

ENGENHARIA DE SOFTWARE II

CAPÍTULO 2 – QUALIDADE E PRODUTIVIDADE DE SOFTWARE, r2

Neste capítulo serão abordados os seguintes itens:

- 2.1 Qualidade: Uma abordagem da engenharia de software
- 2.2 Fatores de acompanhamento da qualidade de software
 - 2.2.1 Processos de controle da qualidade do software
 - 2.2.2 GQS – Garantia de qualidade de *software* (SQA – *Software Quality Assurance*)
 - 2.2.3 Custo da qualidade
 - 2.2.4 Métricas da qualidade do *software*
 - 2.2.5 Principais métricas de qualidade do *software*
- 2.3 Qualidade do produto da engenharia de software: ISO/IEC 9126 e ISO 25000
 - 2.3.1 Objetivo
 - 2.3.2 Estrutura do modelo de qualidade
 - 2.3.3 Modelo de qualidade para qualidade externa e interna
 - 2.3.4 Modelo de qualidade para qualidade em uso
 - 2.3.5 Norma *Software* ISO/IEC 25000 - *Software Product Quality Requirements and Evaluation*
- 2.4 Sistemas da qualidade: ISO 9001 e ISO 9003
 - 2.4.1 A serie ISO 9000
 - 2.4.2 Descrição da serie ISO 9000 e outros relacionados
 - 2.4.3 ISO 9001
 - 2.4.4 ISO 9003
- 2.5 ISO 12207 - Processos do Ciclo de Vida do Software
- 2.6 Modelo de qualidade de *software*: CMMI (*Capability Maturity Model Integration*)
 - 2.6.1 Os cinco níveis de maturidade
 - 2.6.2 ACP – Áreas Chave de Processo (KPA – *Key Process Area*)
- 2.7 Modelo de qualidade de *software*: SPICE – ISO 15504
 - 2.7.1 Elementos normativos da ISO 15504
 - 2.7.2 Dimensão dos processos da ISO 15504
- 2.8 Modelo de qualidade de *software*: MPS.BR
 - 2.8.1 Modelo de maturidade da MPS-BR

2.1 QUALIDADE: UMA ABORDAGEM DA ENGENHARIA DE SOFTWARE

A abordagem da Engenharia de *Software* trabalha buscando uma única meta: produzir *software* de alta qualidade. A qualidade é importante, mas se o usuário não está satisfeito, nada mais realmente importa.



Satisfação do usuário = produto adequado
+ máxima qualidade
+ entrega dentro do orçamento e do cronograma

Principais objetivos da qualidade:

- Reunir um conjunto de características, regras e procedimentos de modo que o software satisfaça às necessidades de seus usuários;
- Levar menos tempo para fazer uma coisa correta, do que explicar o porque foi feito errado;
- Garantir a qualidade mesmo depois de estar pronto o produto;
- Estabelecer cronogramas e custos reais.

2.2 FATORES DE ACOMPANHAMENTO DA QUALIDADE DE SOFTWARE

Os principais fatores da qualidade de software são:

- Controle da qualidade;
- Garantia da qualidade; e
- Custo da qualidade.

2.2.1 Processo de Controle da Qualidade do software



Processo de controle da qualidade do software

2.2.2 GQS – Garantia de Qualidade de Software (SQA – Software Quality Assurance)

O propósito da Garantia de Qualidade de *Software* é fornecer à gerência a visibilidade da eficácia (aquilo que dá bom resultado):

- Examinar minuciosamente um artefato de *software* ou estado do projeto com a finalidade de determinar se há algum desvio com relação aos padrões, diretrizes, especificações, procedimentos e planos aprovados;
- As revisões devem ser aplicadas em vários pontos de função durante o desenvolvimento e servem para descobrir defeitos enquanto estes ainda são relativamente baratos para serem tratados.

2.2.3 Custo da Qualidade

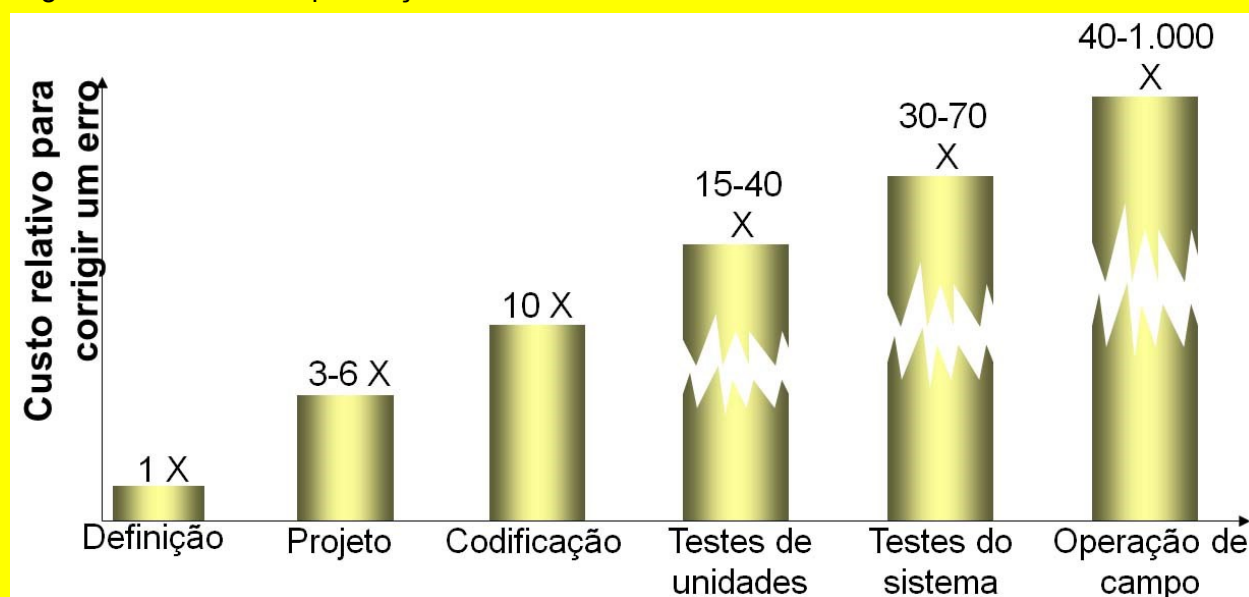
O custo da qualidade pode ser entendido como o preço justo, que o cliente esteja disposto a pagar. Estes custos podem ser divididos em:

- **Custos de prevenção** - Planejamento; Revisões; Ferramentas de diagnósticos e testes; e Treinamento.
- **Custos de avaliação** - Inspeção intra e interprocessos; Precisão e manutenção dos equipamentos.
- **Custos de falha interna** - Refazer; Reparar; e Análise do modo como a falha ocorreu.
- **Custos de falha externa** - Solução das queixas; Devolução e substituição do produto; Manutenção da linha de suporte; e Trabalho de garantia.

