

Qualidade de Software

Referências

Pressman, R.B. *Software Engineering: A Practitioner's Approach*
McGraw-Hill, Third Edition, New-York, EUA

Rocha, A. R.C. and Maldonado, J.C. and Weber, K.C. *Qualidade de Software: Teoria e Prática* Prentice-Hall 2001, SP, Brasil

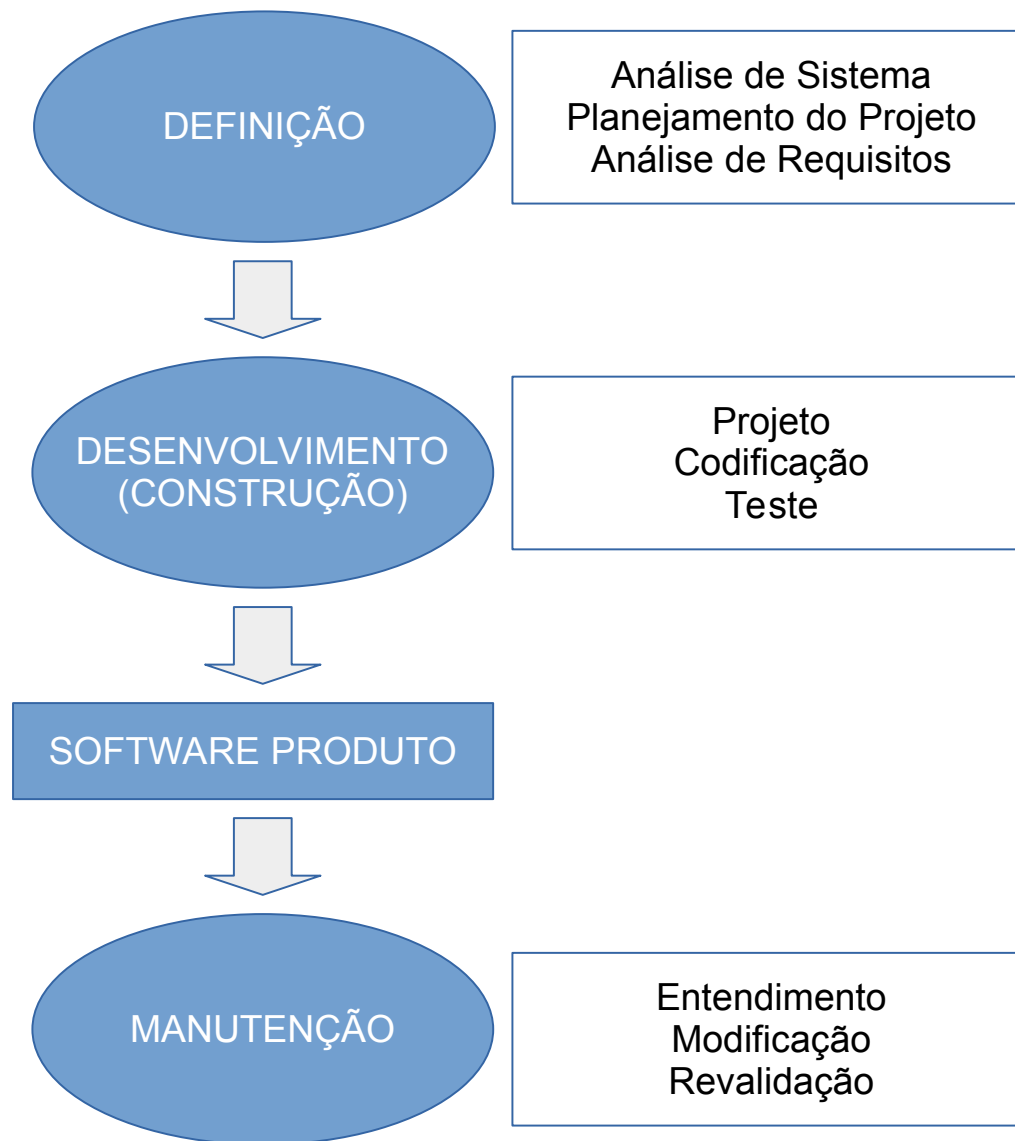
Cortes, M.L. and Chiossi, T.C.S. *Modelos de Qualidade de Software*
Editora da Unicamp 2001, Campinas, SP, Brasil

SEI- Carnegie Mellon University. *The Capability Maturity Model: Guidelines for Improving the Software Process* Addison Wesley - USA

Kan, H.S. *Metrics and Models in Software Quality Engineering*. Addison Wesley, 1995, USA

Rosely Sanches e José Carlos Maldonado – Notas de aula do ICMC – USP São Carlos

Atividades da Engenharia de Software



Atividades de Apoio

- Documentação;
- Gerenciamento de Configuração;
- Verificação;
- Validação;
- Revisão Conjunta;
- Auditoria;
- Resolução de Problemas;
- Garantia da Qualidade de Software;

Atividades da Engenharia de Software

Documentação

Define as atividades para registrar informações produzidas por um processo ou atividade do ciclo de vida. O processo contém atividades para planejar, projetar, desenvolver, produzir, editar, distribuir e manter os documentos necessários a todos os interessados, tais como gerentes, engenheiros e usuários do sistema ou produto de software.

MANUTENÇÃO

Modificação
Revalidação

Atividades de Apoio

- Documentação;
- Gerenciamento de Configuração;
- Verificação;
- Validação;
- Revisão Conjunta;
- Auditoria;
- Resolução de Problemas;
- Garantia da Qualidade de Software;

Atividades da Engenharia de Software

Gerenciamento da Configuração

Conjunto de atividades desenvolvidas para gerenciar e controlar as alterações que ocorrem através de todo processo de software.

MANUTENÇÃO

Modificação
Revalidação

Atividades de Apoio

- Documentação;
- **Gerenciamento de Configuração;**
- Verificação;
- Validação;
- Revisão Conjunta;
- Auditoria;
- Resolução de Problemas;
- Garantia da Qualidade de Software;

Atividades da Engenharia de Software

Verificação

Atividades para determinar se os produtos de software de uma atividade atendem completamente os requisitos ou condições impostas a eles.

“Estamos construindo certo o produto?”

MANUTENÇÃO

Modificação
Revalidação

Atividades de Apoio

- Documentação;
- Gerenciamento de Configuração;
- **Verificação;**
- Validação;
- Revisão Conjunta;
- Auditoria;
- Resolução de Problemas;
- Garantia da Qualidade de Software;

Atividades da Engenharia de Software

Validação

Atividades para assegurar que o produto final corresponde aos requisitos do software.

“Estamos construindo o produto certo?”

MANUTENÇÃO

Modificação
Revalidação

Atividades de Apoio

- Documentação;
- Gerenciamento de Configuração;
- Verificação;
- **Validação;**
- Revisão Conjunta;
- Auditoria;
- Resolução de Problemas;
- Garantia da Qualidade de Software;

Atividades da Engenharia de Software

Revisão Conjunta

Atividades para avaliar se a situação e os produtos de uma atividade de um projeto estão apropriados.

As revisões conjuntas são feitas tanto nos níveis de gerenciamento do projeto como nos níveis técnicos

MANUTENÇÃO

Modificação
Revalidação

Atividades de Apoio

- Documentação;
- Gerenciamento de Configuração;
- Verificação;
- Validação;
- **Revisão Conjunta;**
- Auditoria;
- Resolução de Problemas;
- Garantia da Qualidade de Software;

Atividades da Engenharia de Software

Auditoria

Define as atividades para determinar adequação aos requisitos, planos e contrato, quando apropriado.

MANUTENÇÃO

Modificação
Revalidação

Atividades de Apoio

- Documentação;
- Gerenciamento de Configuração;
- Verificação;
- Validação;
- Revisão Conjunta;
- **Auditoria;**
- Resolução de Problemas;
- Garantia da Qualidade de Software;

Atividades da Engenharia de Software

Resolução de Problemas

Atividades para analisar e resolver problemas de qualquer natureza ou fonte, que são descobertos durante o desenvolvimento, operação, manutenção ou durante outras atividades.

MANUTENÇÃO

Modificação
Revalidação

Atividades de Apoio

- Documentação;
- Gerenciamento de Configuração;
- Verificação;
- Validação;
- Revisão Conjunta;
- Auditoria;
- **Resolução de Problemas;**
- Garantia da Qualidade de Software;

Atividades da Engenharia de Software

Garantia da Qualidade de Software

Define as atividades para fornecer a garantia adequada de que os processos e produtos de software, no ciclo de vida do projeto, estejam em conformidade com os requisitos especificados e sejam aderentes aos planos estabelecidos.

A abrangência do processo inclui questões como garantia da qualidade do produto, do processo e do sistema de qualidade.

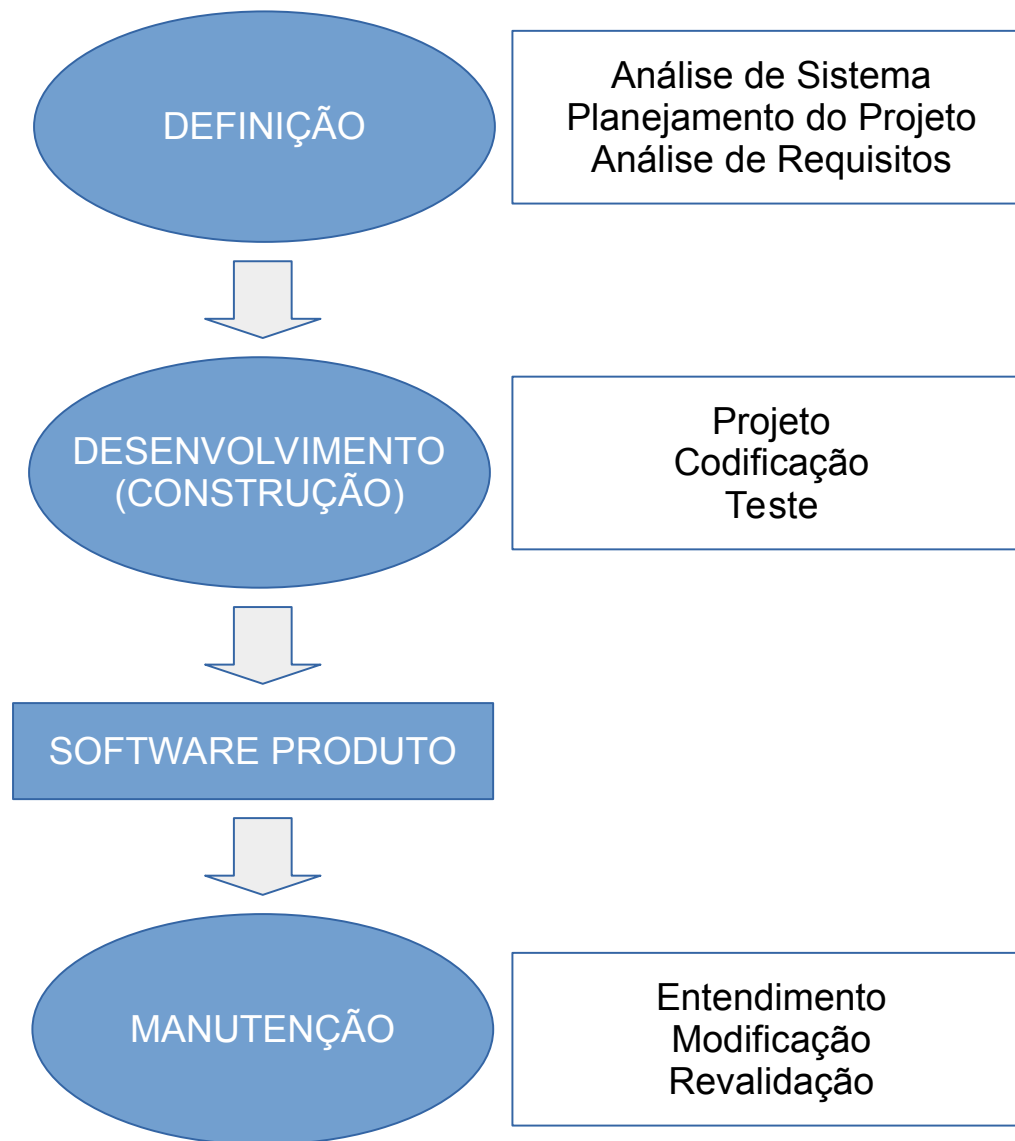
MANUTENÇÃO

Modificação
Revalidação

Atividades de Apoio

- Documentação;
- Gerenciamento de Configuração;
- Verificação;
- Validação;
- Revisão Conjunta;
- Auditoria;
- Resolução de Problemas;
- **Garantia da Qualidade de Software;**

Atividades da Engenharia de Software



Atividades de Apoio

- Documentação;
- Gerenciamento de Configuração;
- Verificação;
- Validação;
- Revisão Conjunta;
- Auditoria;
- Resolução de Problemas;
- Garantia da Qualidade de Software;

Engenharia de Software

- A *Engenharia de Software* é uma disciplina que aplica os princípios de engenharia com o objetivo de produzir software de alta qualidade a baixo custo

Engenharia de Software

- A *Engenharia de Software* é uma disciplina que aplica os princípios de engenharia com o objetivo de produzir software de alta qualidade a baixo custo



O que caracteriza um produto de software com qualidade?

