

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
ESCOLA DE INFORMÁTICA APLICADA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

MODELOS DE QUALIDADE DE SOFTWARE

LUIZ FERNANDO SALGUEIRO E MARCUS VINÍCIUS DO ARTE

ORIENTADOR: PROF. MARCIO BARROS

Agosto – 2005

MODELOS DE QUALIDADE DE SOFTWARE

Projeto de Graduação apresentado à Escola
de Informática Aplicada da Universidade
Federal do Estado do Rio de Janeiro
(UNIRIO) para obtenção do título de
Bacharel em Sistemas de Informação

LUIZ FERNANDO SALGUEIRO E MARCUS VINÍCIUS DO ARTE

ORIENTADOR: PROF. MARCIO BARROS

Sumário

1	Introdução	07
1.1	Objetivos	08
1.2	Estrutura do texto	08
2	Qualidade de Software	09
2.1	Processo	10
2.2	Projeto	11
2.3	Ciclo de Vida	13
2.4	Controle da Qualidade	14
2.5	Garantia da Qualidade	15
2.6	Qualidade dos Produtos de Software	16
2.7	Métricas	18
2.8	Conclusão	19
3	Norma	21
3.1	Norma ISO	21
3.2	Histórico das normas ISO	22
3.3	Norma ISO 12207	23
3.3.1	Processos fundamentais	25
3.3.2	Processos de apoio	26
3.3.3	Processos organizacionais	27
3.4	NBR ISO/IEC 9126	28
3.4.1	Subcaracterísticas de Qualidade de Software segundo a ISO/IEC 9126	28
3.5	ISO 9001:2000	30
3.5.1	Responsabilidade da Administração	32
3.5.2	Gerenciamento dos Recursos	33
3.5.3	Realização do Produto	34
3.5.4	Medição, Análise e Melhoria	34
3.6	Modelo de Qualidade CMM-Capability Maturity Model	35
3.6.1	Maturidade	36
3.6.2	Níveis	36
3.6.3	Descrição dos Níveis CMM	37

3.6.4	Entendendo os níveis de maturidade.....	39
3.7	CMMI.....	41
4	Conclusão.....	45
	Referências Bibliográficas.....	48

Lista de Figuras

Figura 1	Processo de software.....	10
Figura 2	Histórico Normas ISO.....	22
Figura 3	Processo do ciclo do vida do software.....	24
Figura 4	Características da qualidade de software.....	28
Figura 5	Ciclo PDCA.....	30
Figura 6	Organizações maduras X Organizações imaturas.....	36
Figura 7	Níveis de Maturidade.....	38
Figura 8	Organizações certificadas.....	40

Resumo

As necessidades e exigências de mercado têm influenciado as empresas produtoras e prestadoras de serviços de software a alcançar alto patamar de qualidade e produtividade para enfrentarem a competitividade no setor. Muitas dessas organizações buscam novos paradigmas que conduzam a uma melhoria contínua e progressiva da qualidade de seus processos, atenuando os problemas no desenvolvimento de seus produtos.

Pensando nisso, resolvemos realizar uma pesquisa sobre a qualidade de software e os principais padrões existentes, hoje, no mercado. Serão apresentadas abordagens importantes, sobre as normas ISO 9000, os modelos CMM e CMMI, destacando suas principais características e aplicabilidade.

O tema proposto faz parte do cotidiano da organização em que trabalhamos, sendo considerado um de nossos objetivos em relação ao plano de carreira profissional. Acreditamos que este foi um dos grandes fatores de motivação para a pesquisa realizada.

1 Introdução

A Engenharia de Software, embora possua várias definições, é uma disciplina que pode ser vista de forma objetiva como o uso dos princípios básicos da engenharia com a finalidade de desenvolver software de maneira sistemática e econômica, resultando em produtos confiáveis e eficientes [Pressman, 2002].

Ela agrega três elementos fundamentais:

- Métodos – Fornecem uma base técnica de como construir o software. Os métodos incluem um amplo conjunto de tarefas, que abrange análise de requisitos, projeto, construção de programas, teste e manutenção. Métodos de engenharia de software repousam num conjunto de princípios básicos, que regem cada área da tecnologia e incluem também atividades de modelagem;
- Ferramentas – Oferecem suporte aos métodos. Quando ferramentas são integradas, de modo que a informação criada por uma delas possa ser usada por outra, um ambiente para apoio ao desenvolvimento de software é estabelecido;
- Processos – Estabelecem as ligações ente os métodos e as ferramentas. O processo é o adesivo que mantém unidas as camadas de tecnologia e permite o desenvolvimento racional e oportuno de sistemas para computadores.

Obter qualidade nos processos e produtos de software não é uma tarefa trivial. O desconhecimento de ferramentas e a resistência aos processos internos dentro de um ambiente organizacional são alguns dos fatores que dificultam atingir os objetivos de qualidade. No entanto, nada é mais decepcionante do que produzir um software que não satisfaça as necessidades dos clientes.[Maldonado, 2001]

As características da organização, do pessoal técnico e do domínio da aplicação podem causar impacto na qualidade e na produtividade de processos e, conseqüentemente, dos produtos. Além de conhecer os métodos e processos de engenharia de software, é preciso saber como avaliá-los para propiciar uma base sólida à evolução do processo. A

disponibilidade de uma base histórica de dados abrangendo as diversas perspectivas e os fatores que afetam o desenvolvimento também é um elemento indispensável nesse cenário.

1.1 Objetivos

Com base no conhecimento adquirido, este trabalho possui os seguintes objetivos:

- Compreender o papel da qualidade e a sua importância no desenvolvimento de sistemas;
- Identificar os principais fatores para um produto de software de qualidade;
- Apresentar alguns modelos de qualidade do processo de desenvolvimento software, destacando seus principais tópicos.

1.2 Estrutura do texto

Produzir software de qualidade exige, também, o entendimento do que é qualidade de software. A primeira parte do nosso trabalho foi baseada em uma pesquisa voltada para os conceitos de: qualidade, processo, ciclo de vida, projeto, controle e garantia da qualidade e, por último, métricas. Apresentaremos as partes envolvidas em cada um destes itens e suas principais características.

O foco da segunda parte deste documento é direcionada aos modelos e normas para avaliação e melhoria de processos de software, visto que a conquista de uma certificação para uma empresa significa alcançar um padrão internacional de qualidade.

2- Qualidade de Software

Qualidade é, fundamentalmente, a adequação ao uso; ou ainda, a totalidade das características de um produto ou serviço que se relacionam com sua capacidade de atender às necessidades de um cliente [Campos,1992]. A qualidade é atualmente o principal fator que o consumidor busca quando decide pela aquisição de um produto ou serviço e pode ser entendida como o atendimento das necessidades ou aspirações do usuário, seja ele uma pessoa ou uma empresa.

A qualidade deve ser incorporada a um produto ou serviço e requer a conjugação de esforços de todos os membros de uma empresa, a integração de todas as funções e recursos que ela dispõe, desde a alta administração até o mais simples empregado. A estruturação de recursos, métodos, passos e equipamentos envolvidos com a qualidade constitui o sistema de qualidade da empresa.

Demonstrar que a empresa é capaz de atingir e manter a qualidade de seus produtos e serviços e continuamente aperfeiçoá-la é um corolário deste esforço e, por sua repercussão na produtividade e competitividade, vem merecendo permanente cuidado das organizações líderes em suas áreas de atividade. O comprometimento com a política de qualidade é uma responsabilidade do mais alto nível da administração.

Os elementos do sistema de qualidade devem estar estruturados para estabelecer um controle adequado e uma garantia sobre todos os processos operacionais que afetam a qualidade do produto ou serviço. Ações preventivas devem ser enfatizadas para que se evite a ocorrência de problemas sem sacrificar a capacidade de responder e de corrigir falhas que possam ocorrer.

2.1 Processo

Negócios são viabilizados por intermédio de processos. Os processos, portanto, são importantíssimos em uma organização bem sucedida. É comum o entendimento de processo como uma seqüência de etapas executadas para realizar um determinado objetivo [Campos, 1992]. Em um ambiente organizacional existem diversos processos, de produção, de gerência, de controle, de planejamento estratégico, etc.

Processos são a rede produtiva de uma organização, onde cada nó é um ponto de partida para outros novos processos, cada qual com suas características, peculiaridades e tecnologias específicas, mas onde todos concorrem para um único fim: o negócio.

O processo de software envolve métodos, ferramentas e pessoas. Esse conjunto deve estar integrado para que um processo funcione de maneira eficaz. Para um processo funcionar corretamente, ele deve possuir:

- Procedimentos e métodos que descrevam a relação entre as tarefas;
- Ferramentas e equipamentos que ofereçam suporte à realização das tarefas, simplificando e automatizando o trabalho;
- Pessoas com perfil adequado, treinadas nos métodos e nas ferramentas para poderem realizar as atividades previstas, adequadamente.



Figura 1 - Processo de Software (adaptado Paulk, 1995).

2.2 - Projeto

Um projeto é um empreendimento temporário com o objetivo de criar um produto ou um serviço único. Pode ser considerado temporário porque possui um começo e um fim bem definidos. Pode ser único porque o produto ou serviço produzido é de alguma forma diferente de todos os outros produtos ou serviços semelhantes [Zwass, 1998]. É uma representação de engenharia de algo que vai ser construído. Os projetos são desenvolvidos em todos os níveis da organização e podem envolver uma única pessoa ou milhares delas.