Estudo de Caso: “Os dados relatados por Kaplan, Clark e Tang em 1995, são baseados em trabalhos da instalação de desenvolvimento de software da IBM”

- ✚ 7.053 hs. foram gastas para inspecionar 200.000 linhas de código, resultando 3.112 defeitos em potencial. Se considerar o custo de programador equivalente a US\$ 40,00 por hora, temos que o custo total para prevenir defeitos é igual a US\$ 282.120,00, ou seja, aproximadamente US\$ 91,00 por defeito;
- ✚ **Sem inspeções**, mas com programadores super cuidadosos, foram registrados 1 defeito por 1.000 linhas de código. O que significa que 200 defeitos foram corrigidos em campo a um custo de US\$ 25.000,00 por defeito. O que reflete um gasto de US\$ 5.000.000,00, **18 vezes mais caro do que o custo da prevenção.**

Os dados relatados por Kaplan, Clark e Tang em 1995 é a confirmação do gráfico apresentado pela pesquisa de Boehm em 1981, que mostram como o custo da falha aumenta se migrarmos do custo da prevenção até os custos de falhas externas.



2.2.4 Fatores e Métricas da qualidade do software

(Pressman, 2002) Qualidade de *software* é uma mistura complexa de fatores que vão variar com cada aplicação diferente e com os clientes que as encomendam. Os **fatores de qualidade do software** caracterizam um **conjunto de métricas da qualidade** (forma de medição) e que podem ser divididos em dois grupos:

1. Fatores que podem ser medidos diretamente. Exemplo: defeitos por ponto de função;
2. Fatores que podem ser medidos indiretamente. Exemplo: usabilidade ou manutenibilidade (ou manutenibilidade).

As **métricas de qualidade do software** são medidas quantitativas que permite ao pessoal do *software* ter idéia da eficácia do processo de *software* e dos produtos que são conduzidos usando o processo como arcabouço, também são usadas para detectar áreas de problemas de maneira que soluções possam ser apresentadas, e que o processo de *software* possa ser melhorado.

São coletados dados básicos de produtividade e qualidade pelos engenheiros de *software*, esses dados são analisados, comparados com médias anteriores e avaliados para verificar se ocorreu melhoria ou não pelos gerentes de *software*.

Exemplo de análise: Fatores X Métricas da qualidade do software. Fonte: Pressman, 2007.

FATORES DE QUALIDADE \ MÉTRICA DE QUALIDADE	Confiabilidade	Eficiência	Mantenabilidade	Utilização
Auto Documentação			X	
Complexidade	X			
Concisão		X	X	
Consistência	X		X	
Eficiência de Execução		X		X
Instrumentação			X	X
Modularidade	X		X	
Operabilidade		X		X
Simplicidade	X		X	
Tolerância a erro	X			

2.2.5 Principais métricas de qualidade do *software* [Fontes: Pressman, 2007; NBR 9126; NBR 12207]

MÉTRICA	DEFINIÇÃO
Acurácia	Capacidade do produto de software de prover, com o grau de precisão necessário, resultados ou efeitos corretos ou conforme acordados.
Adaptabilidade	Capacidade do produto de software de ser adaptado para diferentes ambientes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado.
Adequação	Capacidade do produto de <i>software</i> de prover um conjunto apropriado de funções para tarefas e objetivos do usuário especificados.
Analísabilidade	Capacidade do produto de software de permitir o diagnóstico de deficiências ou causas de falhas no software, ou a identificação de partes a serem modificadas.
Apreensibilidade	Capacidade do produto de software de possibilitar ao usuário aprender sua aplicação.
Atratividade	Capacidade do produto de software de ser atraente ao usuário.
Auditabilidade	Facilidade no atendimento as normas que pode ser verificado.
Auto documentação	Nível em que o código fonte fornece documentação significativa.
Capacidade	Capacidade do produto de <i>software</i> em ser instalado em ambiente especificado.e atender aos requisitos de funcionamento.
Coexistência	Capacidade do produto de <i>software</i> de coexistir com outros produtos de <i>software</i> independentes, em um ambiente comum, compartilhando recursos comuns.
Confiabilidade	A efetiva realização das funções de um programa um nível de desempenho e precisão especificadas.
Conformidade	Capacidade do produto de <i>software</i> de estar de acordo com normas, convenções ou regulamentações previstas em leis e prescrições similares relacionadas ao atributo do <i>software</i> .
Estabilidade	Capacidade do produto de software de evitar efeitos inesperados decorrentes de modificações no software.
Eficácia	Capacidade do produto de software de permitir que usuários atinjam metas especificadas com acurácia e completitude, em um contexto de uso especificado.
Eficiência	Quantidade de recursos e códigos de computação para realização da função. Quanto menor a quantidade, maior será o desempenho do software.

Expansibilidade	Nível em que o projeto arquitetural de dados ou procedimental pode ser estendido.
Flexibilidade	Esforço necessário para modificar um programa operacional.
Funcionalidade	Capacidade do produto de <i>software</i> de prover funções que atendam às necessidades explícitas e implícitas, quando o <i>software</i> estiver sendo utilizado sob condições especificadas.
Generalidade	Âmbito da aplicação potencial dos componentes do programa.
Integridade	Diz respeito ao acesso do software ou dados por pessoas não autorizadas e que pode ser controlado.
Modificabilidade	Capacidade do produto de software de permitir que uma modificação especificada seja implementada.
Interoperabilidade	Capacidade do produto de <i>software</i> de interagir com um ou mais sistemas especificados.
Integridade	Controle do acesso aos dados de pessoas não autorizadas.
Mantenabilidade	Esforço necessário para localizar e corrigir erros num programa.
Manutenabilidade	Capacidade do produto de software de ser modificado. As modificações podem incluir correções, melhorias ou adaptações do software devido a mudanças no ambiente e nos seus requisitos ou especificações funcionais.
Maturidade	Capacidade do produto de software de evitar falhas decorrentes de defeitos no <i>software</i> .
Modularidade	Diz respeito a independência funcional dos componentes do programa.
Precisão	Nível de precisão dos cálculos e do controle que pode ser verificado.
Operacionalidade	Capacidade do produto de <i>software</i> de possibilitar ao usuário operá-lo e controlá-lo.
Produtividade	Capacidade do produto de software de permitir que seus usuários empreguem quantidade apropriada de recursos em relação à eficácia obtida, em um contexto de uso especificado.
Segurança	Capacidade do produto de <i>software</i> em apresentar níveis aceitáveis de riscos de danos às pessoas, aos negócios, da propriedade, do ambiente, de proteger informações e dados, de forma que pessoas ou sistemas não autorizados não possam lê-los nem modificá-los e que não seja negado o acesso às pessoas ou sistemas autorizados.
Testabilidade	Necessário o teste de um programa para validação, quando criado ou modificado, a fim de garantir que realize a função esperada.
Portabilidade	Capacidade do produto de <i>software</i> ser transferido de um ambiente operacional para outro.
Rastreabilidade	Diz respeito a habilidade de rastrear uma representação de projeto ou componente real do programa.
Recuperabilidade	Capacidade do produto de software de restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha.
Reutilização	Quanto um programa ou parte dele pode ser reusado em outras aplicações.
Satisfação	Capacidade do produto de software de satisfazer usuários, em um contexto de uso especificado. <i>NOTA - Satisfação é a resposta do usuário à interação com o produto e inclui atitudes relacionadas ao uso do produto.</i>
Usabilidade	Capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições especificadas.
Utilização	Esforço necessário para aprender, operar, preparar entradas e interpretar saídas de um programa.

