração e controle de projetos de software em relação aos demais processos relacionados a ele;
- **Fundamentação:** são apresentados os conceitos fundamentais sobre monitoração e controle de projetos para auxiliar no entendimento;
- **Avaliação Inicial do processo:** este item constitui-se normalmente no primeiro passo para a melhoria do processo;
- **Processo de Referência:** apresenta o conjunto de atividades agrupadas em sub-processos e inter-relacionadas para formar um processo de monitoração e controle;
- **Melhores Práticas:** são apresentados os textos compilados do processo de monitoração e controle das normas e modelos de referência.

3.2 ATIVIDADES DO PROCESSO

O conjunto de atividades definidas para o processo foi identificado a partir de práticas definidas nos processos e modelos estudados durante este trabalho. Além destas atividades, foram identificadas algumas necessidades complementares para auxiliar em todo o processo de V&V.

Nesta etapa são apresentadas as atividades propostas para o processo de V&V (Ilustração 9), definindo, para cada uma, o detalhamento necessário para sua execução.

Ilustração 9: Processo de referência.



Fonte: Elaborado pelo autor.

3.2.1 Definição da Equipe de Trabalho

Neste tópico descrevemos exemplos de papéis assumidos por integrantes das equipes de testes.

A equipe de trabalho foi dividida nas seguintes responsabilidades:

- Líder ou Gerente (L): Responsável pela liderança de um projeto de teste específico.
- Arquiteto (A): Responsável pela montagem do ambiente de teste (infraestrutura) e escolha das ferramentas.

- Analista (AN): Responsável pela modelagem e elaboração dos casos de teste e scripts de teste.
- Testador (TE): Responsável pela execução dos casos de teste e scripts de teste.

Toda a equipe de trabalho desempenha um papel na fase do projeto. O Quadro 7 lista as etapas e atividades de cada membro da equipe.

Quadro 7: Matriz de responsabilidades.

Matriz de responsabilidades					
Etapas	Atividades	L	A	AN	TE
Planejamento	Contextualização	E	P		
Preparação	Preparar a verificação	A	E	P	
	Realizar revisões técnicas	A	P	E	
	Análise dos resultados de verificação	A	E	A	
Especificação/Execução	Preparar a validação	A	P	E	
	Realizar a validação				E
	Análise dos resultados da validação	A			E
Entrega	Documentação e comunicação dos resultados	E	A		

Responsabilidades	
E	Execução das atividades
P	Participa na execução (co-responsável)
A	Acompanha/é informado dos resultados

Fonte: Elaborado pelo autor.

3.2.2 Contextualização: Identificação das informações sobre a organização/projeto

Nesta etapa, são feitos os levantamentos de dados da organização/projeto. O levantamento das informações da organização/projeto e seus objetivos contribuem para identificar técnicas/ferramentas que serão necessárias para V&V.

3.2.2.1 Realização das entrevistas de levantamento

As entrevistas com participantes e partes interessadas do projeto podem auxiliar na preparação da V&V. As entrevistas são uma das principais fontes de coleta de dados sobre identificação dos requisitos, regras, perspectivas e características.

Esta atividade aborda e documenta os requisitos do projeto e da entrega, seu limite e controle do escopo para cada fase. Dependendo das características de cada V&V, uma avaliação precisa ser executada caso a caso.

No Quadro 1 é apresentado os objetivos, cada passo da atividade e suas respectivas entradas e saídas.

Quadro 1: Processo de contextualização.

ATV-001 Contextualização	
DESCRIÇÃO	
1. OBJETIVO	
Nesta atividade devem-se realizar entrevistas com as pessoas envolvidas com o projeto para coletar dados sobre identificação dos requisitos e regras.	
2. PASSOS	
<ol style="list-style-type: none"> 1. Realizar entrevistas; <ol style="list-style-type: none"> 1.1. Deve-se agendar um horário com as pessoas envolvidas para fazer o levantamento das atividades e requisitos do projeto; 1.2. Realizar entrevistas com as pessoas envolvidas para identificar as reais necessidades do cliente referente ao projeto; 1.3. A entrevista deve ser documentada. 2. Validar as documentações registradas no passo 1, identificando se ficou alguma dúvida, se está de acordo com o projeto, etc; 3. Definir os requisitos; <ol style="list-style-type: none"> 3.1. Com as documentações elaboradas e aprovadas no passo 2, é possível elaborar os requisitos do projeto; 3.2. Identificar os requisitos como um todo estabelecendo um conjunto de objetivos gerais que o sistema deve cumprir e características do que o sistema deve fazer; 3.3. Elaborar um formulário com os requisitos do projeto. 4. Validar os requisitos identificados no passo 3, de acordo com a semântica do projeto; 5. Após toda a análise, dúvidas questionadas e levantamento dos requisitos é possível seguir para a próxima atividade “ATV-003 Preparar a verificação”. 	
3.1 ARTEFATOS(S) DE ENTRADA	
<ol style="list-style-type: none"> 1. Documentação geral do projeto. 	
3.2 ARTEFATOS(S) DE SAÍDA	
<ol style="list-style-type: none"> 1. Especificação de Requisitos; 2. Relatório descrevendo os problemas e os possíveis benefícios do projeto. 	
4. TÉCNICAS	
- > Entrevistas, Questionários, Brainstorming.	
5. PAPÉIS	