Os projetos têm um ciclo de vida formado por etapas distintas, que se subdividem em tarefas ou atividades que podem ou não ser executadas de uma única vez. Um projeto é composto por uma seqüência de atividades, cujo planejamento é essencial para que ele se encerre em um prazo e custo adequados.

Projetos possuem um conjunto de etapas pelas quais progridem e, no final de cada etapa, um produto é concluído. O número de fases, seus nomes e seus inter-relacionamentos variam de caso a caso. De forma geral podemos dizer que um projeto possui quatro etapas genéricas [Zwass, 1998] :

- *Conceitualização* - formulação do escopo do projeto, definição do problema e de uma possível solução para o mesmo. Nesta fase é realizado um estudo da viabilidade e um plano estratégico é apresentado;
- *Desenvolvimento* - com base nos resultados e conclusões da fase anterior, inicia-se o desenvolvimento dos componentes e a integração das partes envolvidas como um todo;
- *Execução* - é realizada uma simulação do que irá ser implementado. São feitos alguns testes para confirmar se o projeto foi construído corretamente dentro do que foi solicitado;
- *Finalização* - é a considerada a última etapa tendo como objetivo corrigir os erros que foram identificados na fase anterior. Com base nos requerimentos completos produz-se um desenho final para a solução.

As partes envolvidas num projeto são indivíduos e organizações diretamente envolvidas ou aqueles cujos interesses podem ser afetados, de forma positiva ou negativa, no decorrer do seu desenvolvimento ou mesmo após a sua conclusão. Essas partes podem também exercer variadas influências no projeto e nos seus resultados [PMBOK, 2003]. A equipe de gerência do projeto deve identificar as partes envolvidas, conhecer suas necessidades, expectativas e, então, gerenciar e influenciar os requisitos de forma a garantir o seu pleno sucesso. Vejamos quais são as partes envolvidas em um projeto:

- Gerente do projeto – indivíduo responsável pela gerência do projeto;
- Cliente – indivíduo ou organização que fará uso do produto do projeto. Podem existir muitos tipos de clientes. Por exemplo, os clientes de um novo produto farmacêutico incluem os médicos que o prescrevem, os pacientes que o tomam e as companhias de seguro que pagam por ele. Em muitas áreas de aplicação, clientes e usuários são sinônimos, enquanto em outras o termo cliente se aplica à entidade que comprou o resultado do projeto, enquanto usuários são aqueles que usarão diretamente o produto do projeto;
- Organização executora – empresa cujos funcionários estarão mais diretamente envolvidos na execução do projeto;
- Membros da equipe do projeto – o grupo que realiza o trabalho do projeto;
- Patrocinador – indivíduo ou grupo dentro da organização executora que provê os recursos financeiros, em dinheiro ou espécie, para o projeto.

No contexto da engenharia de software, a maioria das organizações encontra os seguintes projetos: [Pressman,2002]

- Projetos de desenvolvimento conceitual: são iniciados para explorar algum conceito novo de negócio ou a aplicação de alguma tecnologia nova;
- Projetos de desenvolvimento de novas aplicações: são efetuados como consequência de um pedido específico de cliente;

- Projetos de aperfeiçoamento de aplicações: ocorrem quando o software existente passa por modificações importantes na função, desempenho ou interfaces que são observáveis pelo usuário final;
- Projetos de manutenção de aplicações: corrigem, adaptam ou ampliam o software existente em aspectos que não são imediatamente óbvios ao usuário final;
- Projetos de reengenharia: são efetuados com o objetivo de reconstruir um sistema existente (herdado) no seu todo ou em parte.

2.3 - Ciclo de Vida

O conjunto de fases de um projeto é conhecido como ciclo de vida do projeto [PMBOK, 2003]. Um projeto normalmente é dividido em várias fases visando um melhor controle gerencial e uma ligação mais adequada de cada um dos seus processos operacionais. No ciclo de vida dos projetos, alguns pontos são definidos: que trabalho deve ser realizado; quem deve estar envolvido em cada fase; qual o valor estimado do seu custo; quais as datas de início e conclusão das atividades; quais os riscos etc.

O processo do ciclo de vida de software tem como principal objetivo fornecer uma estrutura comum para que o adquirente, fornecedor, desenvolvedor, mantenedor, operador, gerentes e técnicos envolvidos com o desenvolvimento de software utilizem uma linguagem comum. Esta linguagem comum é estabelecida na forma de processos bem definidos.

Os processos que envolvem o ciclo de vida do desenvolvimento de um produto de software são agrupados em três classes que representam sua natureza: processos fundamentais, processos de apoio e processos organizacionais [ISO 12207]. Cada processo é definido em termos de suas próprias atividades, e cada atividade é definida em termos de suas tarefas. Uma abordagem mais detalhada sobre essas três classes será apresentada nos próximos capítulos.

2.4 Controle da Qualidade

O controle da qualidade é um modelo gerencial centrado no controle do processo, tendo como meta a satisfação das necessidades das pessoas [Campos,1992]. O objetivo mais importante deste controle é garantir a qualidade do produto para o seu cliente externo ou interno.

A prática consciente do controle de qualidade por todas as pessoas da empresa, assumindo responsabilidades sobre os resultados (do seu processo) e autoridade (meios) sobre o seu processo, é a base do gerenciamento participativo e o pilar de sustentação para o controle da qualidade [Campos,1992].

O controle da qualidade possui três objetivos principais :

- *Planejar a qualidade desejada pelos clientes:* isto implica num esforço de localizar o cliente, saber de suas necessidades, traduzir estas necessidades em características mensuráveis, de forma que seja possível gerenciar todo processo;
- *Manter a qualidade desejada pelo cliente,* cumprindo os padrões previamente acordados;
- *Melhorar a qualidade desejada pelo cliente:* neste caso é preciso localizar os resultados indesejáveis e utilizar um método de solução de problemas com a finalidade de melhorá-los.

O controle da qualidade envolve também a monitoração dos resultados específicos do projeto, para determinar se eles estão de acordo com os padrões de qualidade previamente estabelecidos, além de identificar formas de eliminar as causas dos resultados insatisfatórios. Este controle deve ser executado ao longo de todo o projeto.

2.5 Garantia da Qualidade

A garantia da qualidade é um processo sistemático de verificação para certificar-se de que a inspeção da qualidade e as operações de controle estão sendo conduzidas de forma correta, além de verificar também se os setores de projeto, produção e vendas estão trabalhando em conjunto, alinhados a um mesmo objetivo [Maldonado,2001]. Essa prática é considerada uma atividade essencial para qualquer negócio onde um produto ou serviço será colocado no mercado e será comercializado.

A meta da garantia da qualidade é fornecer à gerência os dados necessários para que fique informada sobre a qualidade do produto, ganhando compreensão e confiança de que a qualidade do produto está satisfazendo suas metas [Pressman, 2002].

A garantia da qualidade é conseguida pelo gerenciamento correto e obstinado de todas as atividades de qualidade em cada projeto e em cada processo, buscando, sistematicamente, eliminar totalmente as suas falhas, pela constante preocupação com a satisfação total das necessidades do consumidor e pela participação e responsabilidade de todos na empresa.

Portanto, a garantia da qualidade envolve alguns pontos importantes [Campos,1992]:

- *Tradição*: um cliente só poderá comprar um produto ou serviço com confiança se a empresa tornar-se conhecida por sua confiabilidade ao longo do tempo. Não se consegue tradição da noite para o dia. A tradição é um conceito comercialmente tão importante que fica difícil avaliar os prejuízos decorrentes de um produto defeituoso colocado no mercado. Isto tem que ser evitado a todo o custo;
- *Satisfação total das necessidades do consumidor*: o produto ou serviço além de não possuir defeitos ou falhas, precisa satisfazer as características verdadeiras da qualidade, ou seja, satisfazer às expectativas do consumidor. Sob este aspecto, é bom lembrar que as expectativas do consumidor não são fixas; elas mudam. Daí o cuidado que se deve ter com propaganda comercial, conteúdo do catálogo de vendas e treinamento de pessoal, para que não sejam criadas expectativas negativas.

Para que uma empresa possa dizer que tem garantia da qualidade ela deve ser capaz de:
[Campos,1992]

- Detectar necessidades humanas não atendidas;
- Especificar produtos/serviços que satisfaçam a estas necessidades;
- Projetar estes produtos/serviços ao mais baixo custo possível;
- Projetar e operar processos que fabriquem este produtos/serviços conforme as especificações, ao mais baixo custo;
- Inspeccionar os produtos de tal forma a verificar a sua conformidade com as especificações;
- Dar assistência técnica e total atenção ao seu cliente, de tal forma a assegurar a sua satisfação por um longo tempo.

2.6 Qualidade dos Produtos de Software

Qualidade de software pode ser vista como um conjunto de características que devem ser alcançadas em um determinado grau para que o produto atenda às necessidades de seus usuários [Weber, 2001].