2.3 QUALIDADE DO PRODUTO DA ENGENHARIA DE SOFTWARE: ISO/IEC 9126 E ISO 25000



**ABNT - Associação
Brasileira de
Normas Técnicas**

Sede:
Rio de Janeiro
Av. Treze de Maio, 13/28º andar
CEP 20003-900 - Caixa Postal 1680
Rio de Janeiro - RJ
Tel.: PABX (21) 3974-2300
Fax: (21) 2240-8249/2220-6436
Endereço eletrônico:
www.abnt.org.br

Copyright © 2003,
ABNT—Associação Brasileira de
Normas Técnicas
Printed in Brazil/
Impresso no Brasil
Todos os direitos reservados

JUN 2003		NBR ISO/IEC 9126-1
Engenharia de software - Qualidade de produto		
Parte 1: Modelo de qualidade		
Origem: Projeto 21:101.01-009:2002 ABNT/CB-21 - Comitê Brasileiro de Computadores e Processamento de Dados CE-21:101.01 - Comissão de Estudo de Qualidade de Software ISO/IEC 9126-1 - Software engineering - Product quality - Part 1: Quality model Descriptors: Software. Quality. Information technology. Evaluation Esta Norma é equivalente à ISO/IEC 9126-1:2001 Esta Norma cancela e substitui a NBR 13596:1996 Válida a partir de 30.07.2003		
Palavras-chave: Qualidade. Software. Avaliação. Engenharia de software		21 páginas

2.3.1 Objetivo

Esta parte da NBR ISO/IEC 9126 descreve **um modelo de qualidade do produto de software, composto de duas partes:**

- qualidade interna e qualidade externa e
- qualidade em uso.

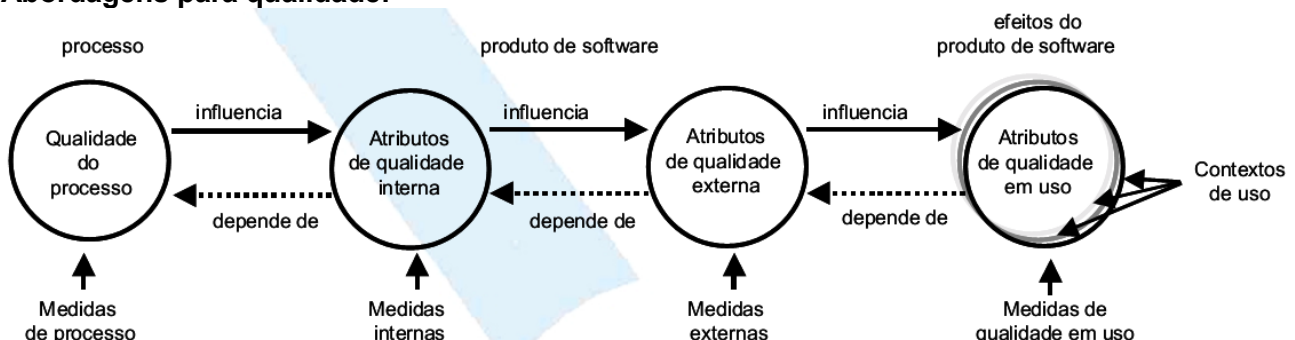
A **primeira parte do modelo especifica seis características** para qualidade interna e externa, as quais são por sua vez subdivididas em subcaracterísticas. Estas subcaracterísticas são manifestadas externamente, quando o *software* é utilizado como parte de um sistema computacional, e são resultantes de atributos internos do software.

Exemplos de usos da NBR ISO/IEC 9126:

- Validar a completitude de uma definição de requisitos;
- Identificar requisitos de software;
- Identificar objetivos de projeto de software;
- Identificar objetivos para teste de software;
- Identificar critérios para garantia de qualidade;
- Identificar critérios de aceitação para produtos finais de software.

2.3.2 Estrutura do modelo de qualidade

Abordagens para qualidade:



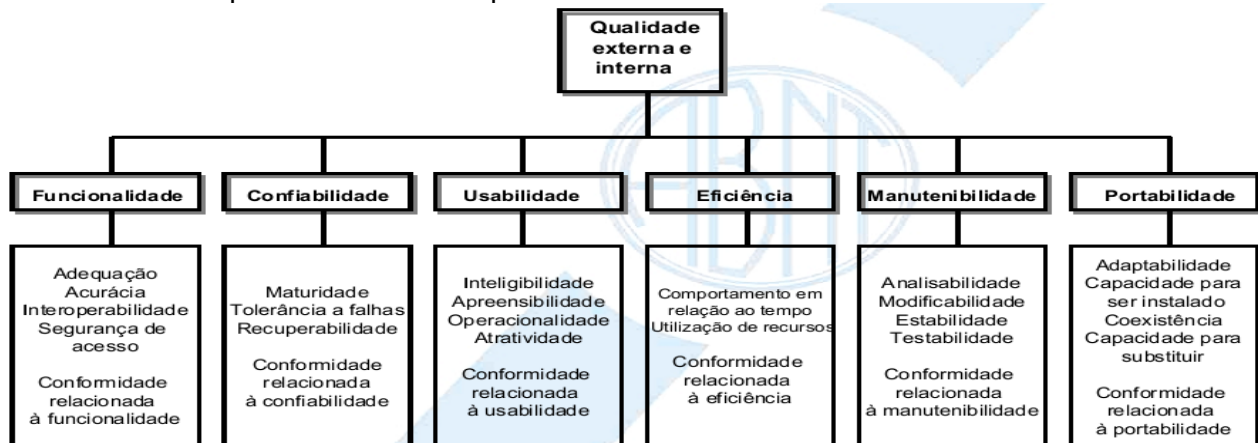
Qualidade no ciclo de vida. Fonte: NBR 9126-1.