Líder ou gerente, Analista e pessoas envolvidas no projeto.

Fonte: Elaborado pelo autor.

3.2.3 Preparar a Verificação

Nesta etapa devem-se selecionar os produtos e métodos a serem verificados levando em conta sua contribuição para atingir os requisitos.

Este documento descreve o planejamento para execução dos testes do projeto, bem como os envolvidos, abrangência, recursos e critérios de aceitação dos artefatos a serem testados.

No Quadro 2 é apresentado os objetivos, cada passo da atividade e suas respectivas entradas e saídas.

Quadro 2: Processo de preparar a verificação.

ATV-002 Preparar a Verificação	
DESCRIÇÃO	
1. OBJETIVO	
Definir o plano de verificação para o projeto. O plano deve descrever as atividades de verificação, cronograma para executar as atividades e os responsáveis pelas atividades.	
2. PASSOS	
<ol style="list-style-type: none"> 1. Neste passo será elaborado o plano de verificação para o projeto; 2. Descrever as atividades de verificação; 3. Identificar os responsáveis por cada atividade de verificação; 4. Estabelecer um cronograma para executar as atividades; 5. Seleção de produtos para a verificação; <ol style="list-style-type: none"> 5.1. Nesta atividade devem ser analisados os produtos que serão produzidos ao longo do projeto; 5.2. Os produtos devem ser selecionados com base nas suas contribuições para o alcance dos objetivos, considerando também os riscos do projeto. 6. Seleção de técnicas e ferramentas; <ol style="list-style-type: none"> 6.1. Selecionar métodos e técnicas de verificação que estão disponíveis para uso e que serão aplicados a cada projeto; 6.2. Selecionar as ferramentas que serão usadas para apoiar a verificação. 7. Definir procedimentos e critérios de verificação; <ol style="list-style-type: none"> 7.1. Estabelecer e manter procedimentos e critérios para verificação. Os critérios são para garantir que os produtos estão de acordo com os requisitos; 	
3.1 ARTEFATOS(S) DE ENTRADA	
<ol style="list-style-type: none"> 1. Relatório descrevendo os problemas e os possíveis benefícios do projeto; 2. Especificação de Requisitos. 	

3.2 ARTEFATOS(S) DE SAÍDA

1. Plano de verificação.

4. TÉCNICAS

- > Entrevistas, Questionários, Brainstorming.

5. PAPÉIS

PERFIL

Arquiteto e Analista

Fonte: Elaborado pelo autor.

3.2.4 Realizar Revisões Técnicas

Realizar revisão dos requisitos identificados, os objetos e problemas do projeto, bem como o plano de verificação, e nesta revisão identificar se estão adequados, consistentes e testáveis. Analisar os dados relativos à preparação, realização e resultados das revisões técnicas.

No Quadro 3 é apresentado os objetivos, cada passo da atividade e suas respectivas entradas e saídas.

Quadro 3: Processo de realizar revisões técnicas.