Interpretações de Qualidade de Software

- Qualidade é um termo que pode ter diferentes interpretações e para se estudar a qualidade de software de maneira efetiva é necessário, inicialmente, obter um consenso em relação à definição de qualidade de software que está sendo abordada.

Interpretações de Qualidade de Software

- Existem muitas definições de qualidade de software propostas na literatura, sob diferentes pontos de vista

Aspectos Importantes das Definições de Qualidade

- As definições enfatizam três aspectos importantes:
 - 1) Os requisitos de software são a base a partir da qual a qualidade é medida. A falta de conformidade aos requisitos significa falta de qualidade;

Aspectos Importantes das Definições de Qualidade

- As definições enfatizam três aspectos importantes:
 - 2) Padrões especificados definem um conjunto de critérios de desenvolvimento que orientam a maneira segundo a qual o software passa pelo trabalho de engenharia. Se os critérios não forem seguidos, o resultado quase que seguramente será a falta de qualidade;

Aspectos Importantes das Definições de Qualidade

- As definições enfatizam três aspectos importantes:
 - 3) Existe um conjunto de requisitos implícitos que freqüentemente não são mencionados na especificação (por exemplo o desejo de uma boa manutenibilidade).

Se o software se adequar aos seus requisitos explícitos, mas deixar de cumprir seus requisitos implícitos, a qualidade do software pode ser comprometida.

Outra Visão de Qualidade de Software

- Existe, ainda, uma visão de qualidade de software do ponto de vista gerencial;
- O software que possa ser desenvolvido dentro do prazo e do orçamento especificados pode ser um software de alta qualidade.

Existência de Várias Visões de Qualidade de Software

- Isso demonstra que dentro da qualidade de software, pode-se definir várias visões diferentes, como tem sido para a definição da qualidade como um termo geral.

Definição Genérica de Qualidade de Software

- De um modo geral, qualidade de software pode ser definida como:

Um conjunto de *atributos* de software que devem ser satisfeitos de modo que o software atenda às necessidades do usuário (seja ele um usuário final, um desenvolvedor ou uma organização).

Qualidade de Software

Definição:

“Um produto de software apresenta qualidade dependendo do grau de satisfação das necessidades dos clientes sob todos os aspectos do produto” [Sanders, 1994].

Qualidade de Software

Definição:

“Qualidade de software é a conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados, e a características implícitas que são esperadas de todo software desenvolvido por profissionais” [Pressman, 1994].

Qualidade de Software

Definição:

*“Qualidade é a totalidade de características e critérios de um produto ou serviço que exercem sua habilidades para satisfazer às necessidades declaradas ou envolvidas”
[ISO9126, 1994].*

Determinação dos Atributos Relevantes

- A determinação dos atributos relevantes para cada software varia em função:
 - do domínio da aplicação;
 - das tecnologias utilizadas;
 - das características específicas do projeto;
 - das necessidades do usuário e da organização.

A Qualidade depende do Ponto de Vista



Usuário



Desenvolvedor



Gerente

Interesse fica concentrado principalmente no uso do software

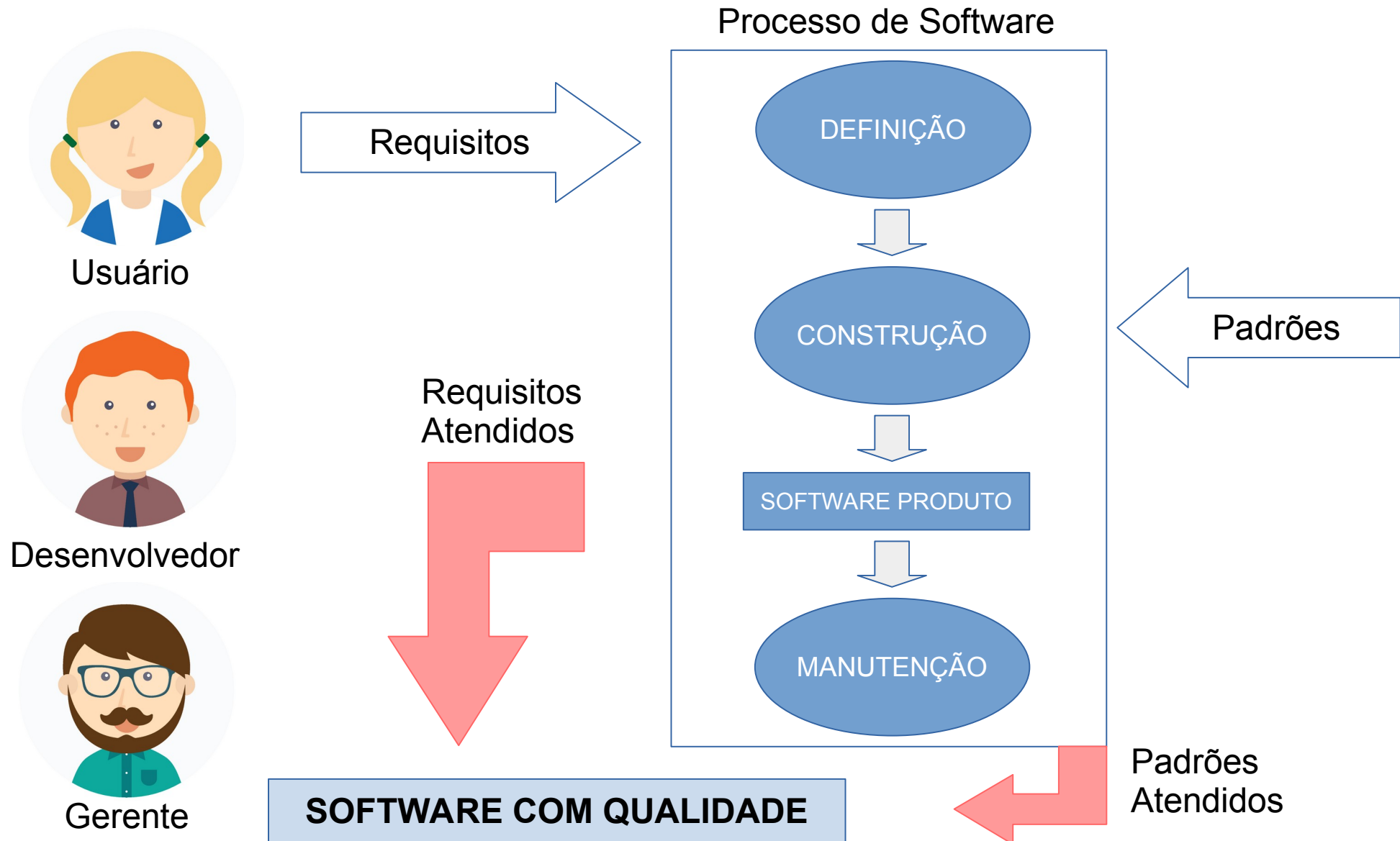
A qualidade fica mais voltada às características internas do software

A qualidade do produto não pode ser desvinculada dos interesses da organização

Requisitos de Software - Base da Qualidade

- Os requisitos de software formam a base a partir da qual a qualidade é avaliada;
- Tipos de requisitos:
 - Explícitos;
 - Implícitos.
- O software deve atender aos requisitos explícitos e implícitos para ter qualidade.

A Qualidade depende do Ponto de Vista



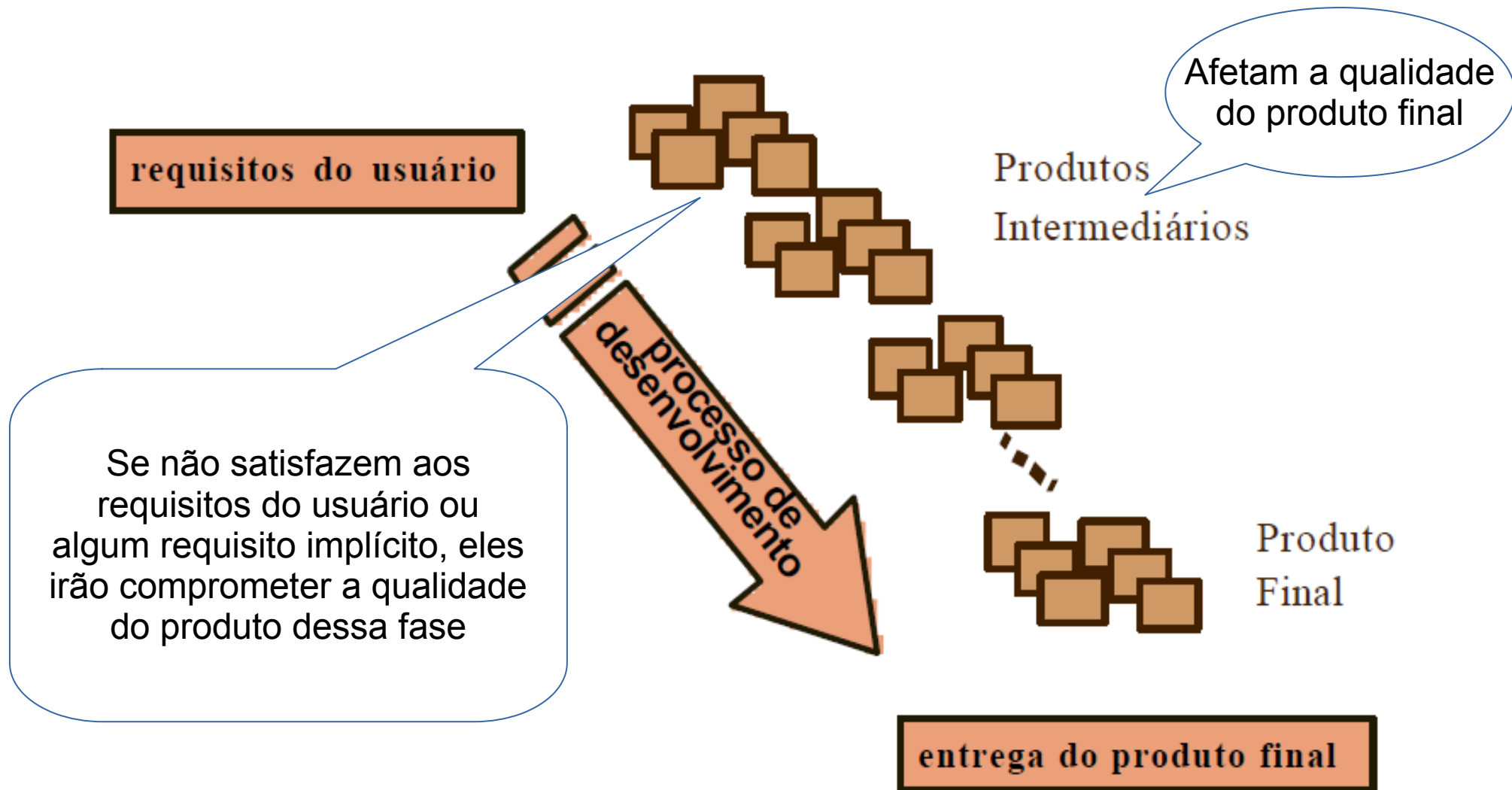
Incorporação da Qualidade

- Raramente qualidade pode ser incorporada ao produto final, após o processo de desenvolvimento ter terminado.

Incorporação da Qualidade

- Dos requisitos do usuário à entrega do produto final, existe um processo de desenvolvimento que é complexo e freqüentemente envolve uma série de estágios que podem comprometer a qualidade do produto final.

Incorporação da Qualidade: Qualidade dos Produtos Intermediários



Qualidade do Produto Final

- Cada produto intermediário tem certos atributos de qualidade que afetam a qualidade do produto intermediário da próxima fase e assim, afetam a qualidade do produto final.