2.3.3 Modelo de qualidade para qualidade externa e interna

O **modelo de qualidade externa e interna** categoriza os **atributos de qualidade de software** em **seis características**:

- Funcionalidade,
- Confiabilidade,
- Usabilidade,
- Eficiência,
- Manutenibilidade e
- Portabilidade

Das quais são, por sua vez, subdivididas em **subcaracterísticas** (figura abaixo). As subcaracterísticas podem ser medidas por meio de métricas externas e internas

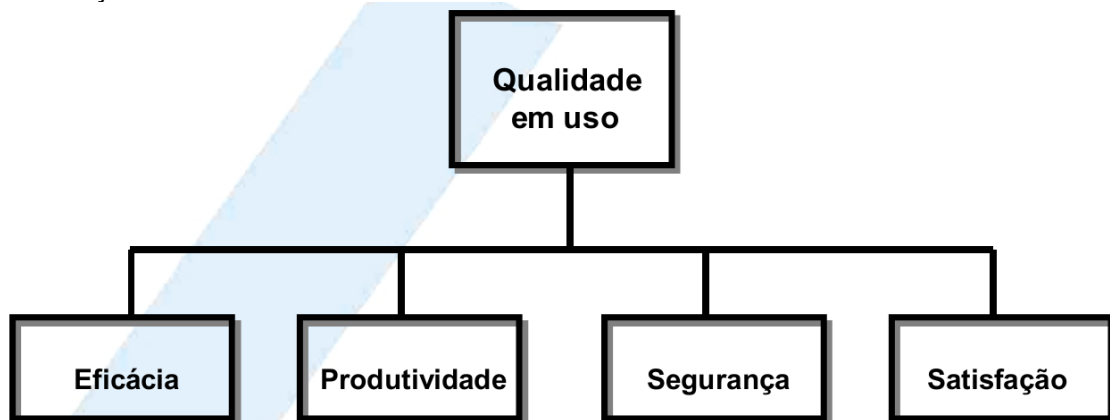


Modelo de qualidade para qualidade externa e interna. Fonte: NBR 9126-1.

2.3.4 Modelo de qualidade para qualidade em uso

Os **atributos de qualidade em uso** são categorizados em **quatro características** (ver figura abaixo):

- Eficácia,
- Produtividade,
- Segurança e
- Satisfação.



Modelo de qualidade para qualidade em uso. Fonte: NBR 9126-1.

2.3.5 Norma Square ISO/IEC 25000 - Software Product Quality Requirements and Evaluation

Esta norma que diz respeito de caracterização e **medição de qualidade de produto de software**. É uma evolução das séries de normas ISO/IEC 9126 e ISO/IEC 14598. *Square* significa *Software Product Quality Requirements and Evaluation* (Requisitos de Qualidade e Avaliação de Produtos de Software).

A reorganização da Norma ISO/IEC 25000 gera uma nova divisão de assuntos em cinco tópicos, sendo que cada divisão é composta por um conjunto de documentos e trata de um assunto em particular:

Requisitos de Qualidade	Modelo de Qualidade	Avaliação
	Gerenciamento de qualidade	
	Medições	

2.4 SISTEMAS DA QUALIDADE: ISO 9001 E ISO 90003

2.4.1 A Serie ISO 9000

A **série ISO 9000**, é uma parte dos diversos padrões emitidos pela ISO, a qual é definida como **Padrões para a Gerência da Qualidade e Garantia de Qualidade**.

Esta serie descreve e explica os requisitos de um sistema de qualidade. No entanto, ISO-9000 não prescreve algum modelo específico para um Sistema de Qualidade. **Para obtê-lo as organizações necessitam desenhar seus próprios sistemas de qualidade. Seu maior objetivo é certificar organizações** no que concerne a Padrões de Qualidade, em uma determinada área da indústria.

Premissa da serie ISO 9000: "se a produção e a administração do sistema de qualidade são corretas, o produto ou serviço produzido também será correto".

2.4.2 Descrição da serie ISO 9000 e outros relacionados

A série ISO-9000, provê diretrizes e modelos para orientar a implementação de um sistema de Qualidade. Seu primeiro conjunto de normas é a ISO 9000 que esta dividida em quatro partes, nas quais se esclarecem as diferencias entre os principais conceitos de qualidade, e provê diretrizes para a seleção e uso:

- **ISO 9000:** Gestão e Garantia da Qualidade e Diretrizes para a seleção e uso das partes:
 - ISO 9000-1: Seleção e Uso
 - ISO 9000-2: Diretrizes genéricas para a aplicação da ISO 9001, ISO 9002 e ISO 9003
 - ISO 9000-3: Diretrizes para a aplicação da ISO-9001 em desenvolvimento, fornecimento e manutenção de software.
 - ISO 9000-4: Aplicações da gestão de dependência.

Enquanto que a ISO 9000 determina os padrões para a gerencia e garantia da qualidade. Demais outras normas especificam a área de aplicação, como é mostrado abaixo:

- **ISO 9001:** Modelo de Garantia de Qualidade em projeto, instalação, desenvolvimento, produção, arquitetura e serviço.
- **ISO 9002:** Modelo de Qualidade em produção, ensaios e instalação.
- **ISO 9003:** Modelo de Garantia de Qualidade em inspeção e ensaios finais (testes).
- **ISO 9004:** Gestão da Qualidade e elementos do sistema da qualidade

As normas **ISO 9001 a ISO 9003** especificam **os requisitos do Sistema de Qualidade para o desenvolvimento do software**. Estas normas devem ser usadas quando existe um contrato entre duas partes, o qual requeira a demonstração da capacidade de um fornecedor em desenvolver e fornecer um produto, software em nosso caso, com uma qualidade garantida.

2.4.3 ISO 9001

A **ISO 9001** é usada para demonstrar capacidade de atender aos **requisitos do cliente**, os regulamentos da própria organização. Define um conjunto de requisitos para o **SGQ – Sistema de Gestão da Qualidade** com base nas seguintes **cláusulas**:

- Responsabilidade Gerencial
- Sistema de Qualidade
- Análise Crítica de Contratos
- Controle de Projetos
- Controle de Dados e Documentos
- Controle de Produtos fornecidos pelo Cliente
- Identificação de Produtos e rastreabilidade
- Controle do Processo
- Teste e Inspeção
- Controle de equipamentos de testes, medições e inspeções
- Situação de testes e inspeções
- Controle de produto sem conformidade
- Ação Corretiva e Preventiva
- Manuseio, armazenamento, empacotamento, proteção e entrega
- Controle de Registros de Qualidade
- Auditorias Internas de Qualidade
- Treinamento
- Assistência Técnica
- Técnicas Estatísticas

2.4.4 ISO 90003

A **ISO 90003** é uma evolução da norma **ISO 9000-3:2000**, que descreve as Diretrizes para a aplicação da ISO 9001. Na **ISO 90003:2004** os principais pontos são:

Planejamento da realização do Produto:

- **Ciclo de Vida** - deve conter as etapas de Teste, Verificação, Acompanhamento (análise crítica) e Validação.
- **Plano de Qualidade** – Para um projeto específico, Requisitos específicos e Definição das alterações em vigor do SGQ.
- **Requisitos da qualidade** (produto e processo).

Processos relacionados ao Cliente:

- **Determinação dos Requisitos do Produto** – Elicitação dos requisitos, Aprovação pelo cliente, Requisitos implícitos, requisitos legais, normativos ou do próprio fornecedor.
- **Análise crítica dos requisitos** – Edital, Proposta, Contrato, Capacidade para atender, Orientações específicas do *software*, padrões, /plataformas/ambientes, ferramentas, ciclo de vida e risco.
- **Comunicação com o cliente** – suporte, manutenção e *helpdesk*.

