



*Engenharia de Software*

**Tema da Aula**

## Conceitos de Qualidade em Software

*Prof. Cristiano R R Portella*

portella@widesoft.com.br



## Qualidade

Qualidade é um conceito subjetivo, que varia para cada local, época, tipo de produto e pessoa que está avaliando.

Qualidade é:

- Superar as expectativas
- Produto sem defeito
- Fazer melhor com menos recursos
- Adequação ao uso
- Produzido por empresa certificada

**Qualidade é o que cada cliente percebe como sendo “Qualidade”!**

Evolução dos conceitos da Qualidade:

1. Nenhuma Qualidade – Revolução Industrial
2. Inspetores da Qualidade – testar/refugar/retrabalhar
3. Controle Estatístico de Processo (CEP) – qualidade realizada no projeto e no processo  
Shewhart – Deming – Juran – Crosby
4. Melhoria Contínua do Processo
  - Qualidade Total (QTM)
  - Certificação da Qualidade (ISO-QS-MIL-Prêmios da Qualidade etc)
  - Qualidade pela Qualidade

Se um software com muitos erros e falhas perde parte do valor para o usuário, um software isento de erros não é, necessariamente, um software de qualidade para seu usuário (programa certo para o problema errado).

As necessidades do usuário são expressas nos requisitos explícitos, mas também nos requisitos implícitos, que normalmente não são declarados, como: flexível, fácil de operar, barato, construído no prazo etc (resumindo: mais prático, mais rápido, mais fácil, mais seguro e mais barato em relação ao processo convencional – antes da informatização).

A ausência de funções necessárias (requisitos declarados), não pode ser compensada por funções auxiliares genéricas não solicitadas, como: calculadora, agenda, cor de tela etc.

*“Um software de Qualidade deve encantar o cliente e não somente funcionar direito e não ter erros”.*

*Bill Gates*

*“Software de Qualidade é aquele que, não apenas satisfaz as exigências, mas também é implementado a tempo e de acordo com o orçamento”.*

*Juran*

### Baixa Qualidade em Software:

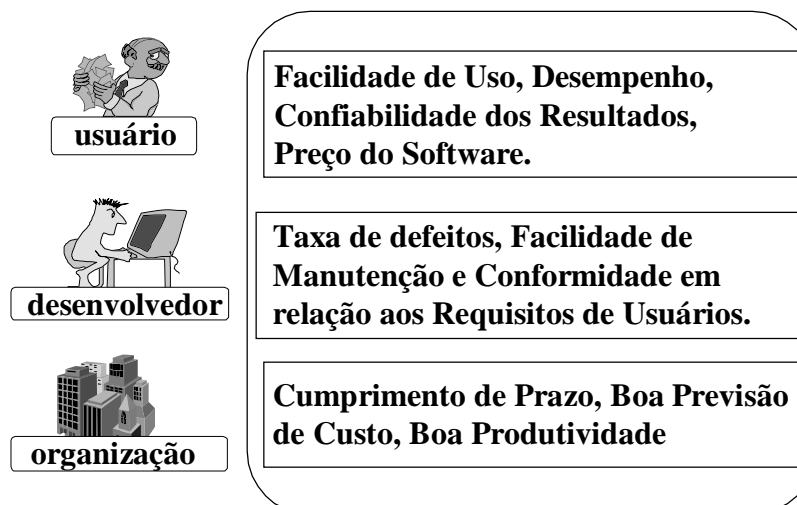
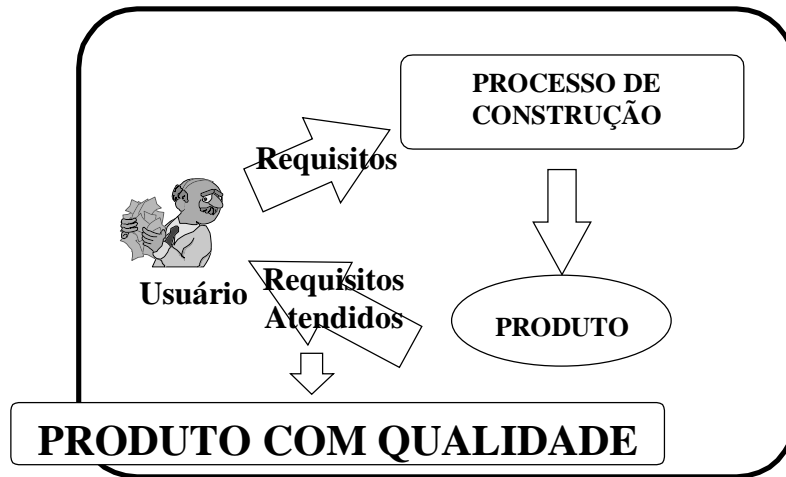
1. Produzir software é um processo de medição difícil (geralmente medição indireta);
2. Metodologia e Técnicas de Engenharia de Software ainda são desconhecidas ou relegadas pela grande maioria dos desenvolvedores de software; e
3. A maioria das técnicas de testagem têm eficiência menor que 30% na detecção de defeitos (Capers 91).