Qualidade de Software



- A qualidade não pode ser incorporada ao produto depois de pronto
- Para que a qualidade possa ser efetivamente incorporada ao produto, ela deve ser um objetivo constante do processo de desenvolvimento.
- Processo de garantia de qualidade do software – atividades de garantia da qualidade de software (GQS)

Histórico

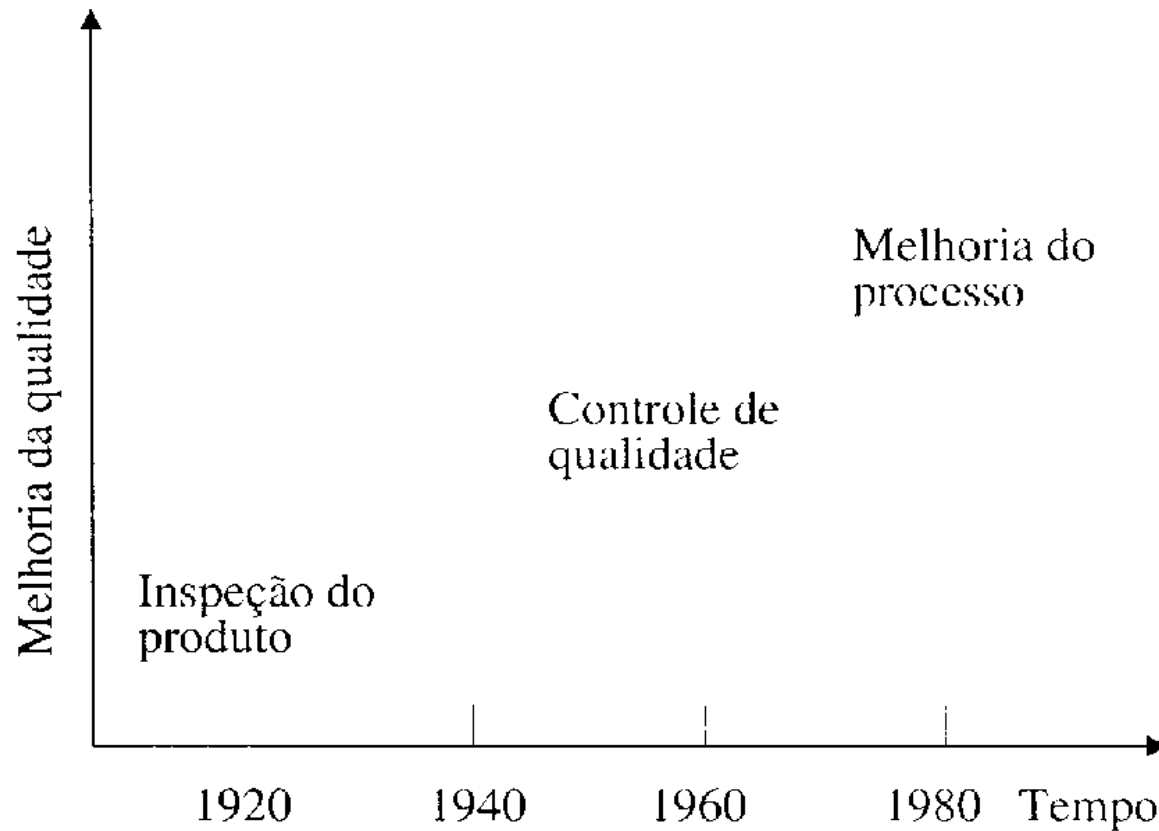


Figura 2-1: Evolução da tecnologia da qualidade [Card 90].

Histórico

Inspeção do produto

Os produtos são examinados para detecção de defeitos

Controle de qualidade

Monitoramento das taxas de defeitos e dos custos para identificação dos elementos defeituosos do processo

Melhoria do processo

O processo de produção é melhorado

Normalização

Normalizar é estabelecer diretrizes para obtenção de um grau ótimo de ordem, em determinado contexto

Objetivos específicos

- Adequação ao propósito

- Compatibilidade

- Controle da variedade

- Segurança

- Proteção do meio ambiente

- Proteção do produto

Organismos para Normalização

Internacionais

IEC – International Electrotechnical Commission –
normalização na área de eletricidade/eletrônica

ISO – International Organization for Standardization –
demais áreas

Nacional

ABNT – Associação Brasileira de Normas Técnicas –
representante Brasileiro na ISO/IEC

Organismos para Normalização

ISO 9000

Família de normas que oferecem um conjunto de diretrizes e requisitos mínimos para gestão da qualidade e garantia de qualidade

Medição

Métricas e técnicas de medição estabelecidas para execução de medições nos produtos e processos desde o desenvolvimento até a obtenção do produto final

Monitoramento e medição de características do produto

Certificação

Comprovação e declaração de que uma empresa tem produtos, processos ou serviços em conformidade com determinadas normas.

Concedida para cada base da empresa individualmente

Conferida por empresa credenciada

Solicitada pela empresa ou cliente da mesma

Realizada por auditores de sistemas de qualidade

Gerenciamento de qualidade

Responsável por:

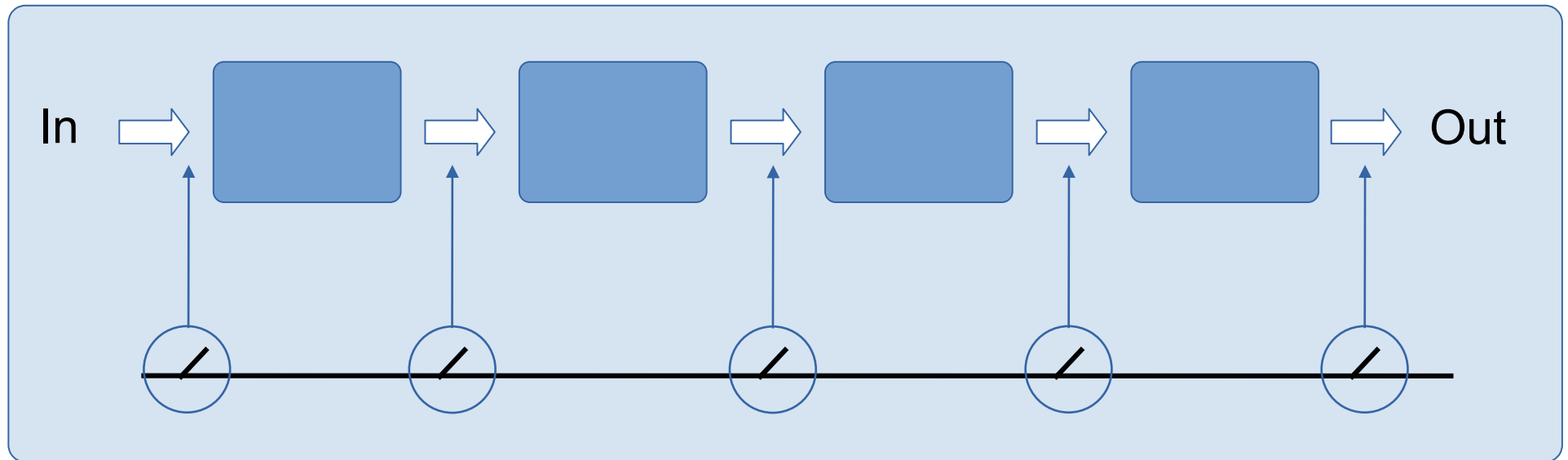
- Garantir o nível de qualidade

- Definir os procedimentos e padrões apropriados

- Verificar se os procedimentos e padrões estão sendo seguidos

Emprego de técnicas para controlar o nível de qualidade e corrigir problemas

Garantia de Qualidade



A ***Garantia da Qualidade de Software*** envolve revisões nos produtos de software e nas atividades para assegurar que os processos e produtos de software, estejam em conformidade com os requisitos especificados e sejam aderentes aos planos estabelecidos.

Garantia de Qualidade de Software - GQS

- É um padrão sistemático e planejado de ações que são exigidas para garantir a qualidade de software

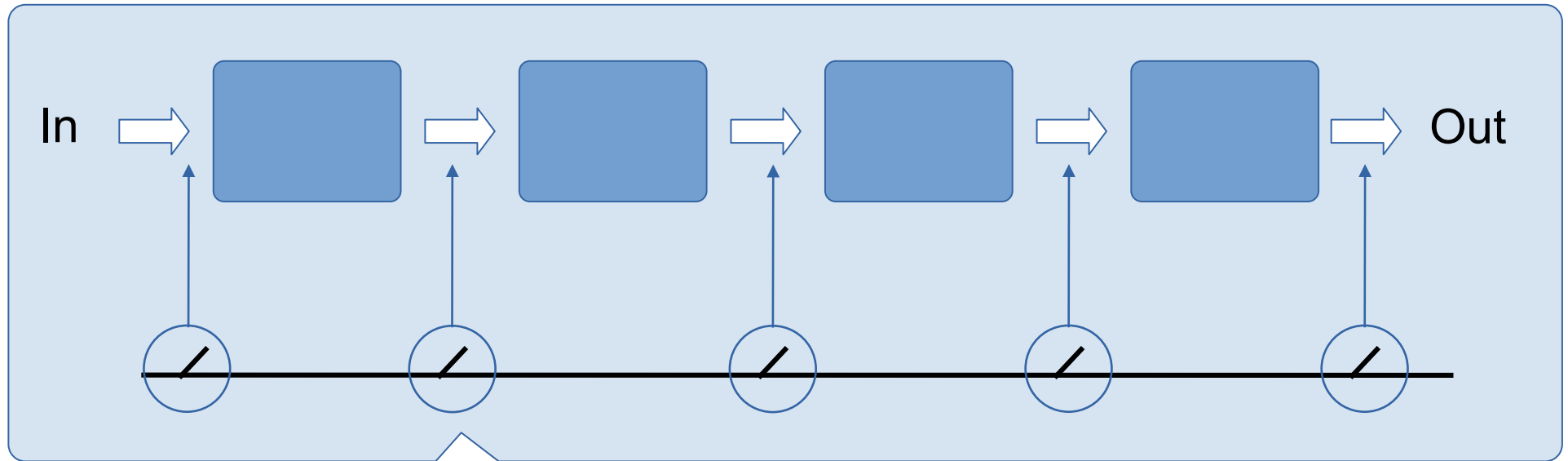
Ações para Garantir a Qualidade de Software

- *Aplicação de Métodos Técnicos*
 - ajudam o analista a conseguir uma especificação de elevada qualidade e o projetista a desenvolver um projeto de elevada qualidade
- *Aplicação de Padrões e Procedimentos Formais*
 - para o processo de engenharia de software

Ações para Garantir a Qualidade de Software

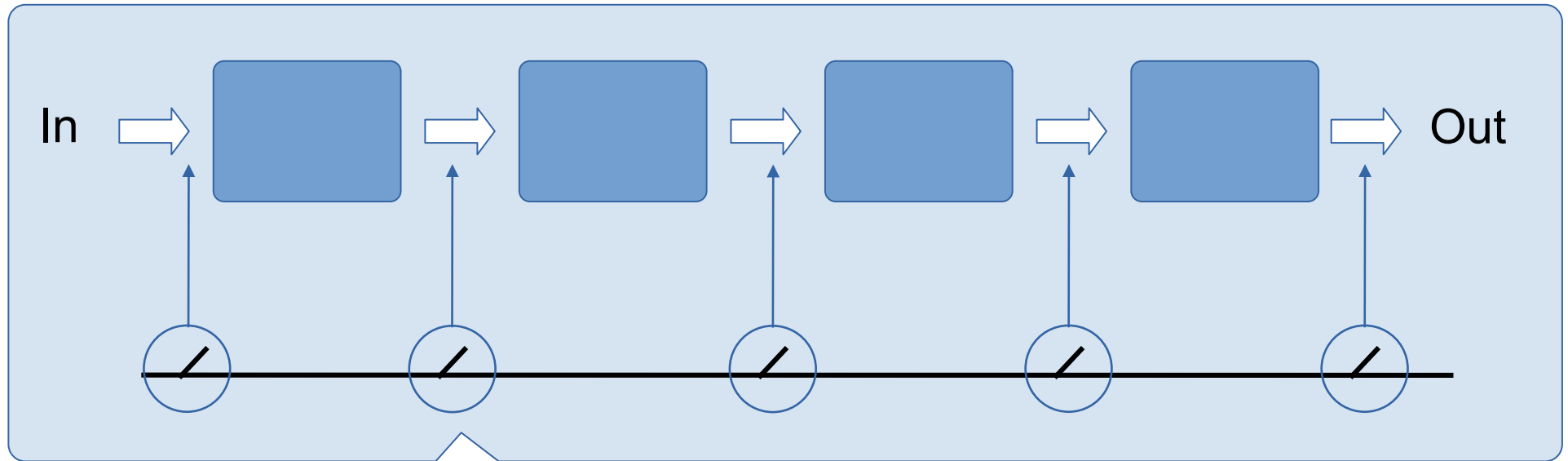
- *Anotação e Manutenção de Registros:*
 - procedimentos para a coleta e disseminação de informações de garantia de qualidade de software.

O Processo de Garantia de Qualidade



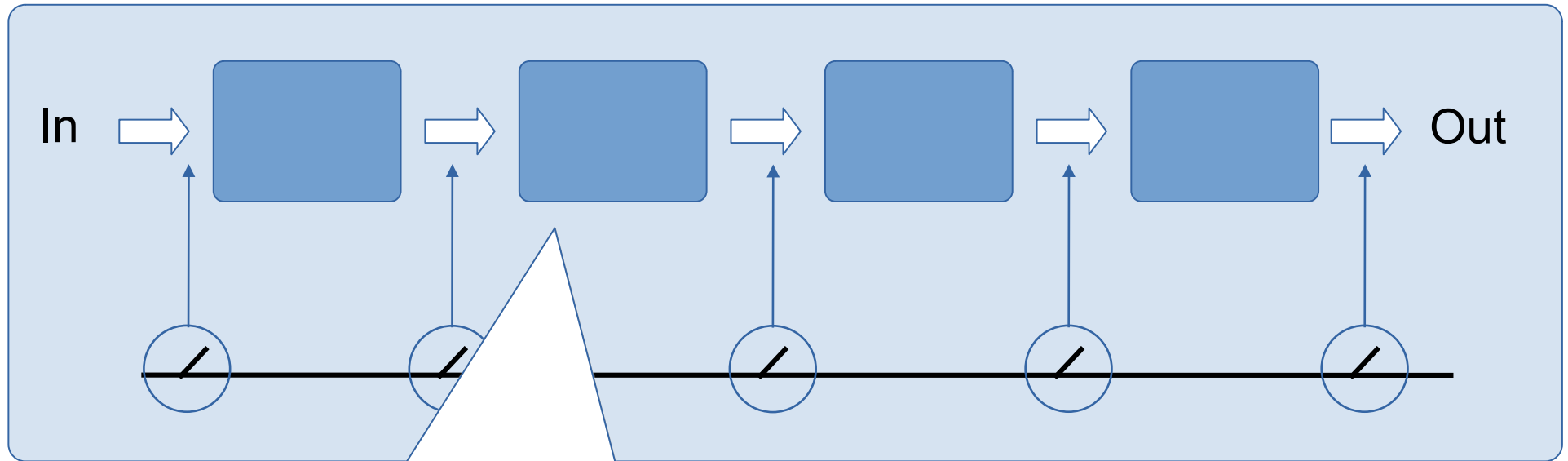
Revisar os produtos intermediários com relação aos ***requisitos de qualidade*** estabelecidos.