Projeto de Desenvolvimento:

- **Planejamento** – WBS – *Work Breakdown Structure* (Estrutura Analítica do Projeto), Cronograma, Etapas, Responsabilidades, Atividades de controle, Interfaces, Recursos e Ambiente.
- **Entradas** – elementos necessários para compor a WBS.
- **Saídas** – saídas das Fases, Produto final, Código fonte e Objeto, Manuais, Documentação e Relatórios de testes.
- **Análise crítica de projeto** – *Design review* e tomada de decisão.
- **Verificação** – simulações, walkthrough, inspeções, demonstrações com protótipo.
- **Validação** – atendimento aos requisitos no ambiente de uso pretendido?; teste: unitário, integração, sistema, regressão e aceitação.
- **Alteração de projeto** – havendo alterações, deve retornar ao planejamento.

Aquisição:

- Aquisição de **Insumos** (ferramentas e *hardware*).
- **Subcontratações** ou aquisição de **Partes das funções do software**.
- **Inspeção e Recebimento** – Verificação, Validação, Aceitação e Testes.

Produção e Fornecimento de serviço:

- **Controle de produção** – Release, Entrega, Instalação, Operação e Manutenção.
- **Identificação e Rastreabilidade** – Gestão de configuração.
- **Propriedade do cliente** – Componentes, Ferramentas, Ambiente operacional e de Testes, e Propriedade intelectual.
- **Preservação do produto** – Armazenamento, Acesso, Proteção contra vírus ou Deterioração.

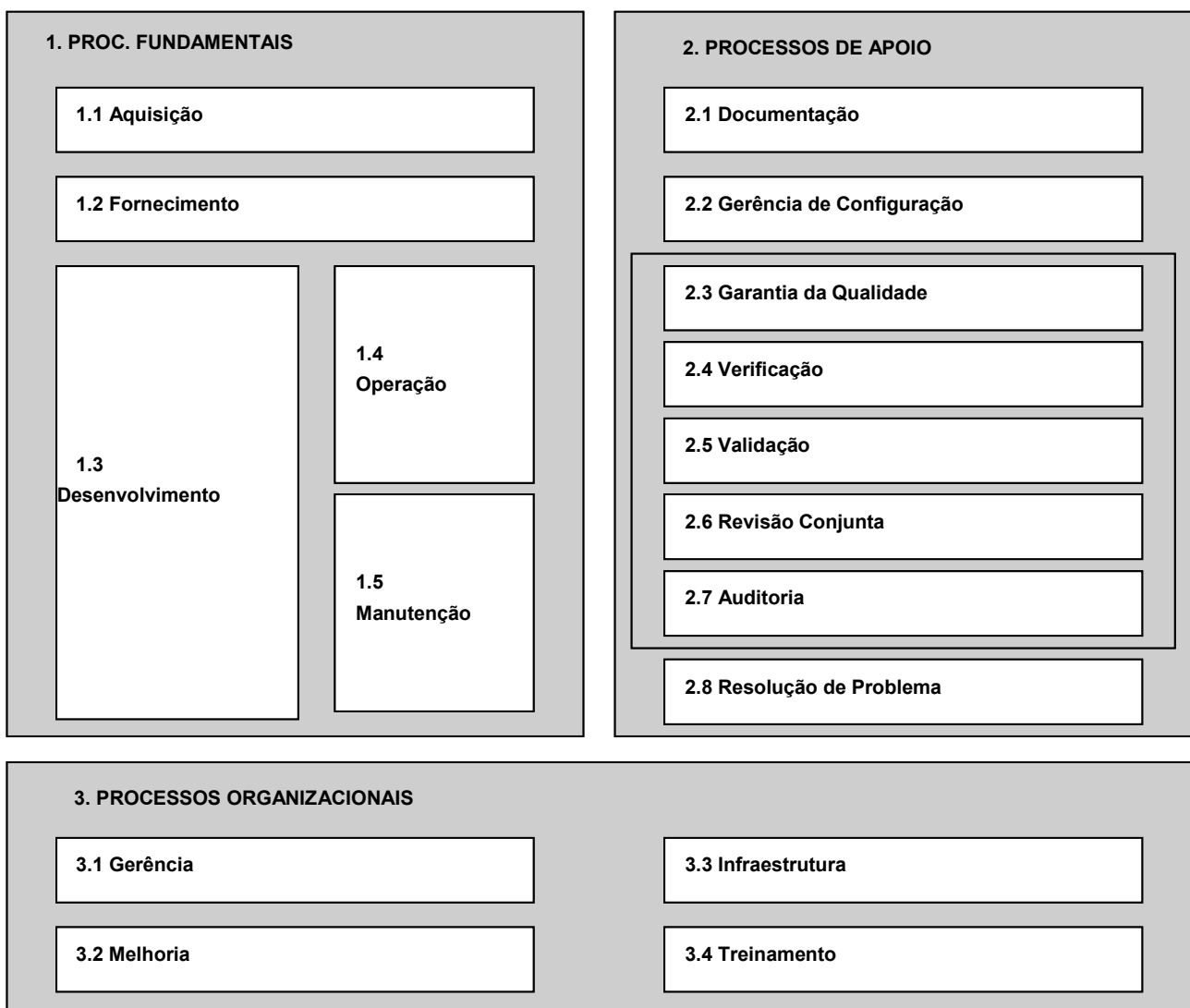
Medição, Análise e Melhoria.

2.5 ISO 12207 – PROCESSOS DO CICLO DE VIDA DO SOFTWARE

Quando o processo envolve a elaboração de um produto, nos referimos a ele **como ciclo de vida**. Assim o processo de desenvolvimento de software pode ser chamado de **ciclo de vida do software**, pois ele descreve a ‘vida’ do produto de software desde a concepção até a implementação, entrega, utilização e manutenção. [PFL04]

A Norma **NBR ISO/IEC 12207 - Processos do Ciclo de Vida do Software** foi criada em 1995 com o **objetivo de fornecer uma estrutura com base em uma linguagem comum** para que o adquirente, fornecedor, desenvolvedor, mantenedor, operador, gerentes e técnicos envolvidos com o desenvolvimento de software. Esta linguagem comum é estabelecida na forma de processos bem definidos.

Esses processos são classificados em três tipos: fundamentais, de apoio e organizacionais representado na Figura abaixo. [NBR12207]



NBR ISO/IEC 12207 - Processos do Ciclo de Vida do Software.

2.6 MODELO DE QUALIDADE DE SOFTWARE: CMMI - CAPABILITY MATURITY MODEL INTEGRATION

Originalmente definido na v1.1 em 1993 por Paulk, o Modelo de Maturidade e Capacidade para Software (CMM – *Capability Maturity Model for Software*) foi baseado em seis anos de experiência com melhoria de processo de software e a contribuição de centenas de revisores.

O CMMI é o modelo de maturidade sugerido atualmente com o fim de unificar e agrupar as diferentes usabilidades do CMM e de outros modelos de processos de melhoria corporativo, tais como o ISO 9001. São raras as empresas no mundo que conseguem atingir o nível 5, a grande maioria fica nos estágios iniciais. No Brasil, até o ano (2007), existiam somente 4 empresas que tinham alcançado o nível 5 do CMMI.