Ao se desenvolver um produto de software, o objetivo não é alcançar a qualidade perfeita, mas sim a qualidade necessária e suficiente para o uso especificado, quando o produto for entregue e realmente usado pelos usuários [Maldonado,2001]. É necessário identificar quais são as características de qualidade necessárias para um determinado produto de software e definir em que grau essas características precisam ser alcançadas para satisfazer às necessidades dos usuários.

Podemos classificar um produto de software segundo as seguintes características [Maldonado, 2001]:

- *Funcionalidade*: refere-se à existência de um conjunto de funções, que satisfazem às necessidades explícitas ou implícitas e suas propriedades específicas;

- *Confiabilidade*: refere-se à capacidade de o software manter seu nível de desempenho sob condições estabelecidas, por um período de tempo;
- *Usabilidade*: refere-se ao esforço necessário para usar um produto de software, bem como o julgamento individual de tal uso por um conjunto explícito ou implícito de usuários;
- *Eficiência*: refere-se ao relacionamento entre o nível de desempenho do software e a quantidade dos recursos utilizados sob as condições estabelecidas;
- *Manutenibilidade*: refere-se ao esforço necessário para fazer modificações específicas no software;
- *Portabilidade*: refere-se à capacidade do software ser transferido de um ambiente para o outro;
- *Efetividade*: refere-se à capacidade do produto de software possibilitar aos usuários atingir metas especificadas com acurácia e completeza em um contexto de uso especificado;
- *Produtividade*: refere-se à capacidade de possibilitar aos usuários utilizar uma quantidade adequada de recursos;
- *Segurança*: refere-se à capacidade de oferecer níveis aceitáveis de risco de danos a pessoas, negócios, software, propriedade ou ao ambiente especificado;
- *Satisfação*: refere-se à capacidade de satisfazer os usuários em um contexto de uso especificado.

Muito tem sido escrito sobre qualidade e seus vários adjetivos. No entanto, o fundamental é que o software seja confiável, ou seja, que seja eficaz e siga os padrões exigidos pelo contexto onde irá atuar.

2.7 Métricas

As métricas e estimativas de software vêm se tornando um dos principais tópicos na Engenharia de Software com a crescente exigência de seus consumidores pela qualidade, rapidez, comodidade e baixo custo de implantação e manutenção. É impossível não enxergar tais técnicas como alavanca para um produto de melhor qualidade, com custos adequados [Pressman, 2002].

Quando falamos de métricas, então, temos que ter em mente que se trata de dados (números) quantitativos que irão mostrar, em forma de indicadores, o estado atual de um determinado projeto [Pressman, 2002]. A busca pela qualidade utilizando métricas de software deve ser aplicada tanto às pessoas que produzem o produto quanto ao processo em que se desenvolve o mesmo produto. Com métricas podemos estimar prazos de entregas finais e ter uma visão global de todo o projeto.

Um projeto de software é um processo de tomada de decisão, onde as métricas são usadas para fornecer uma base de identificação de procedimentos que não estejam em conformidade com os alvos pretendidos e com as medidas de atributos de projetos, além de auxiliar na elaboração de novas soluções, que levam à melhoria da qualidade.

Quando falamos em métricas, devemos destacar os seguintes aspectos [Maldonado,2001] :

- *Tempo*: tempo total do projeto, tempo nas fases de análise, projeto, codificação testes, tempo em reuniões, tempo em retrabalho;
- *Precisão da estimativa de cronograma*: precisão da estimativa de cronograma de todo o projeto e de suas fases;
- *Esforço*: esforço total do projeto, dos recursos, de pessoal, esforço em reuniões de revisão, esforço em retrabalho;
- *Tamanho do sistema*: número de linhas de código; pontos de função ;

- *Número de erros*: número de erros na especificação de requisitos e no projeto do sistema, encontrados em reuniões de revisão, erros de código, encontrados nos teste de unidade;
- *Número de modificações*: número de modificações na especificação de requisitos, projeto ou código após sua aprovação;
- *Densidade de defeitos*: número de erros somados ao número de modificações em relação ao tamanho do sistema;
- *Rotatividade de pessoal*: percentual de pessoas que saíram, entraram ou mudaram de função durante o desenvolvimento do projeto;
- *Produtividade*: número de códigos produzidas por unidade de esforço;
- *Deterioração do software*: relação entre o esforço gasto para corrigir os problemas após a liberação do sistema comparado ao esforço gasto antes da liberação do software para o usuário;
- *Experiência da equipe*: experiência na linguagem de programação, no domínio da aplicação, nas ferramentas, no método e no processo de desenvolvimento, tempo total de experiência do profissional.

2.8 Conclusão

Vivemos, nos dias de hoje, tempo de mudanças. Mudanças muito rápidas e também muito radicais. Novas tecnologias se tornam obsoletas da noite para o dia. Processos sofrem modificações a todo o momento interferindo, nos produtos e serviços que são oferecidos a um mercado cada vez mais competitivo. Nas empresas, os custos são rigorosamente contabilizados. A realidade mostra, ainda, que os menores preços e a qualidade são diferenciais importantíssimos para uma clientela extremamente exigente.

Dentro desse contexto, as empresas especializadas no desenvolvimento de software, obrigatoriamente, têm que se manter atualizadas, perseguir os critérios de qualidade

exigidos pelo mercado, incorporando sempre tecnologias mais apropriadas, a fim de poder enfrentar, com sucesso, a feroz competitividade da atualidade.

No capítulo seguinte, trataremos de melhoria na qualidade do processo de desenvolvimento de software através de padrões e normas mais conceituados do mercado. Veremos que algumas características, tradicionalmente aceitas, são muito importantes para que se obtenha vantagens consideráveis sobre seus concorrentes.

3 - Norma

Uma norma é um documento estabelecido por consenso e aprovado por um organismo reconhecido, que fornece, para o uso comum e repetitivo, regras, diretrizes ou características para atividades ou resultados, visando à obtenção de um grau ótimo de ordenação em um dado contexto [ABNT-Associação Brasileira de Normas Técnicas,1996] . As normas são aprovadas por consenso por seus comitês técnicos, que partem de práticas já em andamento, reunindo as experiências dos diversos países-membros. O processo adotado faz com que estas normas sejam empregadas em praticamente todo o mundo, facilitando o intercâmbio dos produtos e serviços entre os países que a adotam.

3.1 - Norma – ISO

A sigla ISO representa uma organização internacional, não governamental, que elabora normas internacionais. Ela significa “Organização Internacional para Normalização” (International Organization for Standardization). A sigla é uma referência à palavra grega ISO, que significa igualdade. O propósito das normas ISO é desenvolver e promover preceitos e padrões mundiais que traduzam o consenso dos diferentes países do mundo de forma a facilitar o comércio internacional [Maldonado,2001]. No Brasil, a Associação Brasileira de Normas Técnicas (ABNT) é a representante brasileira das normas ISO.

Com a remoção de barreiras comerciais entre os países, novos aspectos nas relações comerciais estão sendo considerados. Há indícios de que a implantação das normas ISO seja um fator relevante à concretização de negócios em diversas partes do mundo: nos Estados Unidos, na Europa, na Ásia e na América Latina, principalmente em relação às transações comerciais do MERCOSUL. Os representantes de diferentes países discutem, analisam e buscam o consenso quanto ao conteúdo das normas ISO.

3.2 - Histórico das normas ISO

Durante os anos oitenta, a maioria das empresas fazia uso de TI (Tecnologia da Informação) para apoiar processos internos (ex. finanças, recursos humanos etc) de modo independente, desconectado de seus processos de negócios. O objetivo dos sistemas de qualidade daquela época era o controle da qualidade dos produtos.

A década de 90 mostra as empresas começando a usar TI para apoiar seus processos de negócio, com foco na cadeia produtiva (ex. produção, distribuição e gestão empresarial). Nesse período, o objetivo das empresas mudou, passando do controle de qualidade de produtos para a garantia da qualidade de processos relacionados a projetos, desenvolvimento, produção, instalação e serviços associados.

A primeira versão das normas internacionais ISO 9001, 9002 e 9003 para a garantia da qualidade foi lançada em 1987. A seguir, um breve histórico:

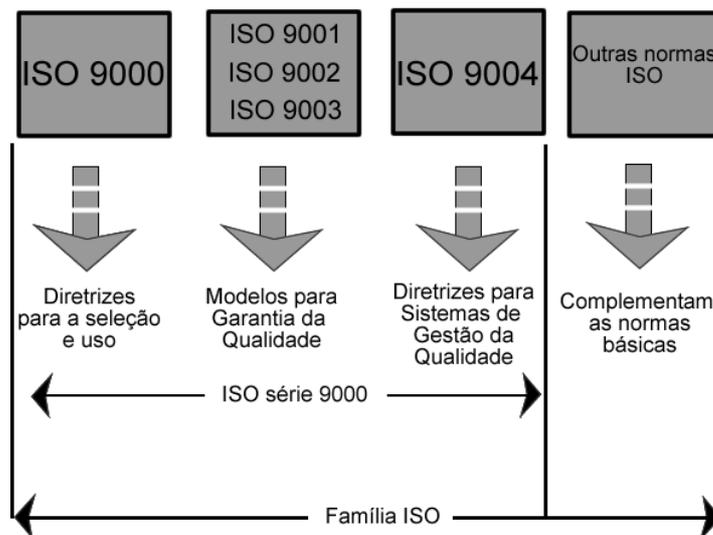


Figura 2 - Histórico Normas ISO .