O Processo de Garantia de Qualidade



Revisar as atividades com relação
aos ***planos*** estabelecidos

O Processo de Garantia de Qualidade



Como avaliar os *produtos* e os *processos*?

Avaliação – qualidade de software

- Qualidade do Produto

- ISO 9126/IEC 9126 publicada em 1991

Traduzida NBR 13596

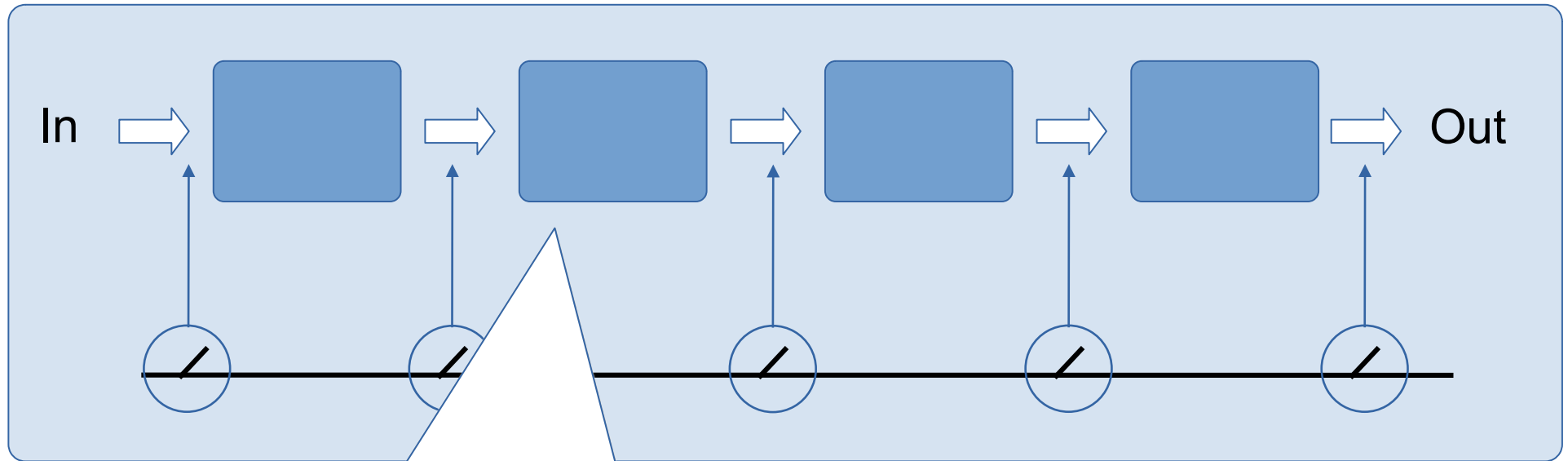
- Qualidade do Processo

- CMMI
- ISO 12207
- SPICE

Avaliação do Processo de Software

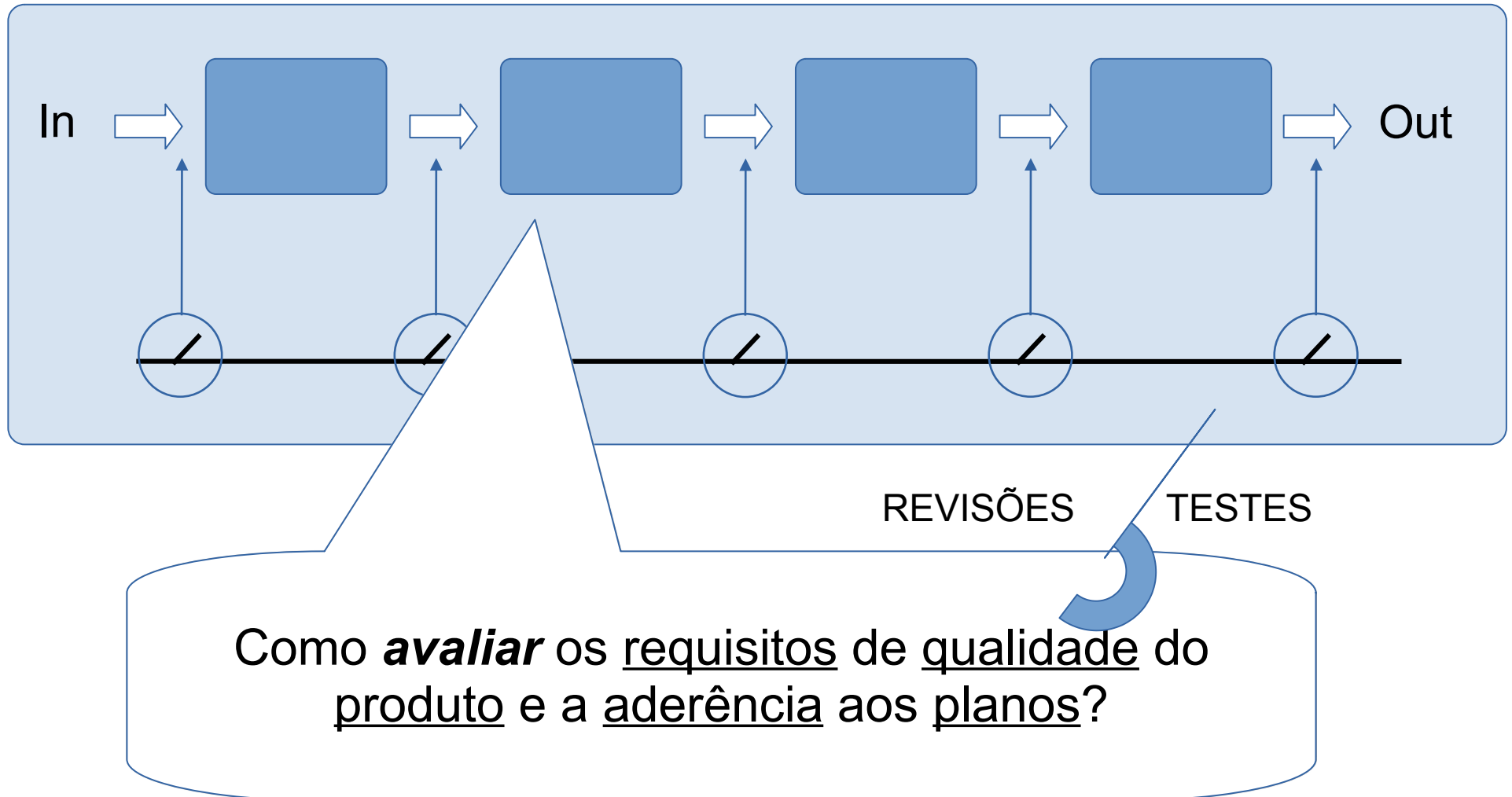
- Os processos devem estar aderentes aos planos estabelecidos.
 - Plano de Projeto (Rastreamento do Progresso);
 - Plano de Gerenciamento de Configuração;
 - Plano de Documentação;
 - Plano de Monitoração de Riscos;
 - Etc.

O Processo de Garantia de Qualidade



Como ***avaliar*** os requisitos de qualidade do produto e a aderência aos planos?

O Processo de Garantia de Qualidade



Atividades de GQS

- Realização de Revisões:
 - para avaliar a qualidade da especificação e projeto.
- Atividades de Teste de Software
 - para ajudar a garantir uma detecção efetiva de erros.
- Auditoria:
 - para determinar adequação aos requisitos, planos e contrato.

Revisões do Software

- As Revisões devem ser aplicadas em vários pontos durante o desenvolvimento do software
- Revisão é uma maneira de usar a diversidade de um grupo de pessoas para:
 - apontar melhorias necessárias ao produto;
 - confirmar a aderência aos planos pré estabelecidos.

Revisões Técnicas Formais - Objetivos

- 1) Descobrir erros de função, lógica ou implementação em qualquer representação do software;
- 2) Verificar se o software que se encontra em revisão atende a seus requisitos;
- 3) Garantir que o software tenha sido representado de acordo com padrões pré-definidos;
- 4) Obter um software que seja desenvolvido uniformemente;
- 5) Tornar os projetos mais administráveis.

Revisões Técnicas Formais

Além disso:

- Espaço de treinamento que possibilita que os engenheiros juniores observem diferentes abordagens a análise, projeto e implementação de software
- Promove backup e continuidade. Várias pessoas se familiarizam com partes do software que de outro modo poderiam não conhecer.

Reunião da Revisão de Software

- Uma Revisão Técnica Formal é conduzida em uma reunião e será bem sucedida se for planejada, controlada e cuidada.
- A Revisão Técnica Formal focaliza uma parte específica (pequena) do software – maior probabilidade de descobrir erros.

Reunião da Revisão de Software

- Independentemente do formato de Revisão Técnica, toda Reunião de Revisão deve estar de acordo com:
 - 1) Envolver de 3 a 5 pessoas na revisão
 - 2) Deve haver uma preparação para a reunião (essa preparação não deve exigir mais de 2 horas de trabalho de cada pessoa)
 - 3) A Reunião de Revisão deve durar menos de 2 horas

Revisão de Software

- Verificação: determinar se os produtos de software de uma atividade atendem completamente os requisitos ou condições impostas a eles.
 - “Estamos construindo certo o produto?”
- Validação: assegurar que o produto final corresponde aos requisitos do software.
 - “Estamos construindo o produto certo?”

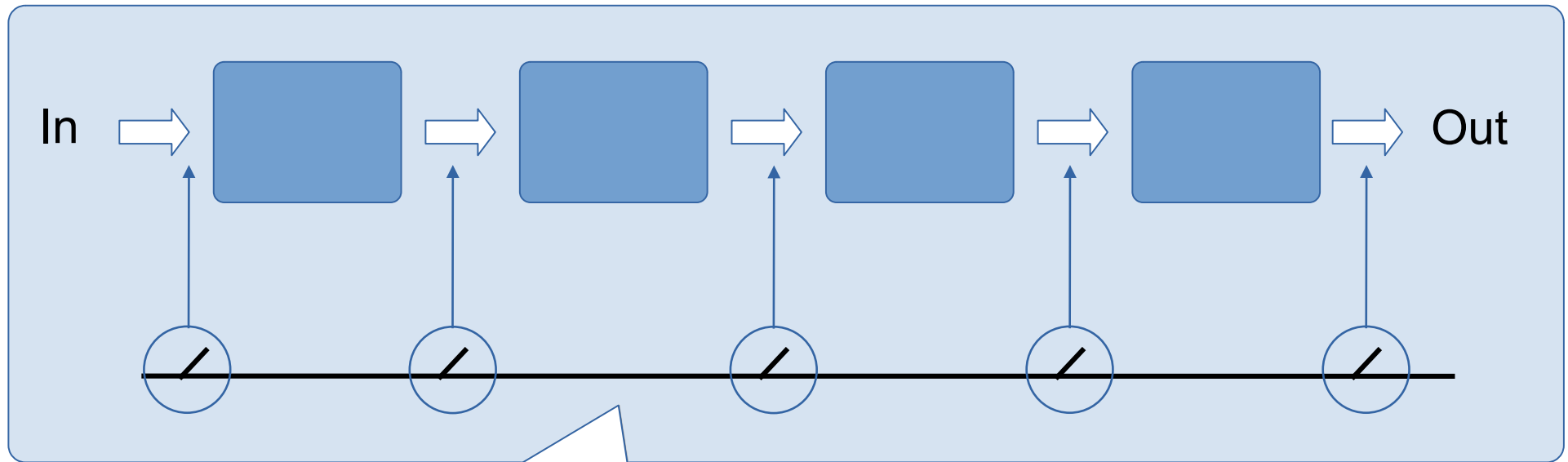
Revisão de Software

- Revisões Conjuntas: avaliar se a situação e os produtos de uma atividade de um projeto estão apropriados.
- As revisões conjuntas são feitas tanto nos níveis de gerenciamento do projeto como nos níveis técnicos.