A melhoria da Qualidade em Software requer:

- ✓ Projeto realizado dentro de um rigor científico, buscando dar qualidade ao produto de software
- ✓ Controle do processo de desenvolvimento
- ✓ Medição do processo de desenvolvimento
- ✓ Técnicas de Garantia da Qualidade (SQA)
- ✓ Atividades auxiliares independentes de fase (gerenciamento de configuração, técnicas de melhoria contínua)
- ✓ Utilização de Métodos, Padrões e Ferramentas adequadas.

### **Qualidade em Software:**

Conformidade a requisitos funcionais e não-funcionais explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de qualquer software desenvolvido profissionalmente.



A Eng. Software preconiza que a Qualidade em Software pode ser determinada através de duas fontes:

**Propriedades ou Fatores da Qualidade:**

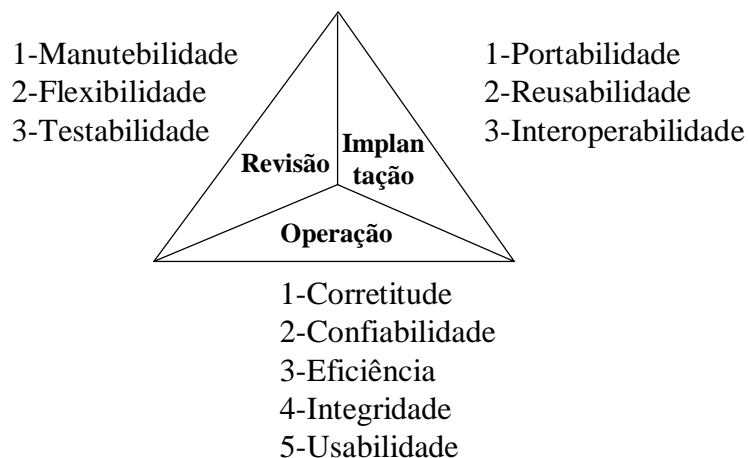
São características de alto nível, cuja medição só pode ser feita de maneira indireta, através do grau de existência dos "Atributos". Segundo McCall e Cavano (78), existem 11 Propriedades.

**Atributos:**

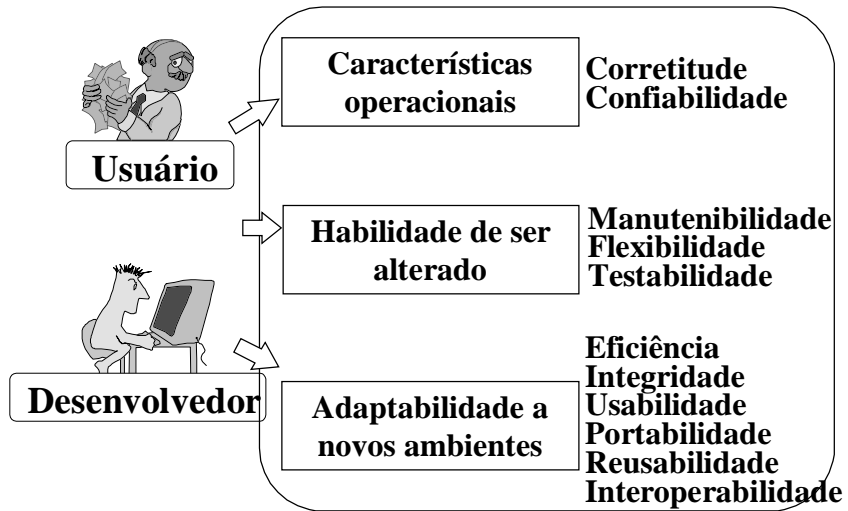
São características de baixo nível, que podem ser medidos diretamente no produto de software. Segundo McCall e Cavano elas são em número de 22.