Fonte: ISO série 9000 - Ao alcance de todos - 8ª edição. MCG Qualidade.

O uso das normas ISO cresceu rapidamente no Brasil. No ano de 1990, apenas 13 empresas possuíam 18 certificados ISO 9000. Já no ano de 2000, 4.744 empresas detinham 6.389 certificados [Weber, 2001], números esses que não param de crescer.

No período de 2000 a 2009, as empresas deverão usar TI como parte integrante dos processos de negócios ligados à cadeia de demanda (ex. relacionamento com clientes, gerência de contas, serviços de pós-venda e processamento de pedidos). Com a nova ISO 9001 versão 2000, os sistemas de gestão de qualidade deverão mudar o seu objetivo da garantia da qualidade para a satisfação dos clientes [Campos,1992].

O processo de certificação de uma empresa de software segundo as normas ISO 9001 segue um conjunto de passos bem definido [ISO, 2000]

- 1º A empresa estabelece o seu sistema de qualidade;
- 2º A empresa faz uma solicitação formal a um órgão certificador, incluindo detalhes do negócio da empresa, escopo da certificação solicitada e cópia do manual de qualidade;
- 3º O órgão certificador faz uma visita à empresa, colhe mais dados e explica o processo de certificação;
- 4º O órgão certificador verifica se a documentação do sistema de qualidade está de acordo com a norma ISO;
- 5º O órgão certificador envia uma equipe à empresa com fins de auditoria. Nesta visita, será verificado se todos na empresa cumprem o que está documentado no manual de qualidade;
- 6º O órgão certificador emite o certificado de qualidade;
- 7º O órgão certificador realiza visitas periódicas à empresa para assegurar que o sistema continua sendo efetivo.

3.3 - Norma ISO 12207

Sob o título de Information Technology - Software Life Cycle Process (tecnologia da informação - Processos de Ciclo de Vida de Software), a ISO 12207 foi criada para estabelecer uma estrutura comum de processos, sendo utilizada como referência em

negócios relacionados a produtos de software [Maldonado,2001]. O desenvolvimento e manutenção do software devem ser conduzidos de forma semelhante a engenharia, ou seja, este padrão formaliza a arquitetura do ciclo de vida do software, que é um assunto básico em Engenharia de Software.

A Norma NBR ISO/IEC 12207 - Processos do Ciclo de Vida do Software tem como principal objetivo fornecer uma estrutura comum para que o comprador, fornecedor, desenvolvedor, mantenedor, operador, gerentes e técnicos envolvidos com o desenvolvimento de software utilizem uma linguagem comum. Esta linguagem comum é estabelecida na forma de processos bem definidos [NBR ISO/IEC 12207].

Esta norma agrupa os processos de ciclo de vida do software em três classes, Processos Fundamentais, Processos de Apoio e Processos Organizacionais.

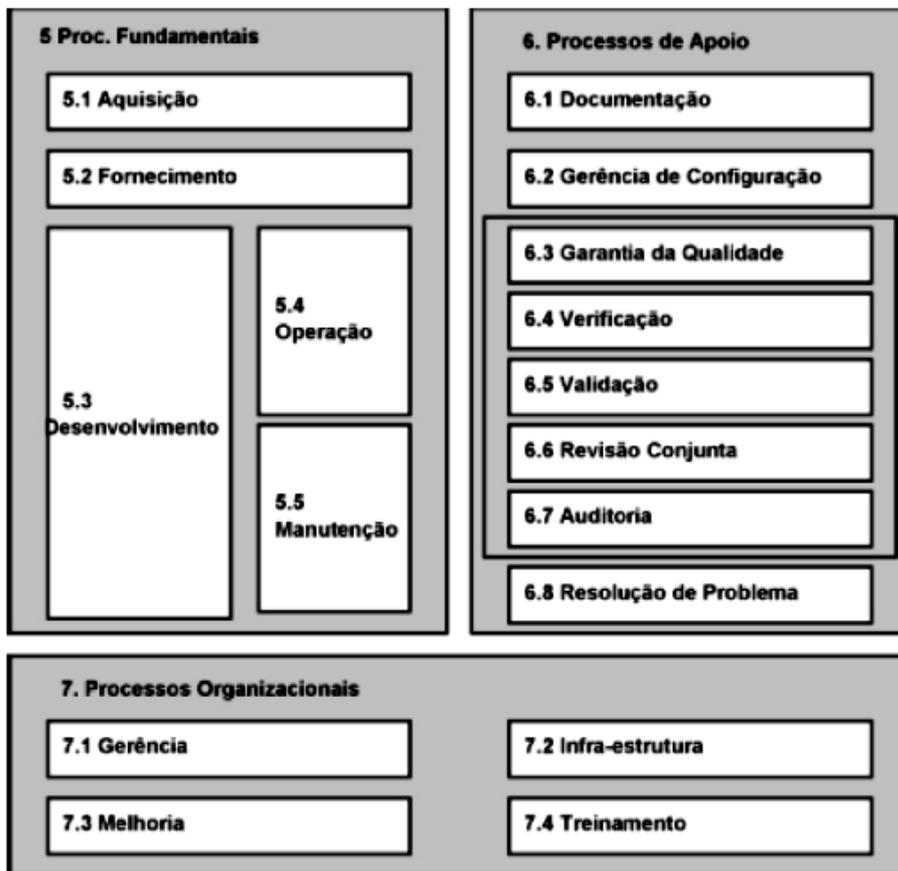


Figura 3 - Processos de ciclo de vida do software [ISO 12207].

3.3.1 - Processos Fundamentais

Os processos fundamentais marcam o início e execução do desenvolvimento, operação ou manutenção do software durante o seu ciclo de vida.

- *Processo de Aquisição*: atividades de quem adquire um software. Inclui a definição da necessidade de adquirir um software (produto ou serviço), o pedido de proposta, a seleção de fornecedor, a gerência da aquisição e a aceitação do software ou serviço de software;
- *Processo de Fornecimento*: define as atividades do fornecedor de software, ou seja, organização que provê o produto de software ao comprador. Inclui preparar uma proposta, a assinatura de contrato determinação de recursos necessários, o desenvolvimento e a execução dos planos de projeto, até a entrega do software;
- *Processo de Desenvolvimento*: esse processo define as atividades do desenvolvedor de software. Inclui a análise de requisitos, o projeto, a codificação, a integração, os testes, a instalação e a aceitação do software;
- *Processo de Operação*: define as atividades do operador. Inclui a operação de um sistema computacional para os usuários, assim como também o suporte operacional;
- *Processo de Manutenção*: define as atividades do mantenedor do software, isto é, gerenciamento de modificações no software para mantê-lo atualizado e em perfeita operação. Quando um sistema necessita de alterações relativas à melhoria, adaptação, ou qualquer alteração no código, esse processo é ativado. Em qualquer situação, sua integridade deve ser preservada.

3.3.2 - Processos de Apoio

Os processos de apoio auxiliam um outro processo e contribuem para o sucesso e qualidade do projeto de software. São empregados e executados, quando necessário por outros processos.

- *Processo de Documentação*: define as atividades para registrar informações produzidas por um processo ou atividade do ciclo de vida. O processo contém o conjunto de atividades que planeja, projeta, desenvolve, produz, edita, distribui e mantém os documentos necessários a todos os interessados, tais como gerentes, engenheiros e usuários do sistema ou produto de software;
- *Processo de Gerência de Configuração*: define atividades que identificam e controlam os itens do software. Inclui o controle de armazenamento, liberações, manipulação, distribuição e modificação de cada um dos itens que compõem o software, estabelecendo suas linhas básicas (baseline); garante a completeza, a consistência e a correção dos itens; controla o armazenamento, a manipulação e a distribuição dos itens.
- *Processo de Garantia da Qualidade*: define as atividades para garantir que os processos e produtos de software estejam em conformidade com os requisitos e os planos estabelecidos;
- *Processo de Verificação*: define as atividades para verificação dos produtos de software, se os mesmos estão atendendo aos requisitos especificados e às condições impostas a eles;
- *Processo de Revisão Conjunta*: define as atividades para avaliar a situação e os produtos de uma atividade de um projeto, se apropriadas. Estas avaliações ou revisões são realizadas nos níveis gerenciais do projeto e nos níveis técnicos, durante o contrato do projeto;

- *Processo de Auditoria*: define as atividades para determinar adequação aos requisitos, planos e contrato, quando for apropriado;
- *Processo de Resolução de Problemas*: define um processo para analisar e resolver os problemas descobertos durante a execução do desenvolvimento, operação, manutenção ou outros processos.

3.3.3 - Processos Organizacionais

Os processos organizacionais são utilizados com o intuito de melhorar continuamente a estrutura e os processos do ciclo de vida do software.

- *Processo de Gerência*: define as atividades que podem ser empregadas por quaisquer das partes que têm que gerenciar seu respectivo processo. Inclui gerenciamento de produto, gerenciamento de projeto e gerenciamento de tarefa do processo de apoio;
- *Processo de Infra-estrutura*: atividades que visam estabelecer o fornecimento de recursos necessários para outros processos. Inclui: hardware, software, ferramentas, técnicas, padrões de desenvolvimento, operação ou manutenção;
- *Processo de Melhoria*: define as atividades para estabelecer, avaliar, medir, controlar e melhorar um processo de ciclo de vida de software;
- *Processo de Treinamento*: define um conjunto de atividades para prover e manter o pessoal treinado. Inclui planejamento e implementação de programas de treinamento.