ATV-003 Revisões técnicas	
DESCRIÇÃO	
1. OBJETIVO	Nesta etapa do processo é realizada uma revisão nos artefatos levantados até o momento, e nesta revisão identificar se as atividades estão adequadas, consistentes e testáveis.
2. PASSOS	<ol style="list-style-type: none">1. Neste passo deve ser realizada uma revisão dos artefatos levantados até o momento;2. Verificar o plano de verificação;<ol style="list-style-type: none">2.1. Avaliar o plano e realizar as adequações e ajustes necessários;2.2. Devem ser considerados os seguintes critérios: cronograma, viabilidade econômica, mão-de-obra e tecnológica;2.3. Qualquer inadequação identificada deve ser corrigida e relatada pelo líder ou gerente para poder obter o comprometimento dos <i>stakeholders</i> com o plano de verificação.3. Verificar o processo planejado;<ol style="list-style-type: none">3.1. Verificar se os processos definidos estão adequados às necessidades do projeto e padrões especificados e realizar os ajustes necessários;3.2. Qualquer inadequação identificada deve ser corrigida e relatada pelo líder ou gerente para poder obter o comprometimento dos <i>stakeholders</i> com o processo.4. Verificar os requisitos;<ol style="list-style-type: none">4.1. Avaliar, através de reuniões, os requisitos segundo os critérios: clareza, completeza, consistência com as necessidades de aquisição, viabilidade de operação e

<p>manutenção;</p> <p>4.2. Identificar os requisitos que não atendem aos critérios especificados e estabelecer as linhas para adequá-los.</p> <p>5. As ocorrências, problemas e possíveis benefícios do projeto devem ser documentos utilizando os seguintes modelos:</p> <p>5.1. Relatório descrevendo os problemas e os possíveis benefícios do projeto;</p> <p>5.2. Relatórios de ocorrências e registro de não-conformidades.</p>
<p>3.1 ARTEFATOS(S) DE ENTRADA</p> <p>1. Relatório descrevendo os problemas e os possíveis benefícios do projeto;</p> <p>2. Especificação de Requisitos.</p>
<p>3.2 ARTEFATOS(S) DE SAÍDA</p> <p>Relatório descrevendo os problemas e os possíveis benefícios do projeto;</p> <p>Relatórios de ocorrências e registro de não-conformidades.</p>
<p>4. TÉCNICAS</p> <p>- > Revisões de software.</p>
<p>5. PAPÉIS</p> <p>PERFIL</p> <p>Líder ou Gerente, Arquiteto e Analista.</p>

Fonte: Elaborado pelo autor.

3.2.5 Análise dos Resultados da Verificação

Analisar o resultado de todas as atividades de verificação e identificar as ações corretivas, garantindo que todas as atividades de verificação foram realizadas de forma planejada.

Documentar e registrar todos os defeitos identificados durante as atividades de verificação. Classificar os defeitos, analisando-os e identificando os que devem ser removidos do sistema e analisar os resultados obtidos nas atividades de verificação.

No Quadro 4 é apresentado os objetivos, cada passo da atividade e suas respectivas entradas e saídas.

Quadro 4: Processo de análise dos resultados da verificação.

ATV-004 Análise dos Resultados da Verificação	
DESCRIÇÃO	
1. OBJETIVO	
Nesta atividade é analisado o resultado de todas as atividades de verificação e, identificar as ações que devem ser tomadas.	
2. ATIVIDADES	
<ol style="list-style-type: none"> 1. Nesta atividade serão analisados os resultados obtidos na atividade ATV 003 – Revisões técnicas; 2. Verificar o projeto detalhado; <ol style="list-style-type: none"> 2.1. Avaliar o modelo de projeto detalhado segundo os critérios: clareza, completeza, rastreabilidade para os requisitos, consistência com o projeto, adequação dos métodos e padrões de projeto utilizados, viabilidade de testes. 2.2. Definir as linhas do plano de ação para adequar o modelo de projeto detalhado nos pontos onde este não atender aos critérios estabelecidos. 3. Realizar a verificação. <ol style="list-style-type: none"> 3.1. Realizar a verificação em todos os casos registrados nos relatórios anteriores; 3.2. Registrar as ocorrências encontradas seguindo padrão especificado no documento Relatório de testes. 	
3. ARTEFATOS	
3.1 ARTEFATOS(S) DE ENTRADA	
1. Relatórios de ocorrências e registro de não-conformidades.	
3.2 ARTEFATOS(S) DE SAÍDA	
1. Relatórios de testes;	
4. TÉCNICAS	
- > Revisões de software.	
5. PAPÉIS	
<small>PERFIL</small> Analista	

Fonte: Elaborado pelo autor.

3.2.6 Preparar a Validação

Selecionar os produtos, os componentes do produto a serem validados e quais os métodos de validação a serem utilizados. Preparar os requisitos de teste, casos de teste e especificações de teste selecionado para análise dos resultados dos testes no ambiente de validação.