Revisão de Software

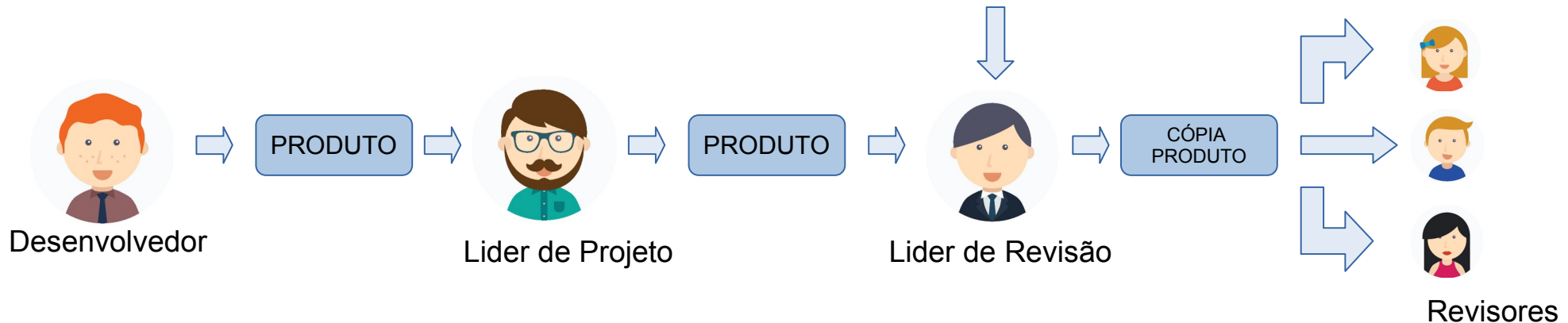
- Resolução de Problemas: analisar e resolver problemas de qualquer natureza ou fonte, que são descobertos durante o desenvolvimento, operação, manutenção ou durante outras atividades.

O Processo de Garantia de Qualidade

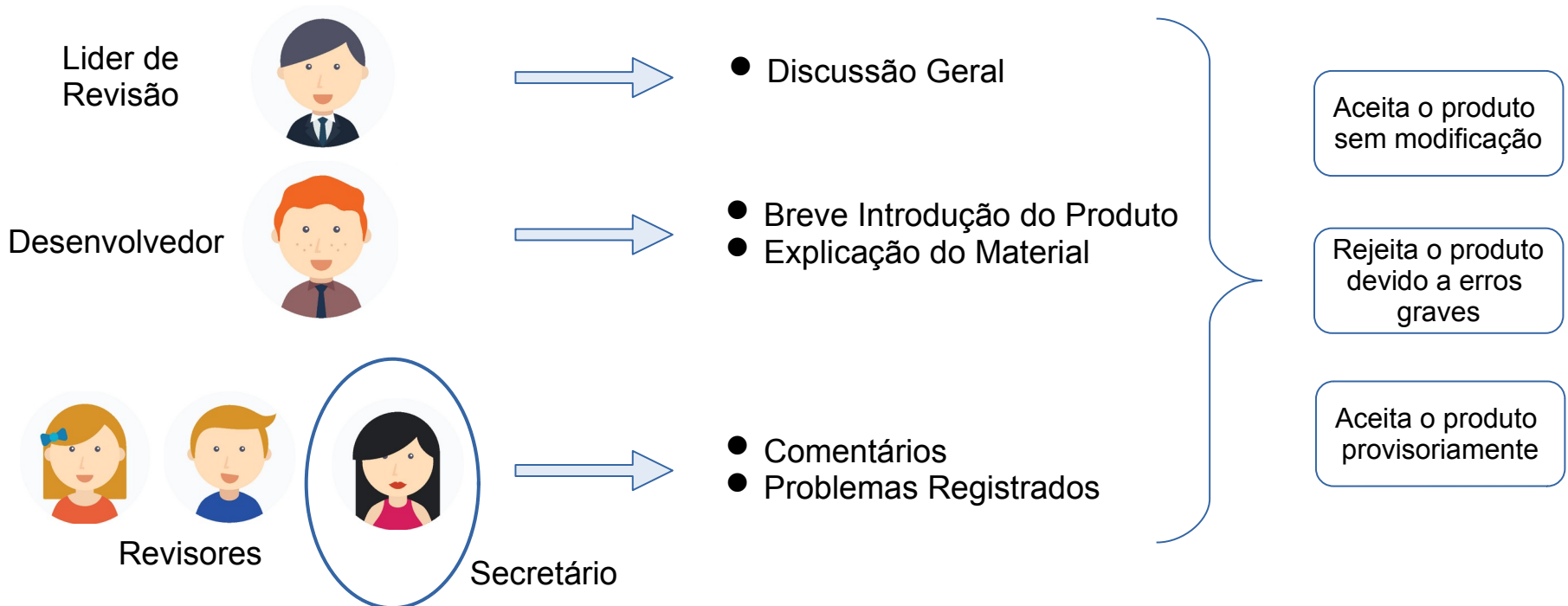


Auditoria: atividades para determinar adequação aos requisitos, planos e contrato.

Reunião da Revisão de Software



REUNIÃO DE REVISÃO



Diretrizes de Revisão de Software

- Devem ser estabelecidas antecipadamente, distribuídas a todos os revisores, ter a concordância de todos e então encaminhada

Diretrizes de Revisão de Software

- Conjunto mínimo de diretrizes para as revisões técnicas formais:
 - 1)Revise o produto, não o produtor.
 - 2)Fixe e mantenha uma agenda
 - 3)Limite o debate e a refutação
 - 4)Enuncie as áreas problemáticas, mas não tente resolver cada problema anotado.

Diretrizes de Revisão de Software

- Conjunto mínimo de diretrizes para as revisões técnicas formais:

5)Faça anotações por escrito.

6)Limite o número de participantes e insista numa preparação antecipada.

7)Desenvolva uma lista de conferência (checklist) para cada produto que provavelmente será revisto.

Diretrizes de Revisão de Software

- Conjunto mínimo de diretrizes para as revisões técnicas formais:
 - 8)Atribua recursos e uma programação de tempo para as Revisões Técnicas Formais
 - 9)Realize um treinamento significativo para todos os revisores.
 - 10)Reveja suas antigas revisões.

Uma Abordagem à Garantia de Qualidade de Software

- A necessidade de qualidade de software é reconhecida por praticamente todos os gerentes e profissionais da área, porém muito poucos estão interessados em estabelecer funções de Garantia de Qualidade de Software Formais.

Uma Abordagem à Garantia de Qualidade de Software

RAZÕES PARA ESSA CONTRADIÇÃO

- Os gerentes relutam em incorrer em custos extras logo de início
- Os profissionais acham que estão fazendo absolutamente tudo o que precisa ser feito
- Ninguém sabe onde colocar essa função organizacionalmente
- Todos querem evitar a burocracia que, segundo entendem, a Garantia de Qualidade de Software introduzirá no processo de engenharia de software

Uma Abordagem à Garantia de Qualidade de Software

- **Aspectos positivos da GQS:**
 - O software terá menos defeitos latentes resultando em redução do esforço e do tempo gasto durante as atividades de teste e manutenção
 - A maior confiabilidade resultará em maior satisfação do cliente
 - Os custos de manutenção podem ser reduzidos
 - O custo do ciclo de vida global do software é reduzido



Uma Abordagem à Garantia de Qualidade de Software

- **Aspectos negativos da GQS:**
 - Difícil de ser instituída em pequenas empresas;
 - Representa uma mudança cultural - e uma mudança nunca é fácil;
 - Exigem gastos que não poderiam ser orçamentados explicitamente.



Qualidade do Produto

Fatores de McCall

Manutenibilidade: posso consertá-lo?

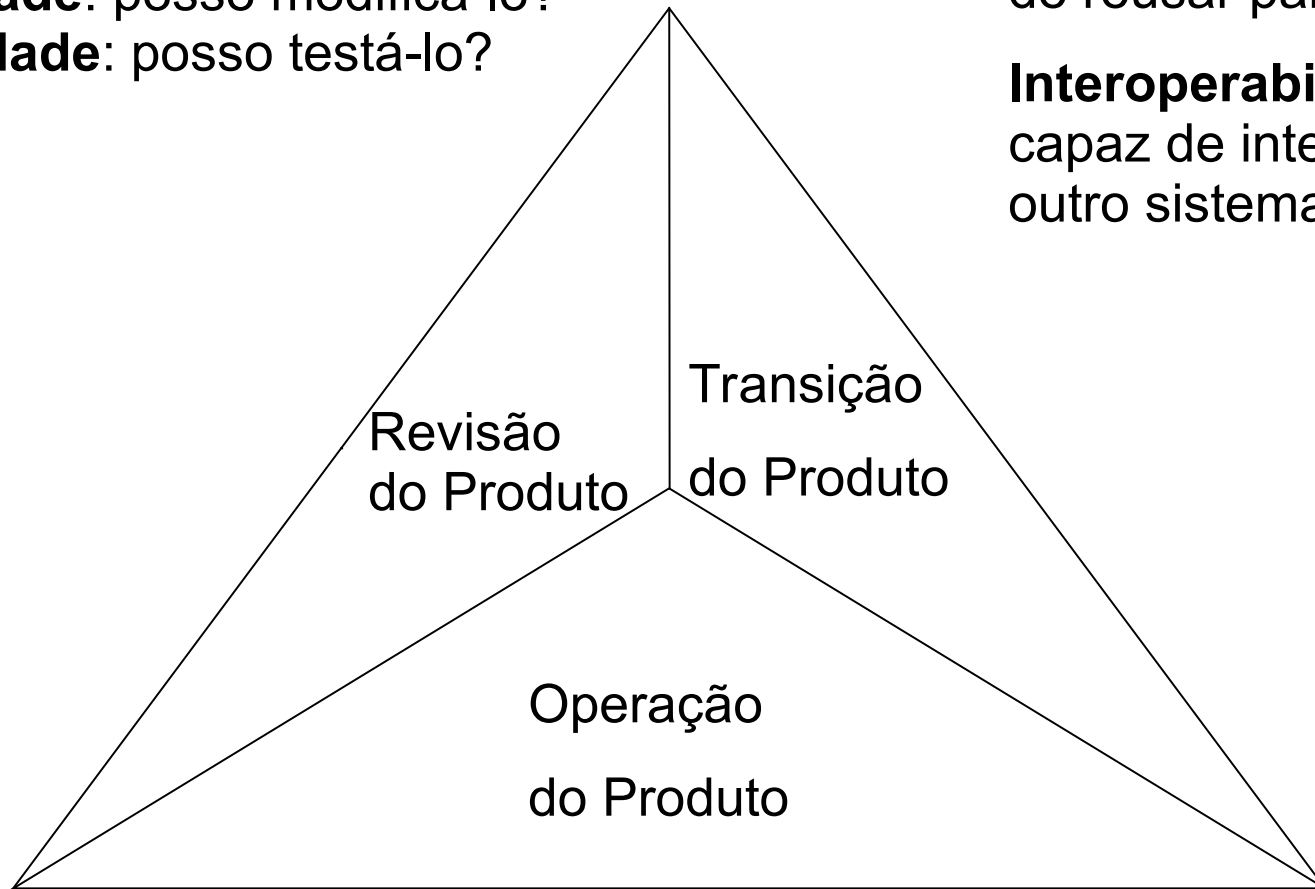
Flexibilidade: posso modificá-lo?

Testabilidade: posso testá-lo?

Portabilidade: serei capaz de usá-lo em outra máquina?

Reusabilidade: serei capaz de reusar partes do software?

Interoperabilidade: serei capaz de interfaceá-lo com outro sistema?



Correção: faz o que eu quero?

Integridade: é seguro?

Eficiência: executará em meu hw tão bem qto pode?

Confiabilidade: fá-lo precisamente todo o tempo?

Usabilidade: posso executá-lo?

Norma ISO/IEC 9126

Duas partes:

Modelo de qualidade para características externas e internas → seis características que se desdobram em sub-características

Modelo de qualidade para qualidade em uso → quatro características

Características

O QUE

Funcionalidade

QUANDO e COMO

Confiabilidade

Usabilidade

Eficiência

Manutenibilidade

Portabilidade

ISO/IEC 9126 - Características

- Padronização mundial para a qualidade de software do produto.
- Baseada em três níveis: Características, Sub-características e Métricas.
- Cada característica é refinada em um conjunto de sub-características e cada sub-característica é avaliada por um conjunto de métricas.

ISO/IEC 9126 - Características

- As características são:
 - Funcionalidade – Satisfaz as necessidades?
 - Confiabilidade – É imune a falhas?
 - Usabilidade – É fácil de usar?
 - Eficiência – É rápido e “enxuto”?
 - Manutenibilidade – É fácil de modificar?
 - Portabilidade – É fácil de usar em outro ambiente?