A norma detalha cada um dos processos acima. Ela define ainda como eles podem ser usados de diferentes maneiras por diferentes organizações (ou parte destas), representando diversos pontos de vista para esta utilização. Cada uma destas visões representa a forma como uma organização emprega estes processos, agrupando-os de acordo com suas necessidades e objetivos.

3.4 - NBR ISO/IEC 9126

A ISO/IEC 9126 fornece o significado de características de qualidade e métricas para avaliação de produtos de software. As métricas de qualidade são baseadas nos atributos internos de software e no comportamento externo do sistema de computador. Essas métricas são aplicáveis ao se especificar requisitos de qualidade e objetivos do produto de software. A seguir, encontra-se na **Figura 4** as características e algumas descrições que definem a qualidade de software segundo a ISO/IEC 9126.

Características	Descrição
Funcionalidade	Evidencia que o conjunto de funções atendem às necessidades explícitas e implícitas para a finalidade a que se destina o produto.
Confiabilidade	Evidencia que o desempenho se mantém ao longo do tempo e em condições estabelecidas.
Usabilidade	Evidencia a facilidade para a utilização de software.
Eficiência	Evidencia que os recursos e os tempos envolvidos são compatíveis com o nível de desempenho requerido para o produto.
Manutenibilidade	Evidencia a facilidade para correções, atualizações e alterações.
Portabilidade	Evidencia que é possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação.

Figura 4 - Características da qualidade de software [ISO/IEC 9126] .

As características são detalhadas em subcaracterísticas, para facilitar o entendimento do que é um software de qualidade.

3.4.1 - Subcaracterísticas da Qualidade de Software segundo a ISO/IEC 9126

- Funcionalidade
 - Adequação: Presença de conjunto de funções e sua apropriação para tarefas;

- Acurácia: Geração de resultados ou efeitos corretos;
- Conformidade: Estar de acordo com normas, convenções e regulamentações.
- Confiabilidade
 - Segurança de Acesso: Capacidade de evitar acesso não autorizado a programas e dados;
 - Maturidade: Indicação de baixa frequência de falhas;
 - Tolerância a falhas: Manter nível de desempenho em caso de falha;
 - Recuperabilidade: Capacidade de se restabelecer e restaurar dados após falha.
- Usabilidade
 - Inteligibilidade: Facilidade de entendimento dos conceitos utilizados;
 - Apreensibilidade: Facilidade de aprendizado;
 - Operacionalidade: Facilidade de operar e controlar a operação.
- Eficiência
 - Comportamento em relação a tempo: tempo de resposta, de processamento;
 - Comportamento em relação a recursos: medida da quantidade de recursos necessários (CPU, disco e memória, dentre outros) e a duração;
- Manutenibilidade
 - Analisabilidade: Facilidade de diagnosticar deficiências e causas de falhas;
 - Modificabilidade: Facilidade de modificação e remoção de defeitos;
 - Estabilidade: Ausência de riscos de efeitos inesperado;
 - Testabilidade: Facilidade de ser testado.
- Portabilidade
 - Adaptabilidade: Capacidade de ser adaptado a ambientes diferentes;
 - Capacidade para ser testado: Medida do esforço necessário para se instalar o produto;
 - Conformidade: Medida do nível de conformidade do produto com padrões referentes à portabilidade;
 - Capacidade para substituir: Medida de esforço necessário para usar o produto em substituição a outro, previamente especificado.

3.5 - ISO 9001:2000

No dia 15 de dezembro de 2000, foi publicada a nova série de normas ISO 9000, após mais de quatro anos de discussões. Usuários de todas as partes do mundo, ouvidos através de uma pesquisa conduzida pela própria ISO, foram muito críticos em relação à ISO 9001 edição de 1994, classificando-a como complexa e um pouco confusa no seu entendimento. Esses aspectos foram detalhadamente analisados, em conjunto com as sugestões dos usuários que queriam uma norma voltada para os processos da organização, para seus clientes e para a melhoria contínua do desempenho do Sistema de Gestão da Qualidade

A ISO 9001 sugere que a aplicação e a gestão de um sistema de processos seja uma forma efetiva de garantir uma boa gestão da qualidade [Maldonado,2001]. Para adotar esta “abordagem de processo”, a norma inclui uma metodologia PDCA (“Planejar-Fazer-Checkar-Agir”) que pode ser aplicada a todos os processos, segue a descrição:

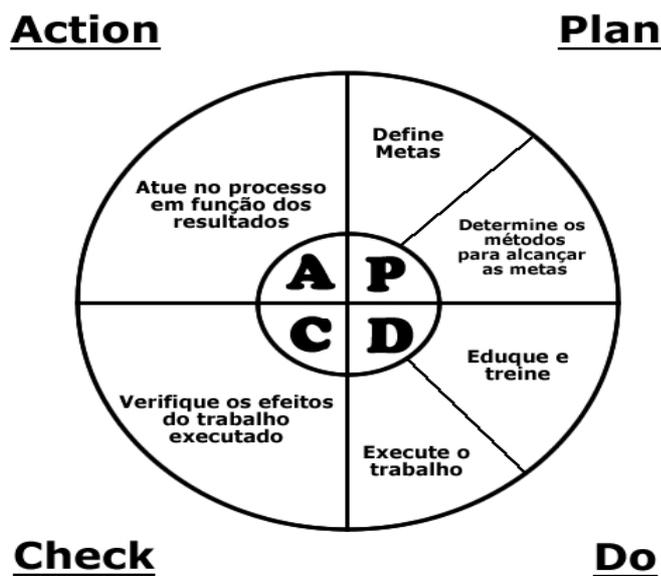


Figura 5 - Ciclo PDCA [Campos, 1992]

- **Planejar:** Estabelecer os objetivos e processos necessários para gerar resultados, de acordo com os requisitos dos clientes e com as políticas da organização;

- **Fazer:** Implementar os processos;
- **Verificar:** Monitorar e medir os processos e produtos em relação às políticas, objetivos e requisitos para o produto e relatar os resultados;
- **Agir:** Tomar ações para melhorar continuamente o desempenho dos processos.

A norma indica que a melhor forma de atender aos requisitos é a criação de procedimentos documentados. Os procedimentos são uma forma especificada para desenvolver uma atividade. Ela propõe um caminho com seis procedimentos:

1. **Controle de documentos:** Os documentos requeridos pelo sistema de gestão da qualidade devem ser controlados, A documentação é essencial para a retenção do know how na organização e não nas mãos dos seus colaboradores, permitindo assim a perpetuação da atividade na organização. Também é importante para facilitar a padronização das atividades e o treinamento de seus colaboradores;
2. **Controle de registros:** Os registros devem ser mantidos legíveis, prontamente identificáveis e recuperáveis. Um procedimento documentado deve ser estabelecido para definir os controles necessários para identificação, armazenamento, proteção, recuperação, tempo de retenção e descarte dos registros;
3. **Auditoria interna:** A Auditoria Interna pode ser definida como sendo uma atividade formal e documentada, executada por pessoas que não tenham responsabilidade direta com a área auditada. Essas pessoas devidamente habilitadas e treinadas, utilizam-se de métodos de coleta para análise de informações.
4. **Controle da não conformidade de produtos:** Preocupa-se em controlar e identificar o número de não conformidades de um produto de acordo com as especificações estabelecidas. É um processo relacionado à pré-auditoria;
5. **Ação corretiva:** é uma ação tomada para eliminar a causa de uma não-conformidade ou outra situação indesejável;
6. **Ação preventiva:** é executada para prevenir a ocorrência de uma não-conformidade.

A existência de procedimentos, instruções e registros de trabalho formalizam todas as atividades que afetam a qualidade. Isto exige a participação de todos os indivíduos da organização. Portanto, o comprometimento com a qualidade aumenta, uma vez que todos participam diretamente da implementação do sistema da qualidade e são os principais responsáveis pelas atividades da empresa.

O cliente influencia o início, o próprio processo e o produto final. O domínio de um procedimento de desenvolvimento de software e do gerenciamento leva ao domínio do processo de produção, o que ajuda a garantir a qualidade do produto final. O sistema de gestão de qualidade é o apoio para a conformidade dessa norma e deve estar estruturado para controlar e divulgar o conjunto dos procedimentos usados pela empresa, facilitar e promover o gerenciamento de mudanças, além de facilitar as atividades de monitoração e de auditorias do sistema da qualidade.

A ISO 9001:2000 está dividida em quatro requisitos: Responsabilidade da administração, Gerenciamento de recursos, Produção e Medição, Análise e Melhoria. [Maldonado, 2001].