No Quadro 5 é apresentado os objetivos, cada passo da atividade e suas respectivas entradas e saídas.

Quadro 5: Processo de preparar a validação.

ATV-005 Preparar a Validação
DESCRIÇÃO
<p>1. OBJETIVO</p> <p>O objetivo principal desta atividade é selecionar produtos, componentes a serem validados e métodos de validação a serem utilizados. Devem-se definir os casos de teste e especificações.</p>
<p>2. PASSOS</p> <ol style="list-style-type: none"> 1. Neste passo será elaborado o plano de validação para o projeto e elaborado os casos de testes; 2. Selecionar produtos para a validação; <ol style="list-style-type: none"> 2.1. Os produtos devem ser selecionados com base as suas contribuições para o alcance dos objetivos, considerando também os riscos do projeto. 3. Selecionar técnicas e ferramentas; <ol style="list-style-type: none"> 3.1. Selecionar métodos e técnicas de validação que estão disponíveis para uso e que serão aplicados a cada projeto; 3.2. Selecionar as ferramentas que serão usadas para apoiar a validação. 4. Definir procedimentos e critérios de validação; <ol style="list-style-type: none"> 4.1. Estabelecer e manter procedimentos e critérios para validação; 4.2. Os critérios são para definidos para garantir que os produtos estão de acordo com os requisitos. 5. Identificar e analisar as responsabilidades do sistema sendo testado; 6. Planejar testes do software; <ol style="list-style-type: none"> 6.1. Desenvolver e documentar para cada requisito, um conjunto de testes; 6.2. Projetar os casos de teste (entradas, saídas e critérios de teste) e procedimentos de teste para conduzir o teste do sistema; 6.3. Definir resultados esperados para cada caso de teste. 7. O sistema deve estar de acordo com o que o usuário deseja; 8. Com base em toda a documentação levantada elaborar o plano de validação.
<p>3.1 ARTEFATOS(S) DE ENTRADA</p> <ol style="list-style-type: none"> 1. Especificação de Requisitos.
<p>3.2 ARTEFATOS(S) DE SAÍDA</p> <ol style="list-style-type: none"> 1. Plano de validação; 2. Casos de testes.
<p>4. TÉCNICAS</p> <p>- > Testes estruturais.</p>
<p>5. PAPÉIS</p> <p>PERFIL</p> <p>Arquiteto e analista</p>

Fonte: Elaborado pelo autor.

3.2.7 Realizar a Validação

Para garantir que os produtos estão prontos, a atividade de validação é executada para garantir que todos os problemas detectados na validação foram identificados e documentados, deve-se também definir quais serão tratados.

No Quadro 6 é apresentado os objetivos, cada passo da atividade e suas respectivas entradas e saídas.

Quadro 6: Processo de realizar a validação.

ATV-006 Realizar a Validação	
DESCRIÇÃO	
1. OBJETIVO	
Garantir que todos os problemas detectados na validação foram identificados e documentados, deve-se também definir quais serão tratados.	
2. ATIVIDADES	
<ol style="list-style-type: none">1. Nesta atividade deve-se garantir que todos os problemas detectados na validação foram identificados e documentados, conforme atividades anteriores;2. Definir procedimentos para encaminhar relatórios da validação às partes envolvidas;<ol style="list-style-type: none">2.1. Devem ser definidos procedimentos para enviar os laudos de validação e relatório de testes gerados pelas atividades;3. Verificar os casos de testes;<ol style="list-style-type: none">3.1. Avaliar os casos de testes considerando os seguintes critérios: rastreabilidade e consistência com os requisitos, cobertura de testes dos requisitos, adequação dos métodos e padrões de teste utilizados, conformidade com os resultados esperados e viabilidade do teste;3.2. Documentar os casos seguindo padrão Relatório de revisão.4. Realizar testes;<ol style="list-style-type: none">4.1. Conduzir e documentar o teste de qualificação. Deve ser garantido que a implementação de cada requisito seja testada para conformidade, um teste bem sucedido mostra que o sistema opera conforme especificado;4.2. Os resultados do teste devem ser documentados, seguindo padrão Relatório de testes.	
3. ARTEFATOS	
3.1 ARTEFATO(S) DE ENTRADA	
<ol style="list-style-type: none">1. Plano de validação;2. Casos de testes.	
3.2 ARTEFATO(S) DE SAÍDA	
<ol style="list-style-type: none">1. Relatório de testes;2. Relatório de revisão.	
4. TÉCNICAS	

- > Teste de aceitação.