Vide Tabela completa de Atributos x Fatores em Pressman Cap.19 na 5a Edição ou Cap 17 na 3a Edição.

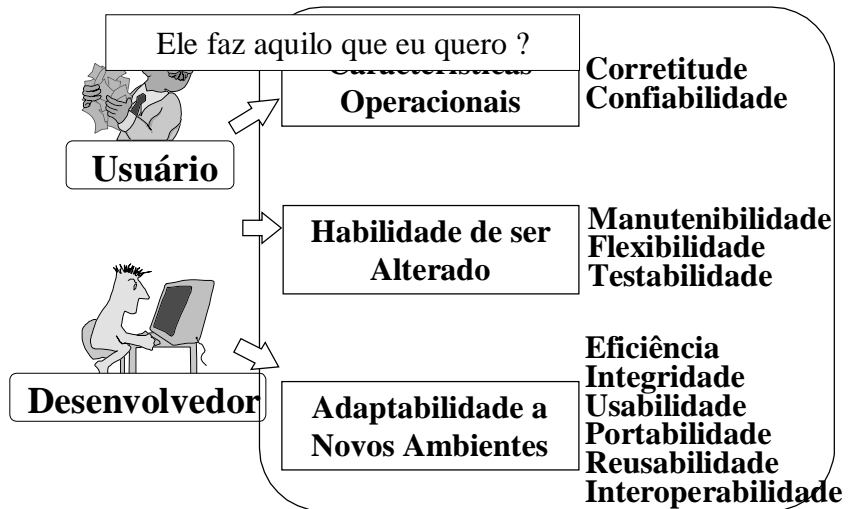
Classificação de McCall e Cavano para os Fatores da Qualidade:



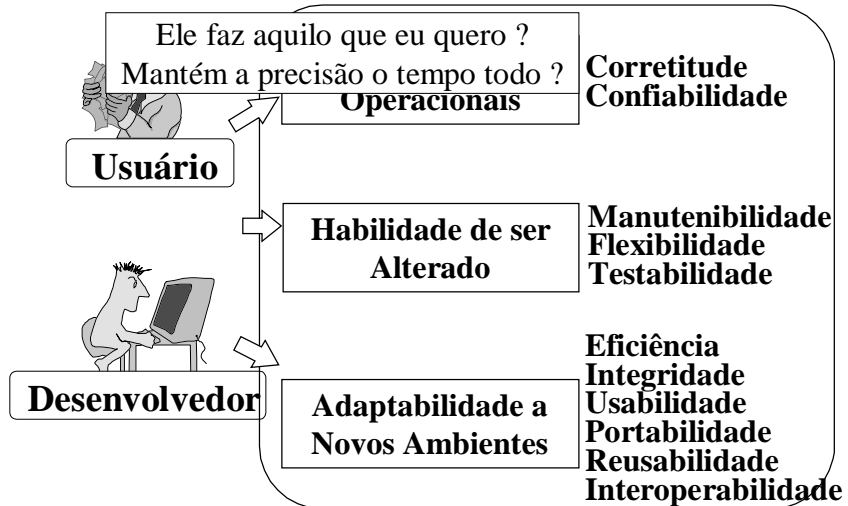
## Qualidade no Produto de Software



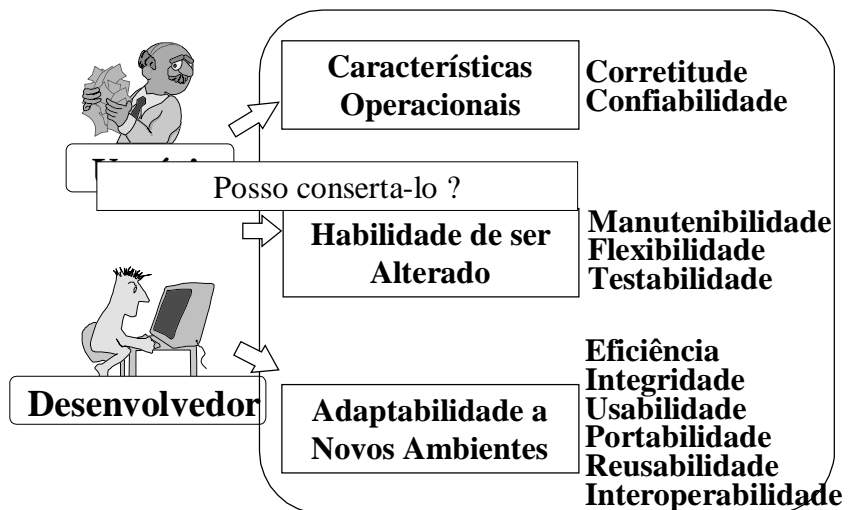
## Qualidade no Produto de Software



## Qualidade no Produto de Software

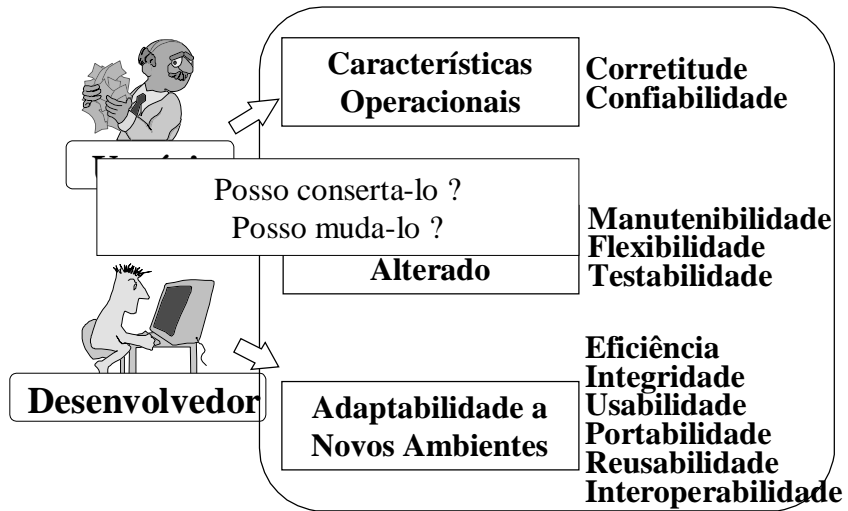


## Qualidade no Produto de Software

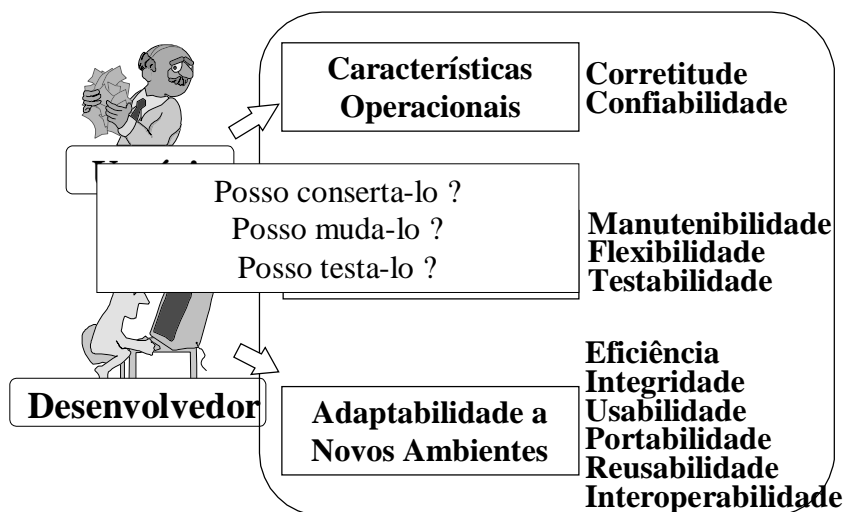




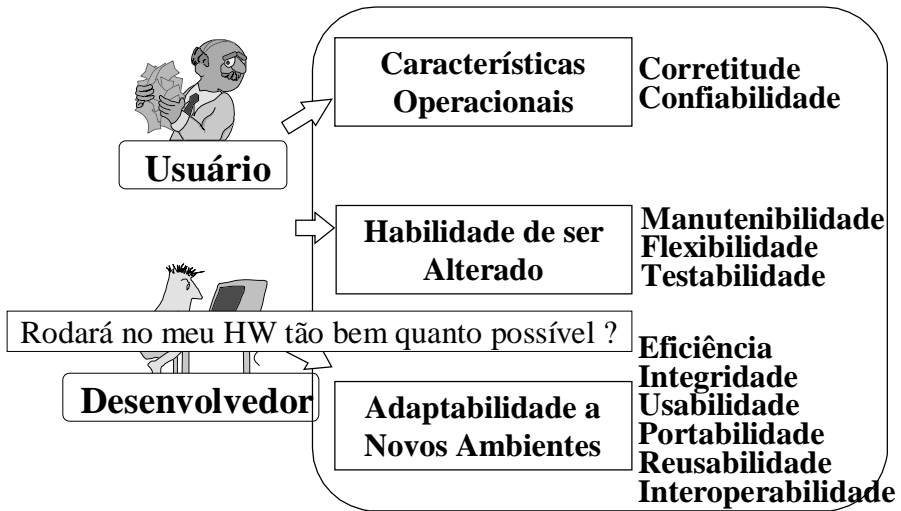
## Qualidade no Produto de Software



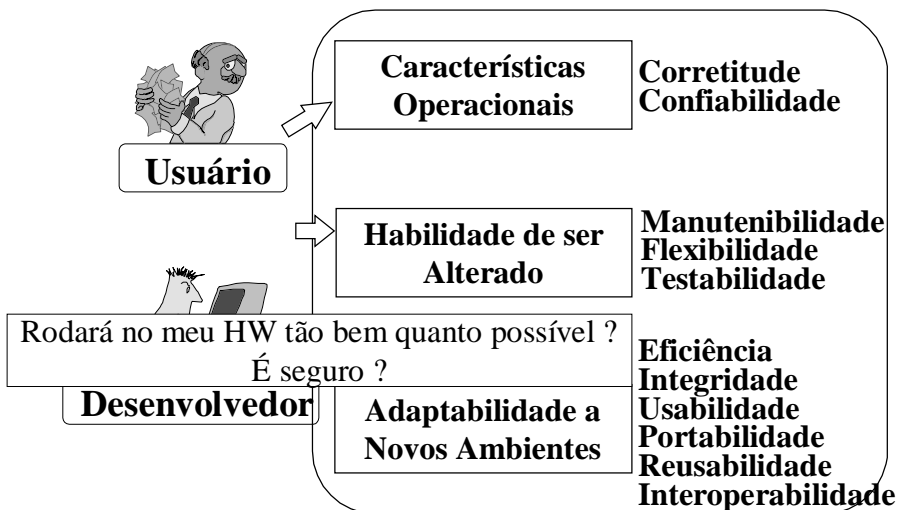
## Qualidade no Produto de Software



## Qualidade no Produto de Software



## Qualidade no Produto de Software



## Qualidade no Produto de Software



**Características Operacionais**

**Corretitude  
 Confiabilidade**

**Habilidade de ser Alterado**

**Manutenibilidade  
 Flexibilidade  
 Testabilidade**

Rodará no meu HW tão bem quanto possível ?  
 É seguro ?  
 É fácil de usar ?

**Eficiência  
 Integridade  
 Usabilidade  
 Portabilidade  
 Reusabilidade  
 Interoperabilidade**

**Novos Ambientes**

## Qualidade no Produto de Software



**Características Operacionais**

**Corretitude  
 Confiabilidade**

**Habilidade de ser Alterado**

**Manutenibilidade  
 Flexibilidade  
 Testabilidade**

Rodará no meu HW tão bem quanto possível ?  
 É seguro ?  
 É fácil de usar ?  
 Poderei usa-lo em outra máquina ?

**Eficiência  
 Integridade  
 Usabilidade  
 Portabilidade  
 Reusabilidade  
 Interoperabilidade**

## Qualidade no Produto de Software



**Características Operacionais**

**Corretitude  
Confiabilidade**

**Habilidade de ser Alterado**

**Manutenibilidade  
Flexibilidade  
Testabilidade**

Rodará no meu HW tão bem quanto possível ?  
É seguro ?  
É fácil de usar ?  
Poderei usa-lo em outra máquina ?  
Poderei utilizar parte dele em outro produto ?