3.5.1 Responsabilidade da Administração

A alta direção da organização, com a nova ISO 9001, deverá ter um papel bem mais ativo em relação ao Sistema de Gestão da Qualidade (SGQ). É requerido agora que a alta administração:

- Forneça evidências de seu comprometimento com o desenvolvimento, a implementação e a melhoria contínua da eficácia do SGQ;
- Assegure que os requisitos do cliente são determinados e cumpridos;
- Estabeleça a política da qualidade e garanta que essa política fornece uma estrutura para definir e analisar criticamente os objetivos da qualidade;
- Garanta que são estabelecidos os objetivos da qualidade nas funções e níveis pertinentes da organização, e assegure que tais objetivos são mensuráveis e consistentes com a política da qualidade;
- Assegure que é realizado o planejamento das atividades para o SGQ;

- Assegure que são definidas e comunicadas as responsabilidades, autoridades e suas inter-relações;
- Designe um representante da direção;
- Assegure que são estabelecidos, na organização, processos apropriados de comunicação interna;
- Conduza análises críticas periódicas do SQG e demonstre que são tomadas decisões e ações relacionadas a atividades de melhoria do SGQ, dos processos e dos produtos da organização.

3.5.2 Gerenciamento dos Recursos

Essa seção requer que a organização determine e forneça recursos para implementar, manter e melhorar continuamente a eficácia do Sistema de Gestão da Qualidade. Também é requerido que sejam determinados e fornecidos os recursos necessários para aumentar a satisfação do cliente, atendendo aos seus requisitos. Outros requisitos dessa seção incluem as seguintes condições:

- Executar atividades que afetam a qualidade do produto com pessoal competente, com base em ações de educação, treinamento, habilidades e experiências apropriados, bem como avaliar a eficácia das respectivas ações tomadas;
- Assegurar que as pessoas estão conscientes da relevância e importância de suas atividades e de como estas contribuem para que os objetivos da qualidade sejam alcançados;
- Manter registros da educação, treinamento, habilidades e experiência do pessoal;
- Identificar, fornecer e manter a infra-estrutura (instalações) necessária para alcançar conformidade com os requisitos do produto, incluindo serviços de apoio como transporte, comunicação e manutenção;
- Determinar e gerenciar os fatores do ambiente de trabalho necessários para alcançar a conformidade do produto.

3.5.3 Realização do Produto

Os principais requisitos dessa seção incluem os elementos a seguir enumerados, devendo o usuário da norma sempre lembrar que, conforme definido em toda a série ISO 9000, o termo "produto" pode também significar "serviço":

- Planejamento e desenvolvimento dos processos necessários para a realização do produto;
- Análise crítica de contrato e comunicação com o cliente, nos processos relacionados ao cliente;
- Requisitos para projeto e desenvolvimento do produto, incluindo o controle de alterações desses elementos;
- Requisitos para aquisição;
- Produção e serviços associados (controle de processos, incluindo processos especiais);
- Identificação e rastreabilidade de requisitos;
- Controle da propriedade do cliente;
- Preservação do produto;
- Controle de dispositivos de monitoramento e medição.

3.5.4 Medição, Análise e Melhoria

Nessa seção, os requisitos para auditoria interna foram ampliados, incluindo agora também a auditoria do SGQ conforme disposições estabelecidas pela própria organização. Outros elementos dessa seção abrangem:

- Planejamento e implementação de processos de monitoramento, medição, análise e melhoria contínua;
- Monitoramento de informações relacionadas ao cliente, como uma das medidas de desempenho do SGQ;
- Condução de auditorias internas;
- Monitoramento e medição de processos;

- Monitoramento e medição do produto;
- Controle de produto não-conforme;
- Análise de dados;
- Melhoria contínua da eficácia do SGQ;
- Ação corretiva;
- Ação preventiva.

Para uma empresa buscar sua certificação segundo a norma ISO 9001 deve conhecer e mostrar que entende os requisitos do cliente, planejar e documentar as atividades que afetam a qualidade; qualificar pessoas, identificar e disponibilizar recursos para manter o sistema da qualidade, registrar as atividades, prevenir não-conformidades, registrar a ocorrência delas e tratar as causas dessas não-conformidades, além de manter um programa de avaliação para a melhoria contínua de todos os seus processos.

3.6 Modelo de Qualidade CMM - Capability Maturity Model

Conhecido como "Modelo de Maturidade da Capacidade", o CMM é uma iniciativa do SEI (Software Engineering Institute) para avaliar e melhorar a capacitação de empresas que desenvolvem software [EDS Quality Plan, 2004]. O projeto CMM foi apoiado pelo Departamento de Defesa do Governo dos Estados Unidos, que é um grande consumidor de software e precisava de um modelo formal que permitisse selecionar os seus fornecedores de software de forma adequada.

Embora não seja uma norma emitida por uma instituição internacional (como a ISO ou o IEEE), esta norma tem tido uma grande aceitação mundial, até mesmo fora do mercado americano. O modelo, publicado em 1992, pode ser obtido na própria Internet com facilidade. O CMM também é chamado de SW-CMM (Software CMM).

3.6.1 Maturidade

O CMM é um modelo para medição da maturidade de uma organização no que diz respeito ao processo de desenvolvimento de software. A definição do que é "Maturidade" pode ser melhor entendida através da análise da **Figura 6**:

<u>Organizações Maduras</u>	<u>Organizações Imaturas</u>
Papéis e responsabilidades bem definidos.	Processo improvisado.
Existe base histórica.	Não existe base histórica.
É possível julgar a qualidade do produto.	Não há maneira objetiva de julgar a qualidade do produto.
A qualidade dos produtos e processos é monitorada.	Qualidade e funcionalidade do produto sacrificados.
O processo pode ser atualizado.	Não há rigor no processo a ser seguido.
Existe comunicação entre o gerente e seu grupo	
Resolução de crises imediatas.	

Figura 6 - Organizações Maduras X Organizações Imaturas.

3.6.2 Níveis

O CMM classifica as organizações em cinco níveis distintos, cada um com suas características próprias. No nível 1, estão as organizações mais imaturas, onde não há nenhuma metodologia implementada e tudo ocorre de forma desorganizada. Por outro lado, no nível 5 estão as organizações mais maduras, onde cada detalhe do processo de desenvolvimento está definido, quantificado e acompanhado e a organização consegue até absorver mudanças no processo sem prejudicar o seu desenvolvimento.

3.6.3 Descrição dos Níveis CMM

Nível 1 – Inicial. O processo de desenvolvimento é desorganizado e caótico. Poucos processos são definidos e o sucesso depende de esforços individuais e heróicos. O processo de software de uma organização de Nível 1 é imprevisível porque é constantemente alterado ou modificado à medida que o trabalho progride. Os cronogramas, os orçamentos, as funcionalidades e a qualidade do produto são geralmente imprevisíveis. O desempenho depende da capacidade dos indivíduos e varia com as suas habilidades, conhecimentos e motivações inatas. Existem poucos processos de software estáveis em evidência e o desempenho só pode ser previsto através da habilidade individual ao invés da capacidade da organização;

Nível 2 - Repetível. Os processos básicos de gerenciamento de projeto estão estabelecidos e permitem acompanhar custo, cronograma e funcionalidade. Os projetos nas organizações de nível 2 têm instalado controles básicos de gestão de software. Os compromissos do projeto são baseados em resultados observados em projetos anteriores e nos requisitos do projeto atual. Os gerentes do projeto acompanham os custos, os cronogramas e as funcionalidades do software. Os problemas com compromissos são identificados quando surgem. Os requisitos de software e os produtos de trabalho desenvolvidos para satisfazê-los são congelados e a integridade dos mesmos é controlada. Os padrões do projeto de software são definidos e a organização garante que eles sejam seguidos fielmente. O projeto de software trabalha com os seus subcontratados, se existirem, para estabelecer uma forte relação cliente-fornecedor;

Nível 3 - Definido. Tanto as atividades de gerenciamento quanto de engenharia do processo de desenvolvimento de software estão documentadas, padronizadas e integradas em um padrão de desenvolvimento da organização. Todos os projetos utilizam uma versão aprovada e adaptada do processo padrão de desenvolvimento de software da organização. A capacidade do processo de software das organizações de nível 3 pode ser resumida como sendo padrão e consistente porque a engenharia de software e as atividades de gestão são estáveis e repetíveis. Dentro de linhas estabelecidas de produtos, o custo, o cronograma e as funcionalidades estão sob controle e a qualidade de software é acompanhada.

Nível 4 - Gerenciado. São coletadas medidas detalhadas da qualidade do produto e processo de desenvolvimento de software. Tanto o produto quanto o processo de desenvolvimento de software são entendidos e controlados quantitativamente. A capacidade do processo de software das organizações de nível 4 pode ser resumida como sendo previsível porque o processo é medido e opera dentro de limites mensuráveis. O processo desse nível permite que a organização antecipe as tendências na qualidade dos produtos e dos processos dentro das fronteiras quantitativas desses limites. Quando esses limites são excedidos, são tomadas ações para corrigir a situação. Os produtos de software são, como era de se esperar, de alta qualidade;

Nível 5 - Otimizado. O nível de maturidade 5 se concentra no melhoramento contínuo do desempenho de processos através de melhorias tecnológicas incrementais e inovadoras. Os objetivos quantitativos de melhoria de processos para a organização são estabelecidos, continuamente revisados para refletir alterações nos objetivos do negócio. O melhoramento contínuo é conseguido através de um "feedback" dos processos e pelo uso pioneiro de tecnologias inovadoras.