5. PAPÉIS

PERFIL

Testador e pessoas envolvidas com o projetos.

Fonte: Elaborado pelo autor.

3.2.8 Análise dos Resultados da Validação

Analisar todos os resultados das atividades de validação e registrar ocorrências e não conformidades identificadas durante o processo de validação.

No Quadro 7 é apresentado os objetivos, cada passo da atividade e suas respectivas entradas e saídas.

Quadro 7: Processo de análise dos resultados da validação.

ATV-007 Análise dos Resultados da Validação	
DESCRIÇÃO	
1. OBJETIVO	
Nesta atividade é analisado o resultado de todas as atividades de validação e identificar as ocorrências e não conformidades.	
2. ATIVIDADES	
<ol style="list-style-type: none">1. Nesta atividade será analisado o resultado de todas as atividades de validação e identificas as ocorrências e não conformidades;2. Verificar os casos de testes e o sistema integrado;<ol style="list-style-type: none">2.1. Avaliar os casos de testes e o sistema integrado segundo os seguintes critérios: cobertura dos testes dos requisitos, adequação dos métodos e padrões de teste utilizados, conformidade com os resultados esperados, viabilidade da operação e manutenção;2.2. O desenvolvedor deve garantir que o sistema integrado está pronto para o teste do sistema;2.3. Registrar não conformidades identificadas e definir as linhas do plano de ação para corrigi-las, conforme Relatórios de ocorrências e registro de não conformidades.3. Realizar testes para homologação interna;<ol style="list-style-type: none">3.1. Realizar o teste do sistema de forma a garantir que a especificação de cada requisito seja testada, conformidade com os requisitos, e que o sistema está pronto para ser entregue;3.2. Os resultados do teste devem ser documentados, conforme Relatório de análise dos resultados.4. Verificar o sistema.<ol style="list-style-type: none">4.1. Avaliar o sistema segundo os seguintes critérios: cobertura de teste dos requisitos do sistema, conformidade com os resultados esperados, viabilidade	

de operação e manutenção; 4.2. Registrar não conformidades identificadas e definir as linhas do plano de ação para corrigi-las, conforme Relatórios de ocorrências e registro de não conformidades.
3. ARTEFATOS
3.1 ARTEFATO(S) DE ENTRADA
1. Relatórios de ocorrências e registro de não conformidades; 2. Relatório de testes; 3. Relatório de revisão.
3.2 ARTEFATO(S) DE SAÍDA
1. Relatórios de ocorrências e registro de não conformidades; 2. Relatório de análise dos resultados.
4. TÉCNICAS
- > Brainstorming.
5. PAPÉIS
PERFIL Analista e testador

Fonte: Elaborado pelo autor.

3.2.9 Documentação e Comunicação dos Resultados

O registro da comunicação entre as várias funções e níveis da organização garante agilidade e confiabilidade na verificação e validação. O processo de comunicação inclui o estabelecimento de planos das atividades de V&V.

A documentação em uma empresa é fundamental para garantir o controle de implantação do Sistema de Gestão. Além de facilitar uma avaliação permanente e possíveis melhorias, reforça a conscientização dos colaboradores das responsabilidades no cumprimento dos objetivos e das metas estabelecidas.

No Quadro 8 é apresentado os objetivos, cada passo da atividade e suas respectivas entradas e saídas.

Quadro 8: Processo de documentação e comunicação dos resultados.

ATV-008 Documentação e comunicação dos resultados
DESCRIÇÃO
1. OBJETIVO
Essa atividade é para documentação e comunicação dos resultados obtido no processo de V&V.

2. ATIVIDADES

1. Elaboração da documentação completa;
 - 1.2. Nesta etapa devem ser recuperados todos os artefatos gerados, bem como os resultados obtidos;
2. Montagem do relatório final.
 - 2.1. Anexar todos os artefatos elaborados ao relatório final, seguindo padrão Relatório final.

3. ARTEFATOS

3.1 ARTEFATO(S) DE ENTRADA

1. Plano de verificação;
2. Plano de validação;
3. Casos de testes;
4. Especificação de Requisitos;
5. Relatórios de ocorrências e registro de não conformidades;
6. Relatório de testes;
7. Relatório de análise dos resultados.

3.2 ARTEFATO(S) DE SAÍDA

1. Relatório final.

4. TÉCNICAS

- > Brainstorming.