O CMMI determina práticas recomendadas em um certo número de **Áreas Chaves de Processo (KPA – Key Process Area)**, que provê a organização do desenvolvimento de software, baseada nas melhores estratégias para controlar, manter e evoluir os processos administrativos do desenvolvimento de software.

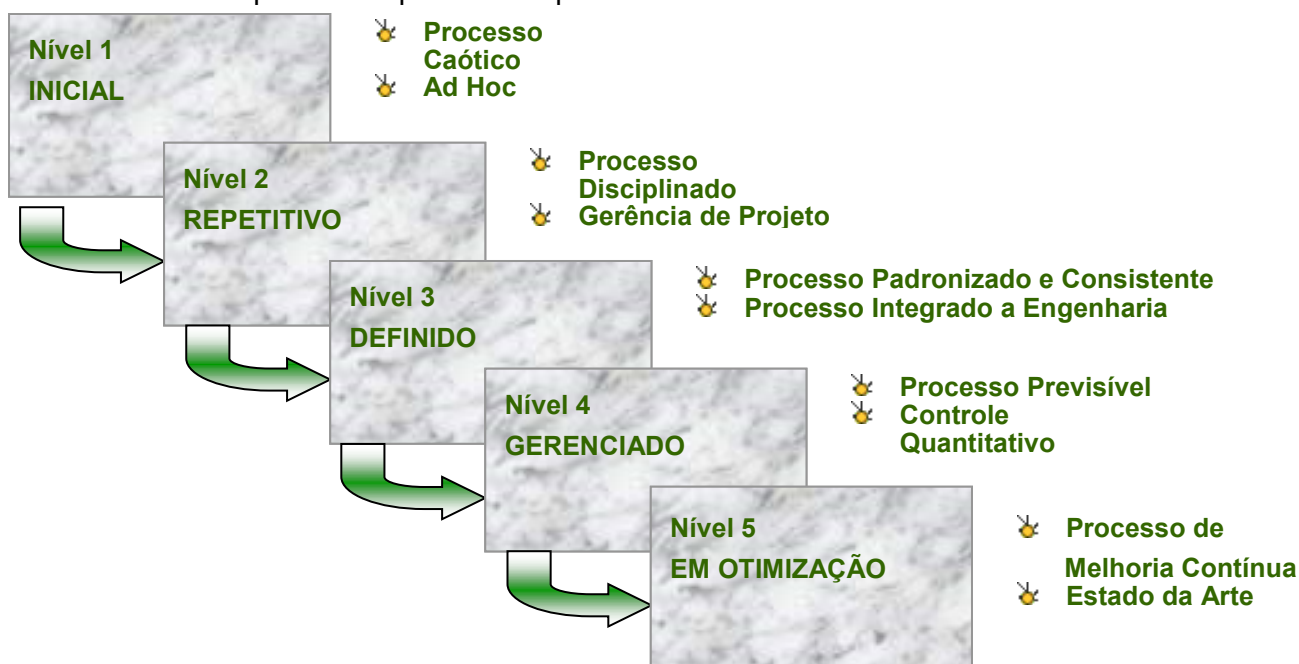
Fases do processo de maturidade:

- **Capacidade do processo de software** – É uma descrição de resultados previstos e resultados que podem ser arquivados para alcançar os objetivos do projeto, que devem ser arquivados para serem utilizados em futuros projetos.
- **Desempenho do processo de software** – Representa o resultado atual comparado com o resultado previsto. Quanto mais o resultado atual se aproximar do resultado previsto, melhor é o desempenho do processo de software.
- **Maturidade do processo de software** – Indica o quanto o processo de software está explicitamente definido. A maturidade mede a consistência entre os resultados previstos antes de se começar o processo com aqueles obtidos após o fechamento do processo. Deve prover de ferramentas administrativas que garantem a medição, o controle e a efetivação do processo.

2.6.1 Os cinco níveis de maturidade

Os níveis de maturidade estão baseados em pequenas etapas no lugar de inovações revolucionárias. Estes níveis estão organizados de forma que garantem a melhoria contínua do processo.

Cada nível de maturidade constitui em um conjunto de metas que quando satisfeita estabilizam um componente importante do processo.



Os cinco níveis de maturidade do CMMI.

2.6.2 ACP – Áreas Chave de Processo (KPA – Key Process Area)

Gerencial Planejamento de projeto de <i>software</i> e gerência	Organizacional Revisão e controle pela gerência sênior	Engenharia de <i>software</i> , Especificação, <i>design</i> , codificação e controle de qualidade
Nível 1 – Inicial (Processo Caótico - indica a fase de iniciação da implementação dos processos)		
Nível 2 – Repetitivo (Processo Disciplinado)		
<ul style="list-style-type: none"> Acompanhamento e Controle de Projeto Gerência de Configuração Gerência de Requisitos Gerência de Subcontratação Medição e e Análise Planejamento do Projeto Garantia de Qualidade de Processo e Produto 		
Nível 3 – Definido (Processo Padronizado e Consistente)		
<ul style="list-style-type: none"> Análise de Decisão e Resolução Gerência de <i>software</i> Integrada 	<ul style="list-style-type: none"> Definição do Processo da Organização Foco no Processo da Organização Programa de Treinamento Validação 	<ul style="list-style-type: none"> Desenvolvimento de requisitos Gerenciamento de Risco Integração do produto Solução técnica dos requisitos Verificação
Nível 4 – Gerenciado (Processo Previsível)		
<ul style="list-style-type: none"> Gerenciamento Quantitativo de Processos 	<ul style="list-style-type: none"> Desempenho do processo organizacional 	
Nível 5 – Otimização (Melhoria Contínua)		
	<ul style="list-style-type: none"> Gestão do processo organizacional 	<ul style="list-style-type: none"> Análise e Resolução de Causas

2.7 MODELO DE QUALIDADE DE SOFTWARE: SPICE – ISO 15504

A **ISO/IEC 15504**, também conhecida como **SPICE** – *Software Process Improvement & Capability dEtermination* (Melhoria do Processo de Software e Determinação da Capacidade) é a norma que define uma série de atividades para **manter a qualidade de software**, foi publicada em outubro de 2003, mas foi seu projeto começou em 1993 pela ISO com base nos modelos já existentes como ISO 9000 e CMMI. Ela é uma “evolução” da ISO/IEC 12207, que possui níveis de capacidade para cada processo.