**Eficiência  
Integridade  
Usabilidade  
Portabilidade  
Reusabilidade  
Interoperabilidade**

## Qualidade no Produto de Software



**Características Operacionais**

**Corretitude  
Confiabilidade**

**Habilidade de ser Alterado**

**Manutenibilidade  
Flexibilidade  
Testabilidade**

Rodará no meu HW tão bem quanto possível ?  
É seguro ?  
É fácil de usar ?  
Poderei usa-lo em outra máquina ?  
Poderei reutiliza-lo em outra máquina ?  
Poderei interfacear com outros sistemas ?

**Eficiência  
Integridade  
Usabilidade  
Portabilidade  
Reusabilidade  
Interoperabilidade**

O fator “**Integridade**” será aferido por três atributos relacionados:

1. Facilidade de verificação e instrumentação:

É possível avaliar a precisão das informações resultantes? Essa verificação pode ser automatizada?

2. Segurança, controle e proteção das informações:

Grau de segurança e proteção das informações.  
Controle sobre alterações indevidas.

3. Tolerância a erros:

Grau de detecção de erros e possibilidade de funcionar mesmo sob condições adversas (tolerância a falhas).

Processo de Software:

Consiste em uma série de atividades, práticas, eventos, ferramentas e métodos que garantem, técnica e administrativamente que o software pode ser desenvolvido com qualidade e de maneira organizada, disciplinada e previsível.

Uma das maiores dificuldades encontradas pelas empresas de software é o gerenciamento de seus processos de software.

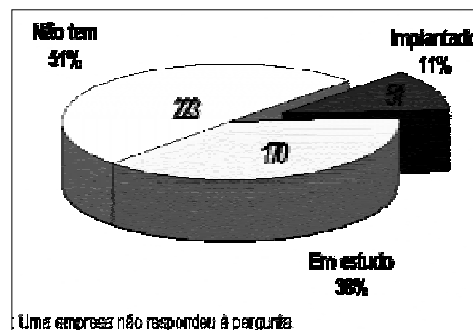
## Situação da Indústria Nacional de Software

MCT-95 entrevista 445 empresas de software quanto a adoção de procedimentos específicos de Garantia da Qualidade em Software.

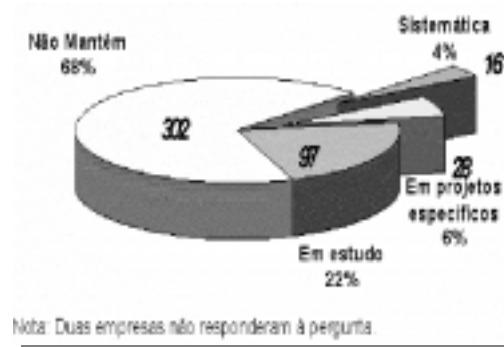
Categorias	No. de Empresas	%
Usam ISO 9126 para auto-avaliação	6	1,4 %
Avaliadas por consultor qualificado	4	0,9 %
Outras formas de auto-avaliação	145	33,2 %
Está se preparando para implantar	157	35,9 %
Não adotam	134	30,7 %

## Situação da Indústria Nacional de Software

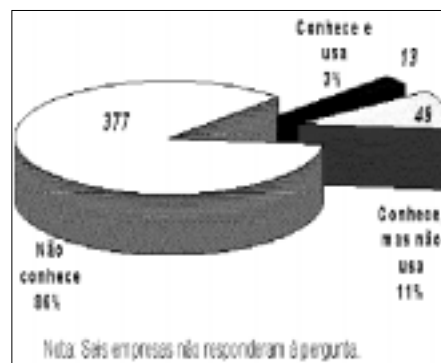
Programa da qualidade total ou similar (MCT-95)



Contabilidade dos custos da qualidade e da não qualidade (MCT-95)



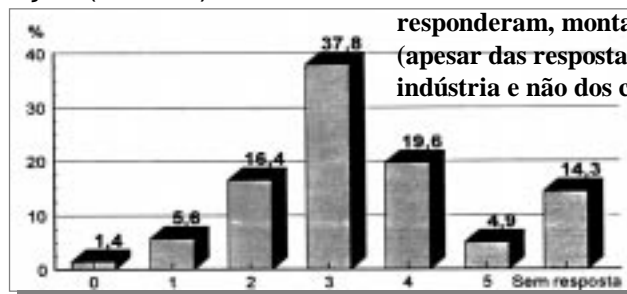
Conhecimento do Modelo CMM - *Capability Maturity Model* - (MCT-95)



## Situação da Indústria Nacional de Software

286 empresas entrevistadas quanto ao grau de satisfação dos clientes com os produtos e serviços (MIS-96).