5. PAPÉIS

PERFIL

Todos os envolvidos no projeto

Fonte: Elaborado pelo autor.

3.2.10 Avaliação do processo

Neste item tratamos sobre a avaliação do processo de V&V desenvolvido neste trabalho. Inicialmente o objetivo de avaliação, consistia em aplicar o processo em uma organização de desenvolvimento de software e avaliá-lo em um projeto real da organização. Entretanto estes objetivos não puderam ser atingidos pelo fato do trabalho ser finalizado em um período onde não havia projeto disponível para a aplicação do mesmo. No entanto, realizei uma avaliação através de questionários com três especialistas da área de testes, homologações e validações da empresa. Estes avaliadores têm em média quatro anos de experiência na área e formação acadêmica de nível superior em TI. As perguntas abaixo do questionário consistem em: avaliar a aplicabilidade e qualidade do processo de V&V em relação ao seu conteúdo:

- Os artefatos (entrada/saída) estão completos e os campos possuem informações?

- A descrição das atividades está completa e possui uma boa organização e apresentação?
- A descrição das atividades é, suficientemente, compreensível e clara de tal forma que ao lê-la o usuário não tenha dúvidas de entendimento?
- Sua característica é ser simples e possível de ser aplicado. O padrão de documentação foca nos itens essenciais do processo de testes que devem ser documentados, a fim de aumentar a percepção da qualidade pelo usuário final do software?
- O conteúdo apresentado pertence ao escopo de Verificação e Validação?
- Os documentos possuem organização clara e objetiva?
- As técnicas apresentadas aumentam a qualidade do produto?
- Pode ser aplicado em um processo de software já existente, ou utilizado no desenvolvimento de um processo próprio?

3.2.10.1 Análise dos Resultados

A análise das respostas obtidas a partir dos questionários aplicados aos três avaliadores demonstra uma aceitação positiva, caso fossem aplicados mais questionários sobre o processo de V&V com mais especialistas seriam obtidas novas opiniões e contribuições para melhoria do mesmo. Abaixo alguns comentários dos três avaliadores:

Em relação às atividades do processo, um problema é a adoção em uma empresa que não segue nenhum procedimento de desenvolvimento. Conforme experiência, depois que todos estão “viciados” em um processo não definido, mudar requer paciência e perseverança de todos os envolvidos. Com relação aos casos de testes, deveriam possuir mais informações, o estado da execução, quem executou, a versão do caso de teste e qual a versão do produto.

Alguns pontos positivos foram destacados: Um deles foi que os planos de verificação e validação ficaram bons mas, que seria interessante fazer a aplicação dos mesmos, analisando os resultados na prática, enriquecendo o trabalho. Outro ponto foi que as atividades avaliadas do processo de V&V foram baseadas em modelos conhecidos na área de engenharia de software.

4 CONCLUSÃO

Este trabalho apresentou conceitos de V&V em projetos de software, com base nos modelos e normas de referência CMMI e MPS.BR. Muitas lições foram aprendidas ao longo da elaboração deste trabalho. Entre elas, destaca-se a importância da utilização de um processo de desenvolvimento e da V&V durante todo o ciclo de vida de um projeto.

O conjunto de atividades voltadas para a V&V colabora para que o gerente de projeto tenha uma visão do projeto em execução em diversos aspectos, tais como: garantia da qualidade, progresso, produtividade, acompanhamento de esforço e variação de tamanho do software. A utilização de informações de projetos realizados e dos projetos já em andamento facilitará a atuação e a decisão da gerência para a conclusão do projeto em questão.

Referente aos objetivos: Aplicar e avaliar o processo de V&V na modelagem de um processo real em uma organização de desenvolvimento de software, estes não foram atingidos pelo fato do trabalho ser finalizado em um período onde não havia projeto disponível para a aplicação do mesmo, bem como, a previsão de término de novos projetos se estenderem mais tempo que o prazo para conclusão deste trabalho conforme o planejado.

Como não foi possível à avaliação de aplicação real do processo, foi realizada uma avaliação com três especialistas da área de qualidade de software da empresa. O resultado desta análise mostrou uma aceitação regular à boa, conforme relatado anteriormente.

Com o objetivo geral alcançado, Desenvolvimento de um Processo de Verificação e Validação para Produtos de Software, o trabalho possibilitou aprofundar conhecimentos de TI, normas e modelos, engenharia de software, muito úteis para o mercado de trabalho e evolução profissional.