O **framework** inclui um modelo de referência, que serve de base para o processo de avaliação. Um conjunto processos fundamentais que orientam para uma boa engenharia de *software*, e estabelece duas dimensões:

1. Dimensão de processo e a
2. Dimensão de capacidade.

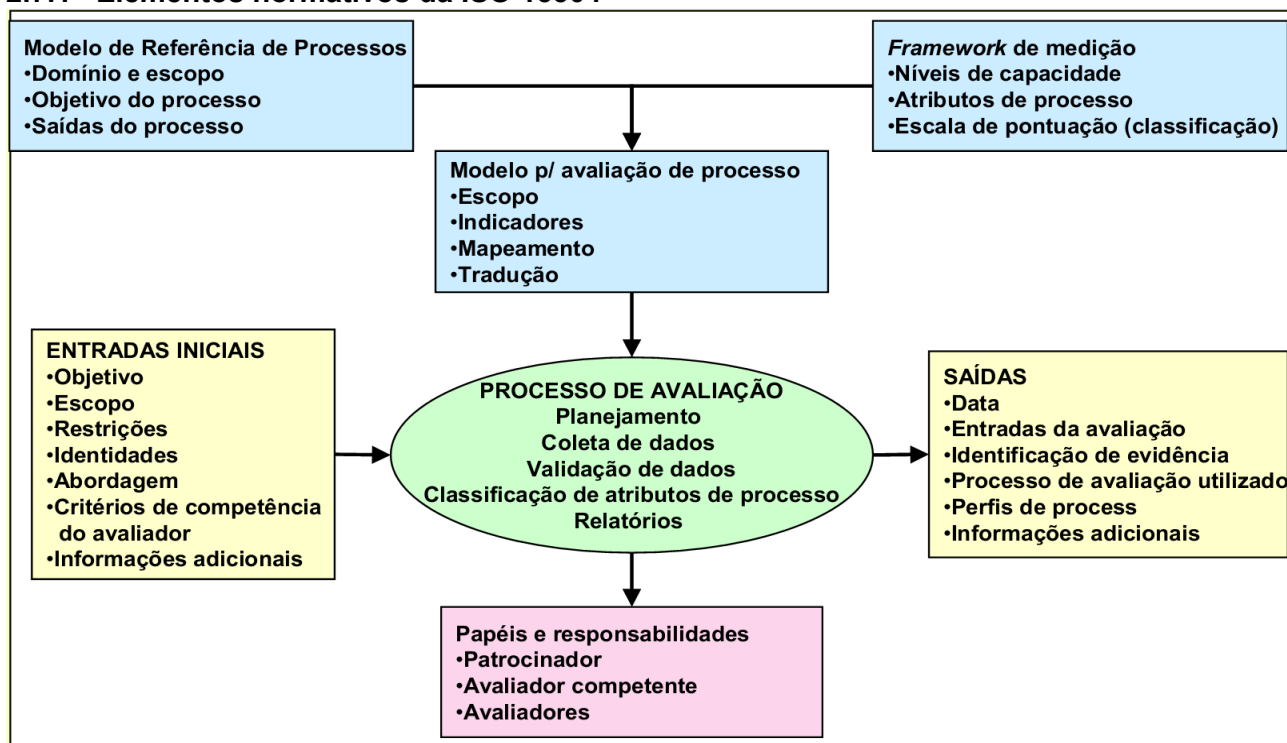
Na **dimensão de processo** o modelo é dividido em **cinco grandes categorias de processo**:

- Cliente-Fornecedor,
- Engenharia,
- Suporte,
- Gerência e
- Organização.

Na **dimensão de capacidade** o objetivo é avaliar a capacitação da organização em cada processo e permitir a sua melhoria. O **modelo de referência do SPICE inclui seis níveis de capacitação** dos processos:

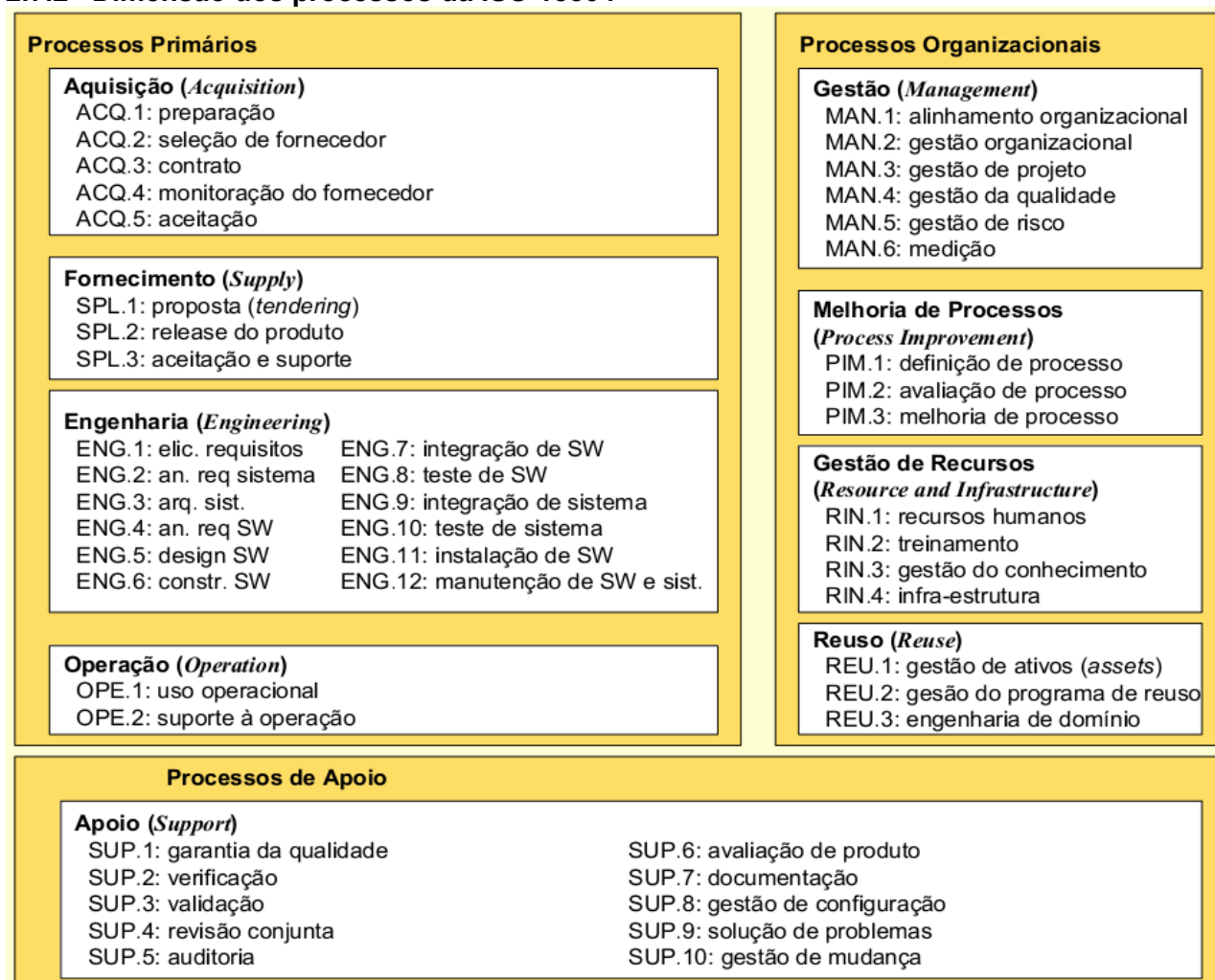
- Incompleto,
- Realizado,
- Gerenciado,
- Estabelecido,
- Previsível ou
- Otimizado.