Soma de 0 a 3 mais os que não responderam, monta 75,5 % (apesar das respostas virem da indústria e não dos clientes.



0 = totalmente insatisfeito

3 = neutro (sem problemas – sem exceder expectativas)

5 = totalmente satisfeito

## Qualidade no Processo de Software

Processo de software não é apenas desenvolvimento. Existe uma série de outras atividades tão importantes quanto a atividade de desenvolvimento.

O SPICE (**S**oftware **P**rocess **I**mprovement and **C**apability **d**etermination) identificou em uma de suas versões, 29 processos dentre os quais apenas 5 ligados ao desenvolvimento de software (análise, desenho, codificação e teste). Os outros revelaram-se mais ligados à parte organizacional da empresa (gerência de configuração, controle de qualidade e infra-estrutura necessários para o desenvolvimento de software, contato com o cliente, estabelecimento de objetivos de negócio, etc).



A área de Informática não se restringe ao desenvolvimento de software (não se pode separar produção de consumo). Ela também oferece outros tipos de serviço como:

- Suporte
- Atendimento
- Treinamento
- Gestão dos recursos de SI/TI
- Prospecção de novas tecnologias.

Esses serviços são avaliados pelos clientes (usuários) a partir de sua rapidez, eficiência, custo, disponibilidade, cortesia etc.

Serviços são itens de difícil verificação segundo padrões da Qualidade, porque:

1. São basicamente intangíveis (e até oferecidos remotamente).
2. São prestados de maneiras diferentes (situações mudam) e por pessoas diferentes.
3. A pessoa que presta o serviço faz parte do serviço.
4. Serviços não são passíveis de inspeção final antes da entrega.
5. Serviços não são estocáveis (disponibilidade).

Como melhorar a qualidade dos serviços da área de Informática ?

1. Ouvir as expectativas dos clientes e definir uma Missão adequada ao Departamento.
2. Medir periodicamente a satisfação dos clientes.
3. Aplicar os Princípios da Qualidade no Departamento, através de Políticas, Procedimentos e medições de Indicadores previamente escolhidos.

Como melhorar a qualidade dos serviços da área de Informática ?

4. Implantar um plano de crescimento de pessoal para a área de Informática (treinamento e valorização).
5. Não abrir mão dos princípios da Qualidade, a partir de justificativas de emergência, urgência, gravidade etc.

Superar expectativas é a melhor forma de aumentar a percepção que o Cliente tem da nossa Qualidade.

Indicadores adequados à Área de Informática:

Quanto ao Produto:

- ✓ Conformidade com os requisitos
- ✓ Ausência de falhas
- ✓ Atendimento aos Atributos (e portanto às Propriedades) da Qualidade em Software, etc

Quanto ao Processo:

- ✓ Quantidade de defeitos detectados
- ✓ Atendimento aos procedimentos técnicos (documentação, padronização etc)
- ✓ Cumprimento de prazos
- ✓ Cumprimento de custos, etc

Itens para um Avaliação de Satisfação do Cliente:

- ✓ Aspectos tangíveis (requisitos, equipamentos etc)
- ✓ Confiabilidade (cumprir o prometido)
- ✓ Presteza (rapidez no cumprimento de prazos)
- ✓ Competência (conhecimentos para o trabalho)
- ✓ Cortesia (educação, respeito)
- ✓ Credibilidade (confiança na honestidade e ética)
- ✓ Acessibilidade (facilidade de contato)
- ✓ Comunicação (manter o cliente informado, em linguagem compreensível; documentação clara)
- ✓ Entender o cliente (esforço para conhecer a área e as necessidades dos clientes)
- ✓ Preço (relação custos x benefícios e relação com mercado).

Itens para um Avaliação de Satisfação do Cliente:

Para cada situação ou área da empresa, o Cliente julga seus serviços segundo alguns critérios que **ele** escolheu como sendo importantes.