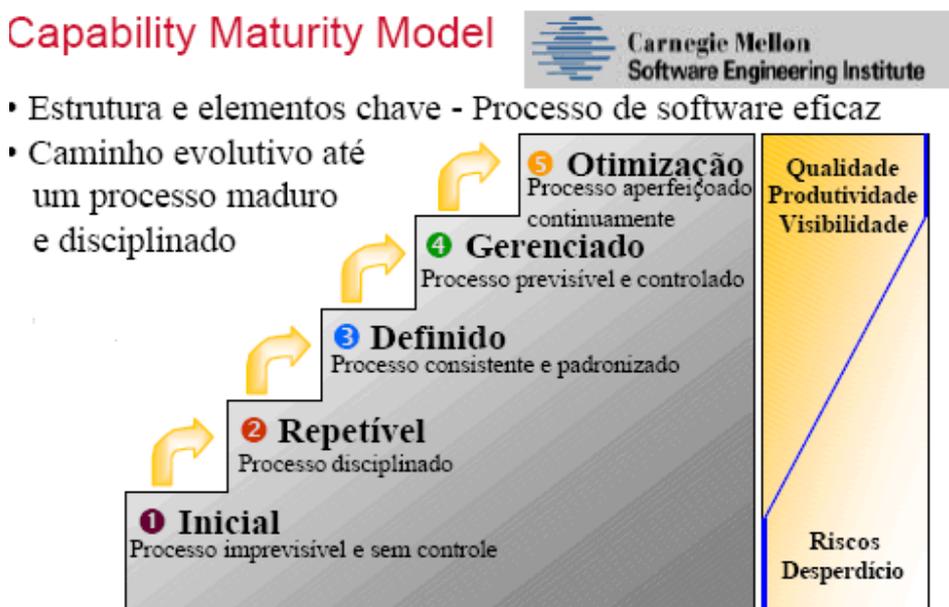
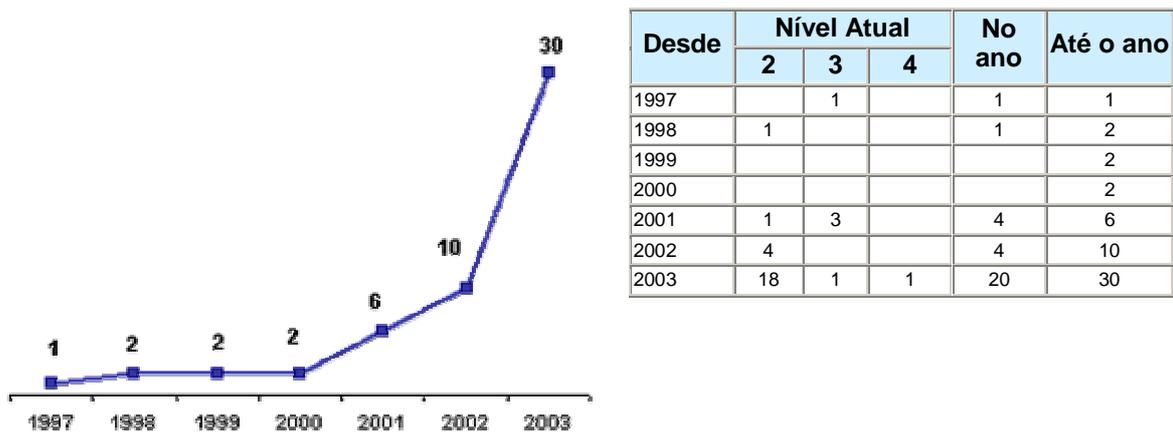


Figura 7 - Níveis de Maturidade [SEI, Software Engineering Institute] .

3.6.4 Entendendo os níveis de maturidade

O modelo CMM descreve atributos essenciais (ou chaves) que seriam esperados para caracterizar uma organização em um nível de maturidade [Maldonado,2001]. É um modelo normativo, no sentido de que as práticas detalhadas caracterizam os tipos normais de comportamento que seriam esperados em uma organização que desenvolve projetos em larga escala em um contexto de contratação governamental. A intenção é que o CMM tenha um nível suficiente de abstração que não restrinja desnecessariamente a maneira que o processo de software é implementado pela organização: ele simplesmente descreve o que normalmente seria esperado que os atributos essenciais do processo de software fossem.

Organizações com Qualificação CMM no Brasil – 1997-2003



Fontes: ISD Brasil, Procesix e imprensa, compilado por MCT/SEITEC/DSI.

Organizações brasileiras avaliadas de maneira oficial pelo SEI¹, ISD² ou Procesix³

Seg	Organizações	UF	Data de Avaliação	Fonte	Data de Referência
Nível 2					
1	Alstom Transportes	SP	jun/02	ISD	nov/03
2	Atech Tecnologias Críticas	SP	set/03	ISD	nov/03
3	C.E.S.A.R	PE	jun/03	ISD	nov/03

4	Citibank		jun/03	site Disoft Revista Forbes Brasil	fev/04 jun/03
5	CPqD	SP	mai/03	ISD	nov/03
6	Credicard	SP	1998	ISD	nov/03
7	DBA Engenharia de Sistemas	RJ	jul/01	ISD	nov/03
8	Dell	RS	fev/03	ISD	nov/03
9	Disoft	SP	abr/03	Procesix	fev/04
10	DTS	SP	dez/03	Procesix	fev/04
11	e-Dablio	RJ	abr/03	ISD	nov/03
12	G&P - Gennari & Peartree	SP	nov/03	ISD	nov/03
13	HP Brasil	RS	set/03	ISD	nov/03
14	Inatel	MG	fev/03	ISD	nov/03
15	Instituto Atlântico	CE	out/03	ISD	nov/03
16	Nec do Brasil	SP	abr/03	ISD	nov/03
17	Politec Brasília	DF	jan/03	ISD	nov/03
18	Politec São Paulo	SP	nov/03	ISD	nov/03
19	SERPRO Brasília	DF	dez/03	Procesix	fev/04
20	SERPRO Fortaleza	CE	out/03	Procesix	fev/04
21	SERPRO Recife	PE	nov/02	ISD	nov/03
22	SERPRO Salvador	BA	nov/03	Procesix	fev/04
23	Stefanini	SP	out/02	ISD	nov/03
24	Tele Design	SP	set/02	ISD	nov/03
Nível 3					
1	Ci&T Software S.A,	SP	mar/04	SEI	abr/04
2	EDS São Paulo Solution Center	SP	ago/01	Gazeta Mercantil	fev/04
3	Ericson do Brasil	SP	mar/01	Gazeta Mercantil	fev/04
4	IBM Fábrica de Software (CMMI)	RJ/SP	jan/03	ISD	nov/03
5	Motorola	SP	dez/01	ISD	nov/03
6	Xerox do Brasil – CDSV (desativada em Janeiro/2005)	ES	dez/97	ISD	nov/03
Nível 4					
1	EDS Rio de Janeiro Solution Center	RJ	dez/03	Gazeta Mercantil	fev/04
Nível 5					
1	Tata Consultancy Services do Brasil S/A	DF	jun/04	Tata Consultancy	jun/04

¹ Software Engineering Institute / Carnegie Mellon University

² Integrated System Diagnostics Brasil

³ Representada pela JDFurlan & Associados

Figura 8 – Organizações certificadas

[<http://www.mct.gov.br/Temas/info/Dsi/qualidad/CMM.htm>]

Em qualquer contexto em que o modelo seja aplicado, deve ser utilizada uma interpretação razoável das práticas. O CMM deve ser interpretado apropriadamente, utilizando-se o conhecimento de peritos quando o ambiente de negócio da organização difere significativamente de uma grande organização fornecedora. O CMM não é prescritivo; ele não diz à organização como melhorar. O modelo descreve a organização em cada nível de maturidade sem prescrever os meios específicos para consegui-lo.

A melhoria do processo de software ocorre dentro do contexto dos planos estratégicos e dos objetivos de negócio da organização, da sua estrutura organizacional, das tecnologias em uso, da sua cultura social e do seu sistema de gestão. O CMM tem o seu foco nos aspectos do processo da Gestão da Qualidade Total.

Exceto no nível 1, todos os níveis são detalhados em áreas-chave de processo. Estas áreas são exatamente aquilo no que a organização deve focar para melhorar o seu processo de desenvolvimento de software. Para que uma empresa possa se qualificar em um determinado nível de maturidade ela deve realizar os processos relacionados às áreas-chave daquele determinado nível.

3.7 - CMMI

A experiência no uso do CMM durante uma década serviu para identificar pontos em que o modelo poderia ser melhorado. Ao mesmo tempo, o surgimento do projeto SPICE (Software Process Improvement and Capability Determination), da ISO, levou à necessidade de compatibilização do CMM com a futura norma ISO 15504, que será o resultado do projeto. Por estas razões, o SEI iniciou um projeto chamado CMMI (CMM Integration).

O CMMI é um modelo para definir e melhorar a capacidade e a maturidade dos processos das organizações que produzem software. As empresas que investem no CMMI têm como principais benefícios o aumento da qualidade dos processos de software e o reconhecimento internacional dessa qualidade. Também têm uma maior e mais efetiva

gerência de projetos de software, no que diz respeito a custo, tempo e recursos utilizados. Esse modelo possibilita uma redução significativa de defeitos nos serviços gerados e maior qualificação do pessoal no atendimento ao cliente, além da customização dos processos de acordo com as necessidades. Com isso, as empresas reduzem o retrabalho, baixam seus custos e agilizam suas soluções, agregando valor para o cliente.