ISO/IEC 9126 - Características

- Funcionalidade: *“Conjunto de atributos que evidenciam a existência de um conjunto de funções e suas propriedades especificadas. As funções são as que satisfazem as necessidades explícitas e implícitas”.*

Funções especificadas e suas propriedades

As funções devem satisfazer as necessidades implícitas e explícitas do usuário

Caracterizam o que o software faz

ISO/IEC 9126 - Características

- Usabilidade: *“Conjunto de atributos que evidenciam o esforço necessário para se poder utilizar o software, bem como o julgamento individual desse uso, por um conjunto explícito ou implícito de usuários”.*

Medida de esforço necessário para um usuário de perfil determinado usar o software

ISO/IEC 9126 - Características

- Eficiência: *“Conjunto de atributos que evidenciam o relacionamento entre o nível de desempenho do software e a quantidade de recursos usados, sob condições de uso estabelecidas”.*

ISO/IEC 9126 - Características

- Confiabilidade: *“Conjunto de atributos que evidenciam a capacidade do software de manter seu nível de desempenho sob certas condições por um dado período de tempo.”*

ISO/IEC 9126 - Características

- Manutenibilidade: *“Conjunto de atributos que evidenciam o esforço necessário para fazer modificações especificadas no software”.*

ISO/IEC 9126 - Características

- Portabilidade: *“Conjunto de atributos que evidenciam a capacidade do software de ser transferido de um ambiente para outro”.*

ISO/IEC 9126 – Sub-características

Funcionalidade

Adequação: presença das funções especificadas

Acurácia: gerar os resultados precisos ou esperados

Interoperabilidade: interagir e interoperar com outros sistemas conforme o esperado

Conformidade: observância de padrões, convenções ou regras já definidas

Segurança de acesso: prevenir acesso não autorizado

ISO/IEC 9126 – Sub-características

Confiabilidade

Maturidade: baixa frequência de falhas

Tolerância a falhas: manter determinados níveis de desempenho mesmo na presença de problemas

Recuperabilidade: restabelecer o nível de desempenho desejado e recuperar dados no caso de falhas

ISO/IEC 9126 – Sub-características

Usabilidade

Inteligibilidade: facilidade do usuário reconhecer como é o funcionamento do produto e sua aplicação

Apreensibilidade: facilidade do usuário em aprender a utilizar o produto

Operacionalidade: facilidade do usuário para operar o produto

ISO/IEC 9126 – Sub-características

Eficiência

Comportamento em relação ao tempo: medida do tempo de resposta e de processamento

Comportamento em relação ao uso de recursos: medida de quantidade de recursos necessários e a duração de seu uso ao executar suas funções

ISO/IEC 9126 – Sub-características

Manutenibilidade

Analísabilidade: medida de esforço necessário para diagnosticar ou localizar deficiências ou defeitos

Modificabilidade: medida de esforço necessário para realizar alterações

Estabilidade: medida do risco de efeitos inesperados provenientes de modificações

Testabilidade: medida de esforço necessário para testar o software alterado

ISO/IEC 9126 – Sub-características

Portabilidade

Adaptabilidade: medida da facilidade de se adaptar o produto em outros ambientes operacionais

Facilidade de instalação: medida de esforço necessário para a instalação do produto

Capacidade para coexistir: medida do nível de conformidade do produto com padrões referentes à portabilidade

Facilidade para substituir: medida do esforço necessário para que o produto seja usado em substituição a outro

ISO/IEC 9126 - Modelo de qualidade em uso - Características

Efetividade

Produto de software permite ao usuário atingir as metas especificadas com acurácia e completeza

Produtividade

Produto de software permite ao usuário usar um número adequado de recursos em relação a efetividade obtida

ISO/IEC 9126 - Modelo de qualidade em uso - Características

Segurança

Produto de software oferece níveis aceitáveis de risco de danos aos elementos relacionados ao seu uso

Satisfação

Produto de software satisfaz ao usuário

Qualidade do processo

Histórico

1986- O DoD patrocinou a fundação do SEI (Software Engineering Institute) na Universidade de Carnegie Mellon – Pittsburgh) com o objetivo de propor um modelo para avaliar seus fornecedores.

O intuito era que esses desenvolvessem sw com o mesmo nível de repetibilidade e controle encontrado em outros setores

Histórico

1987 – uma descrição sumária do modelo de maturidade de sw que gerou m livro “Managing the software process” – Humphrey, 1989.

1991 – 1ª versão do CMM (Versão 1.0)

1993 – surgiu o CMM, com 4 anos de experimentação

08/2012 – versão 1.3 CMMI – Capability Maturity Model Integration – meta-modelo, avalia em níveis cada área.

Processo de Software

Definição

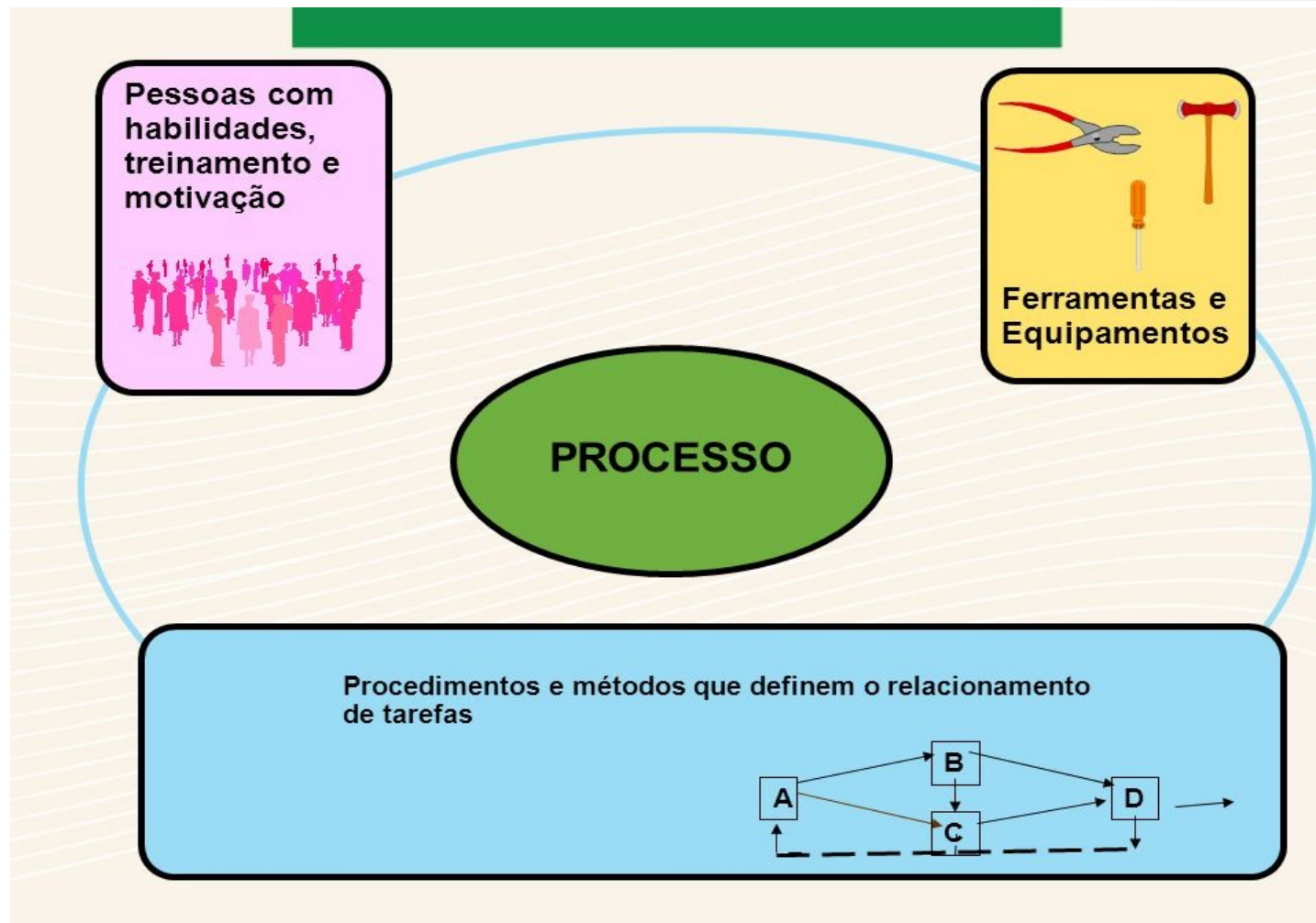
- **processo:** uma sequência de passos realizados para um determinado propósito (IEEE).
- **processo de software:** conjunto de atividades, métodos, práticas e tecnologias que as pessoas utilizam para desenvolver e manter software e produtos relacionados (CMM).

Processo de Software

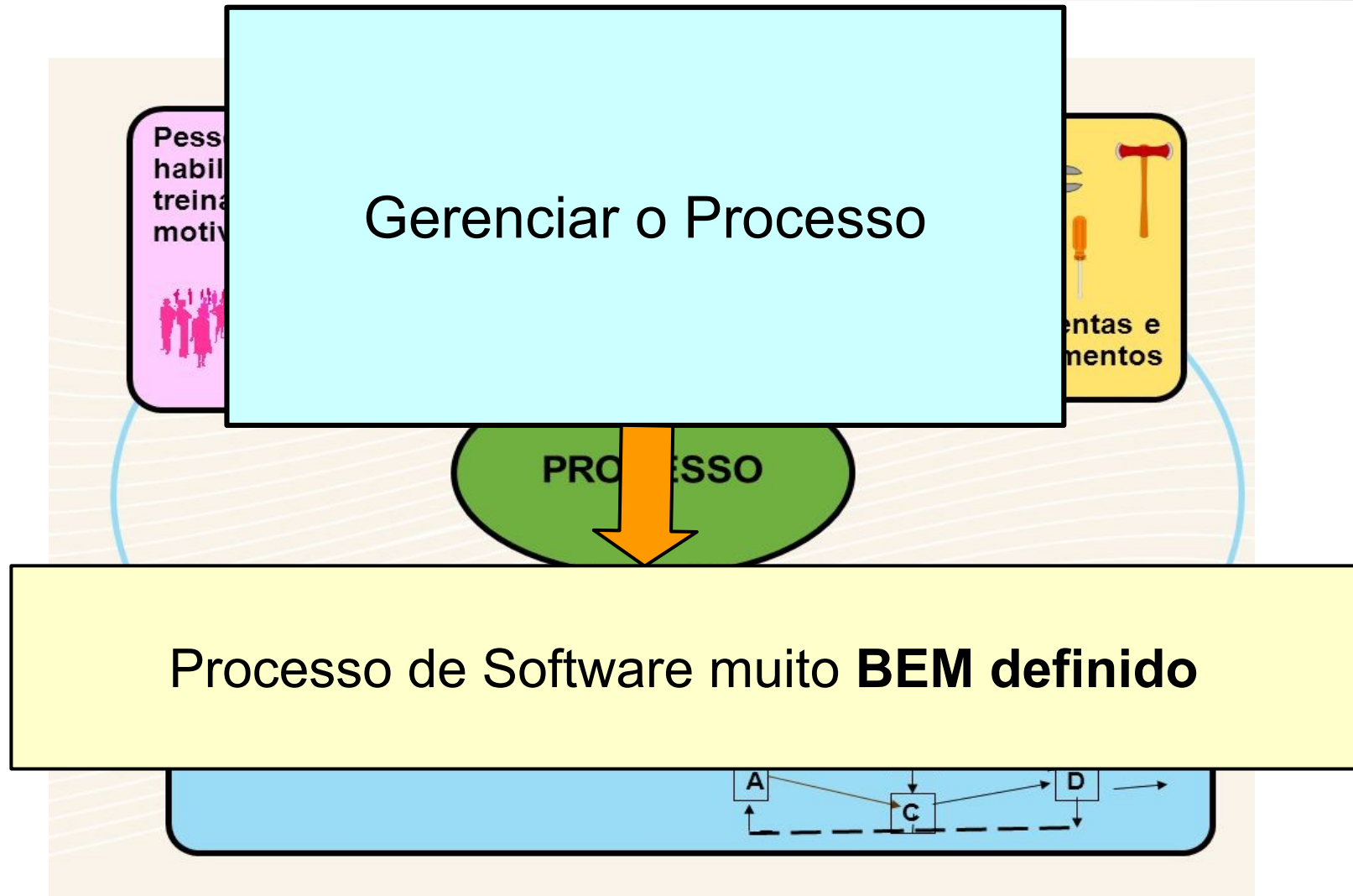
Definição

- **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.

Processo de Software

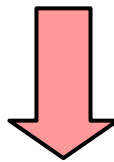


Processo de Software



Modelo de Processo de Software

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



Modelo de Processo de Software

Modelo de Processo de Software

- Um modelo de processo procura descrever formalmente e de maneira organizada todas as atividades que devem ser seguidas para a obtenção segura de um produto de software.
- É importante escolher um modelo apropriado às metas da organização e saber o grau em que esse modelo será implementado.
- Modelos CMM, SPICE e Bootstrap são padrões relacionados a processo de software.

Uma empresa imatura:

- os processos de desenvolvimento são improvisados e se existem não seguidos
- o trabalho é feito em regime de emergência
- dificilmente se cumpre compromissos e prazos
- não se faz planejamento com estimativas realistas
- o sucesso de um projeto depende de alguns excepcionais na equipe

Um Processo de Software Imaturo

- Ad hoc; processo improvisado por profissionais e gerências.
- Não é rigorosamente seguido e o cumprimento não é controlado.
- Altamente dependente dos profissionais atuais.
- Baixa visão do progresso e da qualidade.

Um Processo de Software Imaturo

- A qualidade do produto pode ficar comprometida para que prazos sejam cumpridos.
- Arriscado do ponto de vista do uso de nova tecnologia.
- Custos de manutenção excessivos.
- Qualidade difícil de se prever.

Um Processo de Software Maduro

- Coerente com as linhas de ação, o trabalho é efetivamente concluído.
- Definido, documentado e melhorando constantemente:
 - Compreendido
 - Utilizado
 - Vivo e ativo

Um Processo de Software Maduro

- Com o apoio visível da alta administração e outras gerências.
- Bem controlado – fidelidade ao processo é objeto de auditoria e de controle.
- São utilizadas medições do produto e do processo.
- Uso disciplinado da tecnologia.