Fique atento para identificar o peso que o Cliente dá para cada item. Repasse essa descoberta para a equipe, pedindo que o serviços sejam feitos "a la carte" e não padronizados para um hipotético cliente médio.

Os critérios de avaliação usados pelos Clientes devem ser transformados em Indicadores da Qualidade de serviços.

**Software Quality Assurance-SQA**

A Garantia da Qualidade é um conjunto de atividades que asseguram que todos os esforços serão feitos para garantir que o produto de software tenha a qualidade desejada, a partir de:

- Minimizar o número de defeitos
- Criar mecanismos para controlar o desenvolvimento e manutenção, de forma a preservar prazo e custo
- Garantir que o produto possa ser "usável" pelo mercado
- Melhorar a qualidade de futuras versões do produto

Time independente (time de SQA) para garantir a independência em relação às equipes/chefias de desenvolvimento.

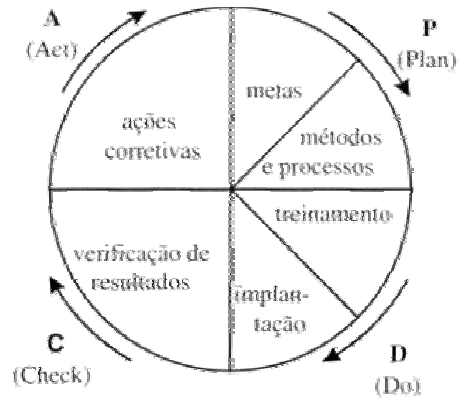
Postura de onbudsman: representante do cliente junto ao desenvolvimento.

Características:

- Rigidez nas avaliações e no cumprimento de padrões
- Traçar planos para casos de não-conformidade
- Independência nas decisões e ações

- ✓ Avaliar e definir ferramentas e técnicas a serem usadas pelo desenvolvimento;
- ✓ Coordenar as atividades de revisões técnicas;
- ✓ Coordenar as atividades de teste;
- ✓ Definir e controlar padrões, normas e procedimentos a serem usados;
- ✓ Auditar o processo;
- ✓ Controlar versões e mudanças no software; e
- ✓ Definir políticas de métricas e registros de medições.

Aplicação do PDCA na Qualidade em Software:



Aplicação dos 5S em Qualidade de Software:

✓ 1-Seiri (Descarte)

- Tenha somente o necessário e na quantidade certa. Livre-se do que for inútil.
  - Backups
  - Documentos
  - Manuais
  - Bibliotecas

Aplicação dos 5S em Qualidade de Software:

✓ 2-Seiton (Arrumação)

- Um lugar certo para cada coisa; cada coisa em seu lugar.
  - Biblioteca (livros e manuais)
  - Mídias antigas
  - Peças e partes de hardware
  - Cadastro de configurações/partes de HW
  - Árvores de diretórios (servidor e estações)
  - Fontes (onde, quem pode acessar, quem acesso...)

Aplicação dos 5S em Qualidade de Software:

✓ 3-Seisoh (Limpeza)

- Mantenha o ambiente de trabalho limpo.
  - Produza código limpo.
  - Destrua informações confidenciais destinadas ao lixo
  - Ao término do expediente, limpe mesas, periféricos etc

Aplicação dos 5S em Qualidade de Software:

✓ 4-Seiketsu (Asseio)

- Mantenha-se mentalmente asseado. Pense, concentre-se.
  - Aja com calma e refletidamente. Não contribua para aumentar o caos.
  - Concentre-se na atividade que estiver desenvolvendo. Limpe sua mente em relação a outros problemas.

Aplicação dos 5S em Qualidade de Software:

✓ 5-Shitsuke (Disciplina)

- Tenha disposição para cumprir as regras.
  - Crie rotinas
  - Descubra a maneira correta de proceder. Aplique essa rotina sempre. Melhore a rotina sempre que possível.
  - Seja metódico nos testes e revisões.
  - Até o trabalho criativo comporta rotinas.
  - Faça os registros sistematicamente.



Também chamado de “Espinha de Peixe” ou Ishikawa, representa o relacionamento entre alguns efeitos e todas as possíveis causas.

Divisão das causas segundo sua origem:

- Máquina
- Material
- Mão-de-Obra
- Método
- Meio Ambiente