REFERÊNCIAS

BARTIÉ, Alexandre. **Garantia da Qualidade de Software: as melhores práticas de engenharia de software aplicadas à sua empresa**. 5. Edição. Rio de Janeiro: Campos, 2002.

BARTIÉ, Alexandre. **Fábrica de Testes – Parte 01**. iMasters, 2006.a. Disponível em: <www.imasters.com.br>. Acesso em: 01 de maio de 2013.

BASTOS, A., RIOS, E., CRISTALLI, R., MOREIRA, T. **Base de conhecimento em teste de software**. – 2ª Ed. rev. – São Paulo: Martins, 2007.

BSTQB, **Certificação em Teste de Software**, 2007 Disponível em: <<http://www.bstqb.org.br>> Acesso em: 02 de maio de 2013.

CORSO, Ariane L. **Guia para Redução do Custo Total de defeitos pela aplicação de Modelos de Processo de Teste de Software**, 2008. Disponível em: <<http://www.unimep.br/phpg/mostracademica/anais/4mostra/pdfs/421.pdf> >, Acesso em 28 de abril de 2013.

CRESPO, A. N.; SILVA, O.J.; BORGES, C.A.; SALVIANO, C.F.; TEIVE, M.; JINO, M. Artigo: **Uma metodologia para teste de software no contexto da melhoria de processo**. São Paulo, 2004.

DIAS NETO, Arilo Cláudio. **Qualidade de Software – Entenda os principais conceitos sobre Testes e Inspeção de Software – Artigo: Introdução a Teste de Software**. Engenharia de Software Magazine, Equipe DevMedia. Ano 1 – 1ª Edição, 2007.

GROSSO, Carlos. **Verificação e Validação no CMMI**, 2006. Disponível em: <<http://www.sinfic.pt>>, Acesso em 25 de abril de 2013.

HAUCK, Jean Carlo Rossa; WANGENHEIM, Christiane Gresse von; THIRY, Marcello. **Suportando a Modelagem de Processo de Monitoração e Controle em**

Micro e Pequenas Empresas, alinhado ao CMMI, MPS.BR e ISO/IEC15504. SBQS - Simpósio Brasileiro de Qualidade de Software, 2007.

HAUCK, Jean Carlo Rossa; WANGENHEIM, Christiane Gresse Von; SOUZA, Richard H.; THIRY, Marcello. Livro: **Process Reference Guides Support for Improving Software Processes in Alignment with Reference Models and Standards**, 2008.

IEEE. **IEEE Standard for Software Test Documentation.** IEEE-SA Standards Board 1999.

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

ISO, **NBR ISO/IEC 12207 - Tecnologia de informação – Processos de ciclo de vida de software**, 1998.

MOLINARI, Leonardo. **Testes de Software: produzindo sistemas melhores e mais confiáveis.** 3. ed. São Paulo: Érica, 2006.

PMI Project Management Institute. **Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos (Guia PMBOK®).** 3 ed. Project Management Institute, Four Campus Boulevard, Newtown Square, Pennsylvania, USA, 2004. Tradução oficial de: “A Guide to the Project Management Body of Knowledge” (PMBOK® Guide).

PRESSMAN, Roger S. **Engenharia de Software.** Tradução de Rosângela Delloso Penteadó. Revisão técnica Fernão Stella R. Germano, José Maldonato, Paulo César Masiero. 6. ed. São Paulo: McGraw-Hill, 2006.

ROCHA, Ana Regina Cavalcanti da; MALDONADO, José Carlos; WEBER, Kival Chaves. **Qualidade de software.** São Paulo: Prentice Hall, 2001.

SEI – Software Engineering Institute. **CMMI®, Version 1.2.** Disponível em: <<http://www.sei.cmu.edu>>. Acesso em: 29 de abril de 2013. 2012.

SOFTEX. MPS.BR – Melhoria de Processo do Software Brasileiro – Guia de implementação – Parte 4: Nível D (Versão 1.0). 2006.

SOFTEX. MPS.BR – Melhoria de Processo do Software Brasileiro – Guia geral.

Disponível em:<

http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_Geral_2009.pdf >. Acesso em: 30 de abril de 2013.

SOMMERVILLE, Ian. Engenharia de Software. Tradução de André Maurício de Andrade Ribeiro. Revisão técnica de Kechi Hiramã. 6. ed. São Paulo: Addison Wesley, 2003.