Os objetivos do CMMI são, basicamente, voltados para a redução do custo da implementação de melhoria de processo. São eles [Ahern, 2001] :

- Eliminação de inconsistências e redução de duplicidades;
- Melhoria da clareza e entendimento;
- Utilização de terminologia comum e estilo consistente;
- Estabelecimento de regras de construção uniformes;
- Manutenção de componentes comuns;
- Consistência com a futura norma ISO/IEC 15504;
- Sensibilidade às implicações dos esforços legados.

A busca pela certificação CMMI representa a afirmação da filosofia de aprimoramento contínuo dos produtos e a concretização de um importante passo rumo à exportação de software.

A principal mudança do CMMI em relação ao CMM é a possibilidade de utilização de duas diferentes abordagens para a melhoria de processos. Estas duas abordagens são conhecidas como o “representação contínua” e o “representação em estágios”. Existem muitas razões para uma empresa selecionar uma representação ou outra. Pode ser que uma organização escolha utilizar a representação que lhe seja mais familiar. As listas seguintes descrevem algumas das possíveis vantagens e desvantagens de selecionar cada uma das representações:

- Representação Contínua: Permite que se selecione a seqüência de melhorias que melhor atende aos objetivos de negócios e reduz as áreas de risco da sua

organização. Possibilita comparações dentro e entre organizações em uma área de processo;

- Representação em Estágios: Oferece uma seqüência comprovada de melhorias, começando com práticas básicas de gerenciamento e progredindo por um caminho pré-definido e comprovado de níveis sucessivos, cada um servindo como base para o próximo.

Atualmente existem três áreas de conhecimento disponíveis, quando for selecionar um modelo CMMI: Engenharia de Sistemas, Engenharia de Software e Desenvolvimento Integrado de Produtos e Processos (Integrated Product and Process Development – IPPD)

- Engenharia de Sistemas: A engenharia de sistemas cobre o desenvolvimento de sistemas completos, que podem ou não incluir software. Os engenheiros de sistemas concentram-se em transformar necessidades, expectativas e restrições dos clientes em soluções de produtos e fornecer suporte a estas soluções de produtos durante toda a vida do produto.
- Engenharia de Software: A engenharia de software cobre o desenvolvimento de sistemas de software. Engenheiros de software se concentram em aplicar abordagens sistemáticas, disciplinadas e quantificáveis ao desenvolvimento, operação e manutenção de software.
- Desenvolvimento Integrado de Produtos e Processos: O Desenvolvimento Integrado de Produtos e Processos (Integrated Product and Process Development – IPPD) é uma abordagem sistemática que organiza uma colaboração pontual de patrocinadores relevantes durante toda a vida do produto para melhor satisfazer as necessidades, expectativas e requisitos do cliente. Os processos que oferecem suporte a uma abordagem IPPD são integrados com os outros processos da organização. As áreas de processos, metas e práticas específicas do IPPD isoladas não conseguem criar um desenvolvimento integrado de produtos e processos. Se um projeto ou organização escolhe utilizar o IPPD, tem que executar as práticas específicas do IPPD juntamente com as outras práticas específicas utilizadas para produzir os produtos (por exemplo, as áreas de processos de Engenharia). Isto é, se

uma organização ou projeto deseja utilizar o IPPD, ela escolhe um modelo com uma ou mais disciplinas além do IPPD.

O projeto CMMI envolveu uma grande quantidade de pessoas de diferentes organizações do mundo todo. Estas organizações utilizavam um modelo CMM ou múltiplos CMMs e estavam interessadas nos benefícios do desenvolvimento de um framework integrado para auxiliar a melhoria de processos no âmbito do empreendimento como um todo.

O trabalho do projeto CMMI é patrocinado pelo Departamento de Defesa dos Estados Unidos (Department of Defense – DoD), especificamente pelo departamento da Sub-Secretaria de Defesa, Aquisição, Tecnologia e Logística (Office of the Under Secretary of Defense, Acquisition, Technology, and Logistics - OUSD/AT&L). O patrocínio da indústria é garantido pelo Comitê de Engenharia de Sistemas da Associação Industrial da Defesa Nacional (National Defense Industrial Association - NDIA).

O produto de software passa a ser , cada vez mais, um componente comum em uma série de outros produtos, desde carros, eletrodomésticos, elevadores, telefones, até sistemas de informação organizacionais. De um produto exige-se qualidade e preço. Portanto, como produto, o software deve ter o nível de qualidade exigido e procurar ser desenvolvido com o menor custo possível.

Produzir software de qualidade é uma meta básica da engenharia de software, que disponibiliza métodos, técnicas e ferramentas para esse fim. Muito tem sido escrito sobre qualidade e seus vários adjetivos. No entanto, o fundamental é que o software seja confiável, ou seja, que seja eficaz e siga os padrões exigidos pelo contexto onde irá atuar. Cada vez mais a sociedade pressiona o setor de software para que a característica “qualidade” seja preponderante. Normas internacionais como a ISO e iniciativas como as do SEI (Software Engineering Institute) são exemplos disso.

É importante não esquecer que a produção de software é um processo que envolve, como parte fundamental, seres humanos. As tecnologias de produção de software, tem por objetivo automatizar ao máximo a produção, mas ainda são fundamentalmente dependentes da qualidade das equipes, das pessoas, de todo o time envolvido num processo de desenvolvimento.

4 - Conclusão

De acordo com os dados pesquisados, verificamos que para um processo funcionar adequadamente é preciso haver treinamento. Qualquer mudança ou informações adicionais inseridas nos processos devem ser comunicadas, disponibilizando materiais de consulta, apresentações, tutoriais a fim de melhorar o aprimoramento profissional das pessoas envolvidas.

Um outro elemento que consideramos fundamental é a documentação, onde serão criadas bases necessárias para uma melhor utilização e manutenção do produto de software. Percebemos que pouca atenção tem sido dada à documentação gerada durante os projetos de desenvolvimento, que tem resultado em documentos mal-elaborados, de difícil compreensão, inadequados e incompletos.

Tendo em vista os aspectos relacionados, percebemos que um processo utilizado de forma correta gera um produto final muito mais eficiente. Durante o processo de ciclo de vida do desenvolvimento verificamos que há um maior controle de todas as informações relacionadas ao projeto, custos, recursos, tempo, riscos etc.

Atualmente existe uma série de normas dos mais variados tipos e com aplicabilidade em diversas áreas. A nossa escolha tem como base o conhecimento pré-adquirido no curso de graduação, na nossa experiência profissional, visto que grande parte destas práticas, são exercidas diariamente no nosso ambiente de trabalho.

Procuramos nesta pesquisa sinalizar também a importância e a necessidade da implantação de normas e modelos de qualidade nas empresas de TI. De fato, uma certificação traz uma série de benefícios para as organizações, observamos que muitas vezes, as empresas certificadas levam uma série de vantagens nas licitações para desenvolvimento de sistemas. Sendo assim, o selo de qualidade faz bastante diferença no mercado.

Alguns Eventos da área:

- SBQS - Simpósio Brasileiro de Qualidade de Software (antigo WQS - Workshop de Qualidade de Software)
- CITS - "Conferência de Qualidade de Software"
- SIMPROS - Simpósio Internacional de Melhoria de Processo de Software
- PBQP - Projetos e Workshops do Programa Brasileiro de Qualidade e Produtividade - Sw.
- SPINs Regionais - Reuniões dos grupos de Software Process Improvement Network

Referências Bibliográficas

[Ahern, 2001] DENNIS M. AHERN, AARON CLOUSE, e RICHARD TURNER, CMMI Distilled: A Practical Introduction to Integrated Process Improvement, SEI Series in Software Engineering, Addison-Wesley, 306 pages, 2001.

[Campos, 1992] CAMPOS, VICENTE TQC – Controle da Qualidade Total. Belo Horizonte: Bloch Ed, 1992.

[EDS Quality Plan, 2004] Electronic Data System do Brasil, Plano de Qualidade, Rio de Janeiro, 2004.

[ISO. 2000] ISO 9001:2000. Quality Management Systems. Requirements, 2000.

[Maldonado, 2001] – Qualidade de Software , Teoria e prática. São Paulo: Pearson, 2001.

[MCT,1996] MCT . Qualidade no setor de software brasileiro: 1995. Brasília, DF. <http://www.mct.gov.br>

[PMBOK, 2003] - PMBOK - A guide to the Project Management Body of Knowledge. Disciplina de Gerência de Projetos em Informática. Rio de Janeiro: UFRJ,2003.

[Pressman, 2002] PRESSMAN, R. Engenharia de Software. Rio de Janeiro: McGraw-Hill, 2002.

[Weber, 2001] WEBER, K.C.; ROCHA, A.R.C e NASCIMENTO, C.J. Qualidade e produtividade em software, 4ª edição renovada. São Paulo, Makron Books, 2001.

[Zwass, 1998] ZWASS, V. Foundations of Information System. Wisconsin. Irwin McGraw-Hill, jun 1998.