Maturidade de Processo de Software

- A qualidade do processo de software pode ser analisada através do nível de maturidade do processo.
- A maturidade dos processos de software de uma organização influencia na sua capacidade de atingir metas de custo, qualidade e cronograma.

Maturidade

Reflete uma medida do quanto o processo pode ser definido, gerenciado, medido, controlado e executado de maneira eficaz.

Capacidade

Descreve a faixa de resultados de um processo de sw esperados dentro de uma margem de probabilidade

Modelo de Processo de Software

Benefícios de uma melhoria baseada em um modelo:

- Estabelece uma linguagem comum.
- Constrói um conjunto de processos e procedimentos desenvolvidos com sugestões de uma ampla participação da comunidade de software.
- Oferece uma estrutura para se priorizar as ações
- Provê uma estrutura para se realizar avaliações confiáveis e consistentes.
- Auxilia comparações em diversas indústrias.

Modelo de Processo de Software

Riscos de uma melhoria baseada em um modelo:

- Modelos são simplificações do mundo real.
- Modelos não são suficientemente abrangentes.
- Interpretação e adaptação à situações particulares devem estar ajustadas aos objetivos do negócio.
- É necessário bom senso para se utilizar modelos corretamente e com visão.

Valor da Melhoria de Processo

- A melhoria do processo de software oferece um retorno no investimento que pode ser medido – quando é medido.
- Benefícios adicionais são intangíveis e não podem ser quantificados facilmente.
- O CMM é uma ferramenta útil para orientação no processo de melhoria.

Capability Maturity Model (CMM)

O primeiro ponto importante a se observar no modelo CMM:

- É um modelo que se refere ao processo pelo qual o produto é construído.

Capability Maturity Model (CMM)

- Supõe-se que uma organização que possui um processo maduro tem maiores probabilidades de produzir bons produtos consistentemente do que uma outra cujo processo é imaturo e caótico.

Capability Maturity Model (CMM)

- O CMM é um modelo de referência para a qualidade de processo de produção de software.
- Através de um processo formal de avaliação, uma organização é classificada em um “nível de maturidade” que varia de um a cinco.

Capability Maturity Model (CMM)

- O nível de maturidade indica em que medida os processos daquela organização são maduros.
- Quanto maior o nível de maturidade, melhores e mais maduros são os processos.

Os 5 Níveis de Maturidades do CMM



Os 5 Níveis de Maturidades do CMM

- Os 5 Níveis do CMM são organizados de modo a estabelecer as prioridades na condução de programas de melhoria do processo de software.
- Cada Nível é considerado como pré-requisito para o nível seguinte.
- Não é possível o “salto” de níveis.
- Cada nível apoia-se nas competências que a organização desenvolveu no nível imediatamente inferior.

INICIAL

Organizações
Caóticas

ADO

Organizações
em Melhoria
contínua

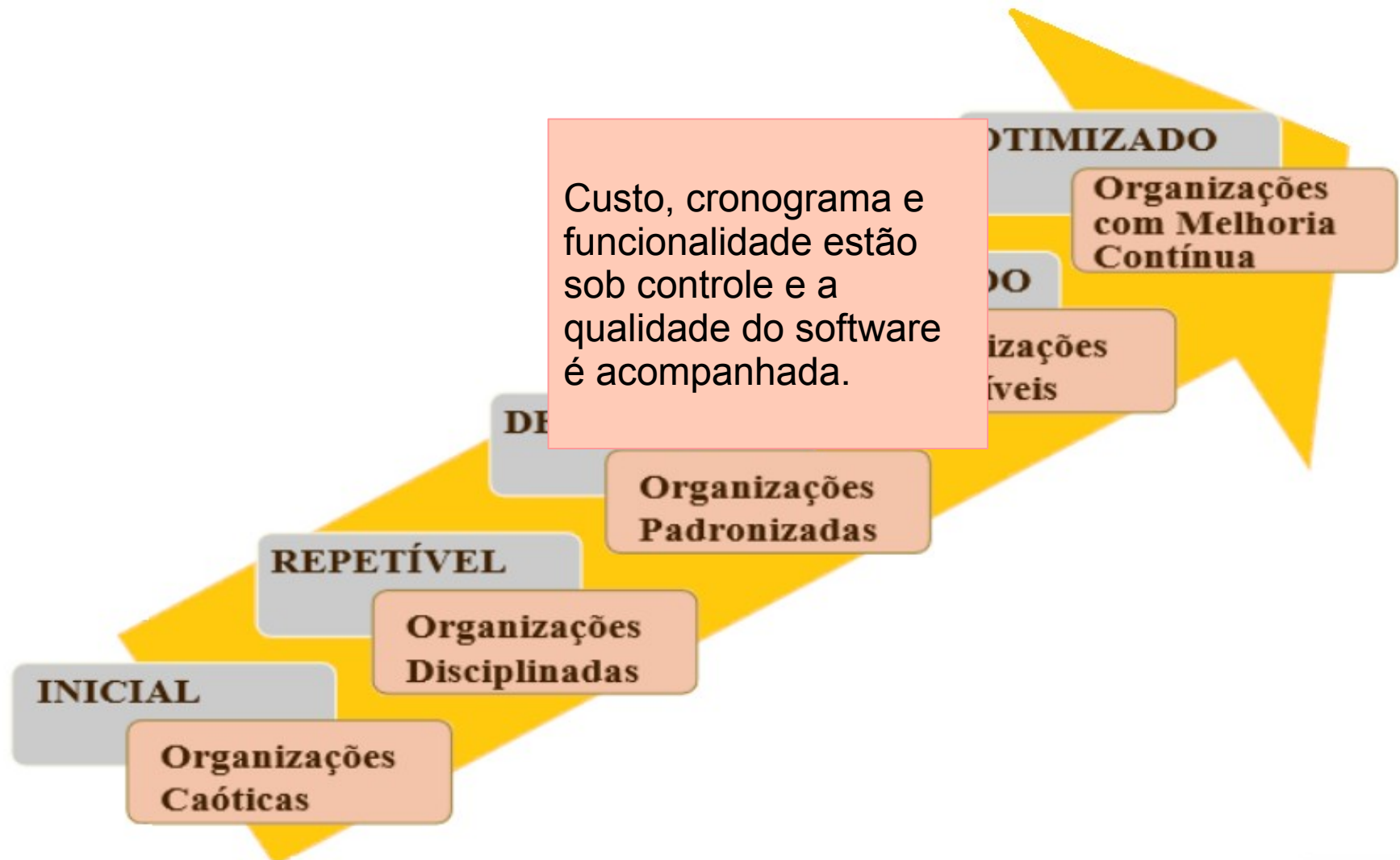
Os 5 Níveis de Maturidades do CMM



Os 5 Níveis de Maturidades do CMM



Os 5 Níveis de Maturidades do CMM



Os 5 Níveis de Maturidades do CMM

Os processos de software, tanto para atividades administrativas quanto para de engenharia de software estão documentados, padronizados e integrados em um processo de software padrão para a organização.

Todos os projetos usam uma versão aprovada do processo de software padrão da organização para desenvolvimento e manutenção de software.



Os 5 Níveis de Maturidades do CMM



Os 5 Níveis de Maturidades do CMM



Nível 1 - Inicial

- não existe repetibilidade de processos
- em geral, não se cumprem prazos
- os projetos custam mais do que o previsto
- depende-se de habilidades pessoais
- as qualidades, procedimentos e o conhecimento pertencem às pessoas

Nível 2 - Repetível

- políticas e procedimentos para gerenciar o desenvolvimento de sw existem e são obedecidas.
- planejamento formal de novos projetos (baseados em experiência anterior)
- os compromissos são assumidos com bases realistas na experiência e nos requisitos documentados
- o desenvolvimento é acompanhado e os planos revisados quanto aos prazos, custos, estimativas e funcionalidades

Nível 2 - Repetível

- existem mecanismos formais para corrigir desvios
- a gestão de requisitos é formalizada para se assegurar que as expectativas do cliente estão sendo atendidas
- controla-se o relacionamento com os sub-contratados
- não há entretanto padronização, o gerenciamento é feito pro projeto e os processo podem ser repetidos com resultados previsíveis.

Nível 3 - Definido

- processos utilizados são padronizados em toda a organização.
- processos técnicos são considerados paralelamente aos gerencias
- SEPG – para criar os processos da organização e assegurar treinamento

Nível 4 - Gerenciado

- são estabelecidas métricas quantitativas para seus produtos e processos.
- medidas são coletadas, avaliadas e analisadas continuamente.
- controle estatístico do processo: a gestão agora é feita com base quantitativas.

Nível 5 – Em otimização

- a organização está engajada na melhoria contínua de seus projeto e também na otimização
- é capaz de agir de forma preventiva
- reduz-se o retrabalho e desperdício
- utilização controlada e disciplinada da inovação nos métodos e tecnologias utilizadas
- melhorias em processos e tecnologias são planejadas e executadas como atividade de rotina

Pessoas e os níveis

1. O sucesso depende de indivíduos e heróis, relacionamento descoordenado e as vezes conflitantes
2. O sucesso ainda depende de indivíduos, mas existe apoio gerencial e treinamento para algumas funções
3. Grupos trabalham de maneira coordenada, treinamento é planejado
4. Forte sentimento de trabalho em equipe
5. Todos estão engajados no programa de melhoria contínua.

Tecnologia e os níveis

- a introdução de novas é arriscada
- atividades que estão bem definidas facilitam a introdução de novas
- novas tecnologias são avaliadas qualitativamente
- novas tecnologias são avaliadas quantitativamente
- novas tecnologias são planejadas e introduzidas com total controle

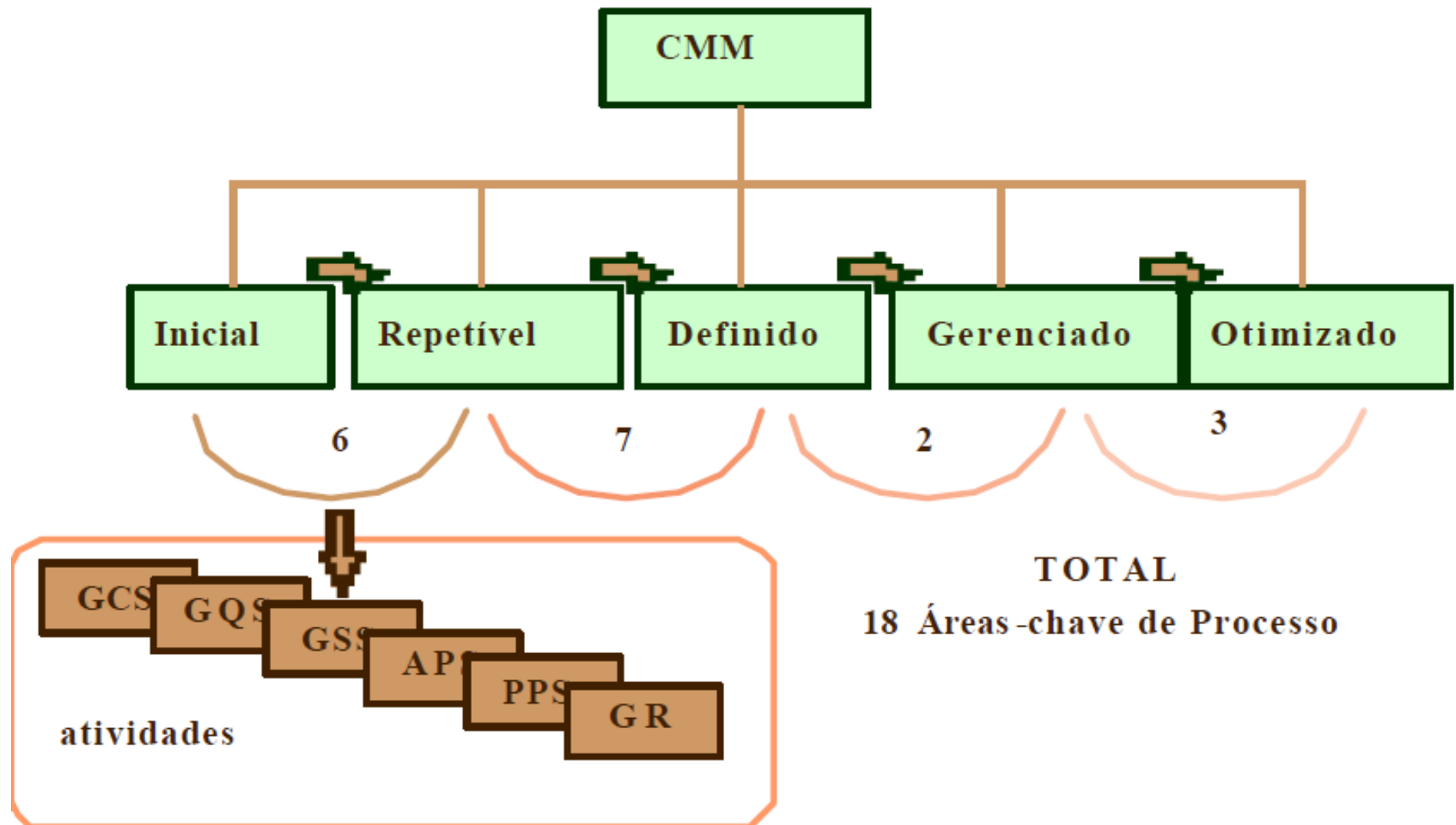
Medidas e os níveis

- a coleta é ad-hoc
- a coleta de dados de atividades de planejamento é feita de maneira sistemática
- coleta sistemática para todos os processos definidos e compartilhada na organização
- coleta padronizada na organização. Dados utilizados para entender os processos de maneira quantitativa e estabilizá-los.
- dados usados para avaliar e selecionar possibilidades de melhoria

Como Melhorar o Processo de Software?