2.7.1 Elementos normativos da ISO 15504



Elementos normativos da ISSO 15504 – SPICE.

2.7.2 Dimensão dos processos da ISO 15504



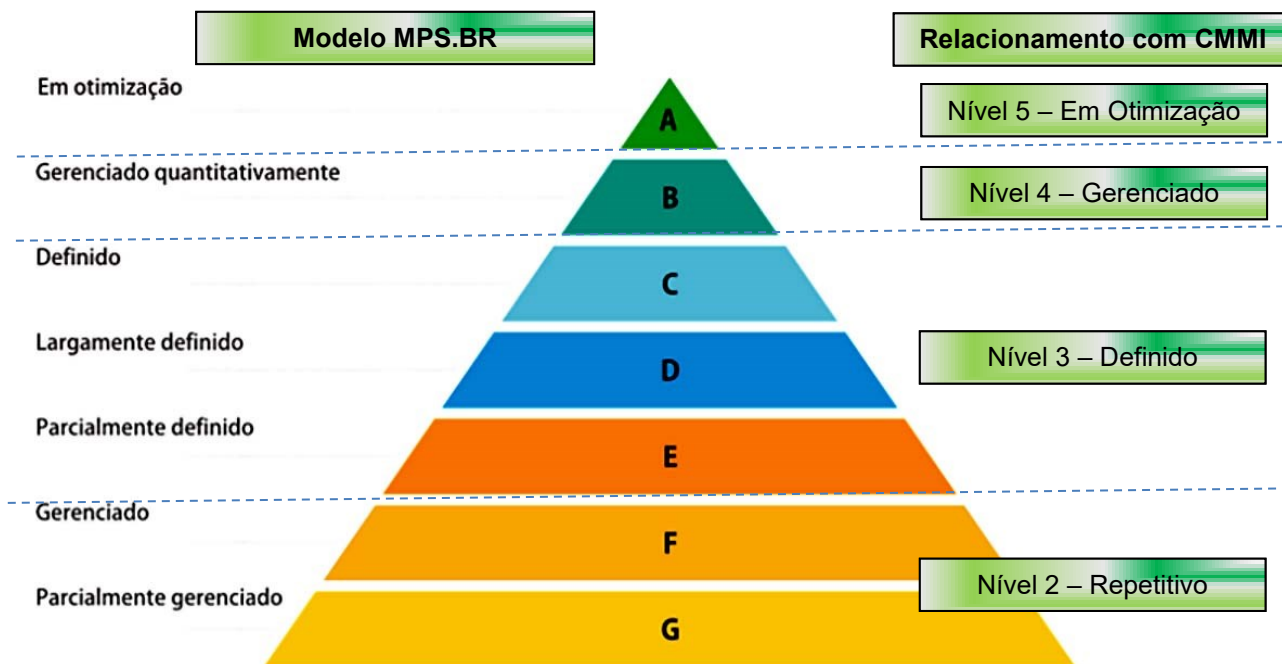
Dimensão dos processos da ISO 15505 - SPICE

2.8 MODELO DE QUALIDADE DE SOFTWARE: MPS.BR

O MPS-BR (Melhoria do Processo de Software Brasileiro) é uma metodologia voltada à área de desenvolvimento de sistemas e que foi criada por um conjunto de organizações ligadas ao desenvolvimento de software. Dentre as instituições envolvidas pode-se citar: a Softex (SP), a RioSoft (RJ), o COPPE/UFRJ (RJ) e o CESAR (PE). Estas organizações são não governamentais e muitas vezes de origem acadêmica, possuindo uma atuação de destaque junto à comunidade de *software* brasileira.

Enfatiza-se, dentro do MPS-BR, o uso das principais abordagens internacionais voltadas para a definição, a avaliação e a melhoria dos processos de software. O MPS-BR é compatível com as práticas do CMMI, como mostra a Figura abaixo com uma estrutura de níveis de maturidade, de forma similar àquela existente dentro do CMMI.

2.8.1 Modelo de maturidade da MPS-BR



Modelo MPS.BR definido em 7 níveis e seus relacionamentos com CMMI.

BIBLIOGRAFIA

- [CMM00] CMMI Product Development Team. **CMMI for Systems Engineering/Software Engineering, Version 1.02. Staged Representation**. Pittsburgh, U.S.A: Carnegie Mellon, SEI, 2000.
- CÔRTEZ, Mario L. **Modelos de Qualidade de Software: ISO 15504**. São Paulo, IC-UNICAMP.
- [FIO99] FIORINI, Soeli T.; STAA, Arndt von; BAPTISTA, Renan Martins. **Engenharia de Software com CMM**. Rio de Janeiro: BRASPORT, 1999.
- [NBR 9126-1] ABNT – Associação Brasileira de Normas Técnicas. **NBR ISO/IEC 9126-1 – Engenharia de software – Qualidade de produto**. Rio de Janeiro: ABNT, Junho de 2003.
- [NBR9126/14598] ABNT – Associação Brasileira de Normas Técnicas. **Guia para utilização das normas sobre avaliação de qualidade de produto de software - NBR ISO/IEC 9126 e NBR ISO/IEC 14598**. Curitiba: ABNT, Maio de 1999.
- [NBR 12207] ABNT – Associação Brasileira de Normas Técnicas. **NBR ISO/IEC 12207 – Tecnologia de informação - Processos de ciclo de vida de software**. Rio de Janeiro: ABNT, 1998.
- [PAU93] PAULK, Mark C.; CURTIS, Bill; CHRISSIS, Mary Beth; WEBER, Charles V. **The Capability Maturity Model for Software v1.1**. Pittsburgh – USA: SEI - Software Engineering Institute Customer Relations, 1993. Artigo.
- [PFL04] PFLEEGER, Shari Lawrence. **Engenharia de Software. – 2ª ed.** São Paulo: Prentice Hall, 2004.
- [PRE07] PRESSMAN, Ph.D. Roger S. **Engenharia de Software. – 6ª ed.** Rio de Janeiro: McGraw-Hill, 2007.
- [PRE02] PRESSMAN, Ph.D. Roger S. **Engenharia de Software. – 5ª ed.** Rio de Janeiro: McGraw-Hill, 2002.
- [QPS02] MCT. **Qualidade e Produtividade no Setor de Software Brasileiro N.4 (2002)**. Brasília: Ministério da Ciência e Tecnologia / Secretaria de Política de Informática, 2002.
- [SAL01] SALVIANO, Clenio F. **Introdução aos modelos de Processo de Software: ISO/IEC 15504 (SPICE), CMM e CMMI**. São Paulo: Simpósio, ITI - Instituto nacional de Tecnologia da Informação, 2001.
- [SOM03] SOMMERVILLE, Ian. **Engenharia de software**. São Paulo: Pearson Addison Wesley, 2003.