- Realizando um grupo de atividades correlatadas, denominadas ÁREAS-CHAVE DE PROCESSO – (Key process areas – KPA_ que, quando efetuadas coletivamente, alcançam um conjunto de metas consideradas importantes na implementação da competência do processo.

Exemplo: Áreas-chave de Processo



O Modelo CMM

INICIAL (1) => REPETÍVEL (2)

REPETÍVEL

Organizações
Disciplinadas

- 6 – Gerenciamento da Configuração de Software
- 5 – Garantia da Qualidade de Software
- 4 – Gerenciamento de Subcontrato de Software
- 3 – Acompanhamento de Projeto de Software
- 2 – Planejamento de Projeto de Software
- 1 – Gerenciamento de Requisitos

INICIAL

Organizações
Caóticas

O Modelo CMM

INICIAL (1) => REPETÍVEL (2)

REPETÍVEL

Organizações
Disciplinadas

6 – Gerenciamento da Configuração de Software
5 – Garantia da Qualidade de Software

Esta kpa tem o duplo objetivo de assegurar a qualidade tanto do produto sendo construído (o software) quanto do processo de sua construção.

Em termos de produto, isso significa que o CMM exige que a organização defina métodos, técnicas e padrões para verificação da qualidade, estabelecendo, por exemplo, quais os **tipos de teste** que deverão ser realizados, o uso de **revisões**, **inspeções** etc.

Já em relação ao processo, esta kpa procura garantir que todas as outras kpas estejam sendo respeitadas, garantindo, por exemplo, que o gerente de projetos esteja fazendo estimativas corretamente usando o modelo definido para a organização.

O Modelo CMM

REPETÍVEL (2) => DEFINIDO (3)

DEFINIDO

Organizações
Padronizadas

- 7 – Revisões (peer review)
- 6 – Coordenação Intergrupos
- 5 – Engenharia de Produto de Software
- 4 – Gerenciamento de Software Integrado
- 3 – Programa de Treinamento
- 2 – Definição do Processo da Organização
- 1 – Foco no Processo da Organização

REPETÍVEL

Organizações
Disciplinadas

O Modelo CMM

REPETÍVEL (2) => DEFINIDO (3)

DEFINIDO

Organizações
Padronizadas

7 – Revisões (peer review)

6 – Coordenação Intergrupos

5 – Engenharia de Produto de Software

4 – Esta etapa procura garantir que o processo de

3 – software definido seja executado de modo

2 – que sejam produzidos produtos corretos e

1 – consistentes de forma eficaz e eficiente.

REPETÍVEL

Organizações
Disciplinadas

O Modelo CMM

DEFINIDO(3) => GERENCIADO (4)

GERENCIADO

Organizações
Gerenciadas

2 – Gereciamento quantitativo do processo
1 – Gerenciamento da qualidade de software

DEFINIDO

Organizações
Padronizadas

O Modelo CMM

GERENCIADO(4) => EM OTIMIZAÇÃO(5)

EM OTIMIZAÇÃO

Organizações
Em otimização

- 3 – Prevenção de defeitos
- 2 – Gerenciamento de mudanças na tecnologia
- 1 – Gerenciamento de mudanças no processo

GERENCIADO

Organizações
Gerenciadas

Classificação para as Práticas

Comprometimento – ações da alta gerência da organização

Habilidades- refere-se à capacitação das pessoas e a disponibilidade de recursos humanos e materiais

Atividades: atividades e tarefas

Medidas – monitoram a situação, usadas para controle e melhoria.

Verificação- para comparar o que é realizado com o estabelecido nos procedimentos e planos

Melhoria de processos e avaliações

- Saber em que situação a empresa encontra-se e em qual situação quer estar
- Traçar um plano de ação
- CMM modelo de referência – roteiro para melhoria

Tipos de avaliação - CMM

Software Process Assessment – SPA –
determina a situação atual dos processos de sw
para fins de melhoria

Software Capability Evaluation – avaliar os
subcontratados ou potenciais fornecedores

SPA -Avaliação para melhoria

- definição dos objetivos e comprometimento dos gerentes
- formação da equipe de avaliação, de 6 a 10, alguém licenciado pelo SEI –líder
- administração de um questionário de avaliação de maturidade
- atividades de análise de documentos, relatórios, entrevistas

Exercício

Identifique o nível do CMM no qual as empresas A e B se encontram.

* Empresa A: a introdução de novas tecnologias é arriscada, podendo gerar resultados catastróficos.

* Empresa B: novas tecnologias são avaliadas quantitativamente.

ISO 12207

- ISO/IEC 12207 – *Information Technology Software Life-Cycle Processes*.
- NBR ISO/IEC 12207 – Tecnologia da Informação – Processos de Ciclo de Vida de Software.
- Framework para processos de ciclo de vida com terminologia bem definida.
- Cobre todo o ciclo de vida de desenvolvimento do software desde a concepção das ideias iniciais sobre o software até este ser retirado de execução.

Processos Fundamentais

Aquisição

Fornecimento

Desenvolvimento

Operação

Manutenção

Processos de Apoio

Documentação

Gerência de Configuração

Garantia da Qualidade

Verificação

Validação

Revisão Conjunta

Auditoria

Resolução de Problema

Processos Organizacionais

Gerência

Melhoria

Infraestrutura

Treinamento

Adaptação

ISO 12207: Processos

Os processos são agrupados em três classes de acordo com suas naturezas [ISO 12207]:

- Processos Fundamentais
- Processos de Apoio
- Processos Organizacionais

ISO 12207: Processos

- Processos Fundamentais: englobam as atividades referentes ao contrato entre o cliente e o fornecedor, ao desenvolvimento, à operação e à manutenção do software.
- Processos de Apoio: auxiliam outros processos como parte integrante dos mesmos, com propósitos distintos contribuindo para a qualidade do projeto de software.

ISO 12207: Processos

- Processos Organizacionais: estabelecem e implementam uma estrutura que forneça uma base aos processos e ao pessoal envolvido no desenvolvimento de software.
- Processos de Adaptação: definem as atividades necessárias para adaptar a norma a uma organização ou projeto específico.

ISO 12207: Atividades dos Processos Fundamentais

Processo	Atividades
Aquisição	Iniciação Preparação do Pedido de Proposta Preparação e Atualização do Contrato Monitoramento do Fornecedor Aceitação e Conclusão
Fornecimento	Iniciação Preparação de Resposta Contrato Planejamento Execução e Controle Revisão e Avaliação Entrega e Conclusão

ISO 12207: Atividades dos Processos Fundamentais

Processo	Atividades
Desenvolvimento	Implementação do Processo Análise dos Requisitos do Sistema Projeto da Arquitetura do Sistema Análise dos Requisitos do Software Projeto Arquitetural do Software Projeto Detalhado do Software Codificação e Testes do software Integração do Software Testes de Qualificação do Software Integração do Sistema Teste de Qualificação do Sistema Instalação do Software Suporte à Aceitação do Software

ISO 12207: Atividades dos Processos Fundamentais

Processo	Atividades
Operação	Implementação do Processo Teste Operacional Operação do Sistema Suporte ao Usuário
Manutenção	Implementação do Processo Análise do Problema e da Modificação Implementação da Modificação Revisão/Aceitação da Manutenção Migração Descontinuação do Software

ISO 12207: Atividades dos Processos de Apoio

Processo	Atividades
Documentação	Implementação do Processo Projeto e Desenvolvimento Produção Manutenção
Gerência de Configuração	Implementação do Processo Identificação da Configuração Controle da Configuração Relato da Situação Avaliação da Configuração Gerência de Liberação e Distribuição

ISO 12207: Atividades dos Processos de Apoio

Processo	Atividades
Garantia da Qualidade	Implementação do Processo Garantia do Produto Garantia do Processo Sistema de Garantia da Qualidade
Verificação	Implementação do Processo Verificação
Validação	Implementação do Processo Validação

ISO 12207: Atividades dos Processos Organizacionais

Processo	Atividades
Gerência	Iniciação e Definição do Escopo Planejamento Execução e Controle Revisão e Avaliação Conclusão
Infraestrutura	Implementação do Processo Estabelecimento da Infraestrutura Manutenção da Infraestrutura

ISO 12207: Atividades dos Processos Organizacionais

Processo	Atividades
Melhoria	Estabelecimento do Processo Avaliação do Processo Melhoria do Processo
Treinamento	Implementação do Processo Desenvolvimento do Material de Treinamento Implementação do Plano de Treinamento

SPICE

- 1993: SPICE (*Software Process Improvement and Capability Determination*)
- *ISSO/ IEC 15504*
- Objetivo: gerar normas para avaliação de processos de software, visando a melhoria contínua do processo e a determinação da sua capacidade.

Benefícios do SPICE

Para avaliadores:

- Uma estrutura que define todos os aspectos para conduzir avaliações.

Benefícios do SPICE

Para fornecedores:

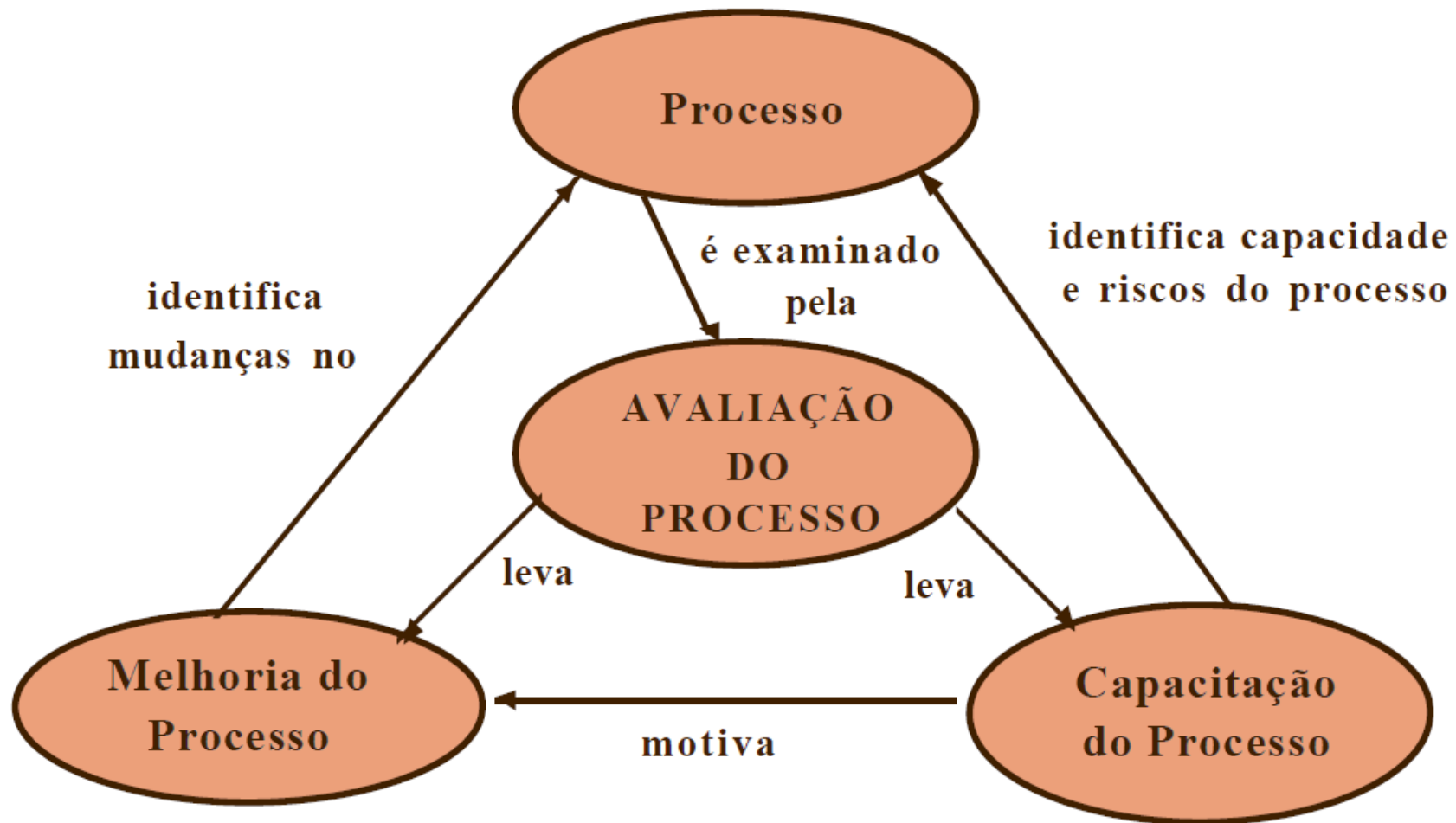
- Determinar a capacidade atual e potencial de seus processos de software.
- Efetuar a melhoria ou uma estrutura que define um roteiro para a melhoria do processo de software.

Benefícios do SPICE

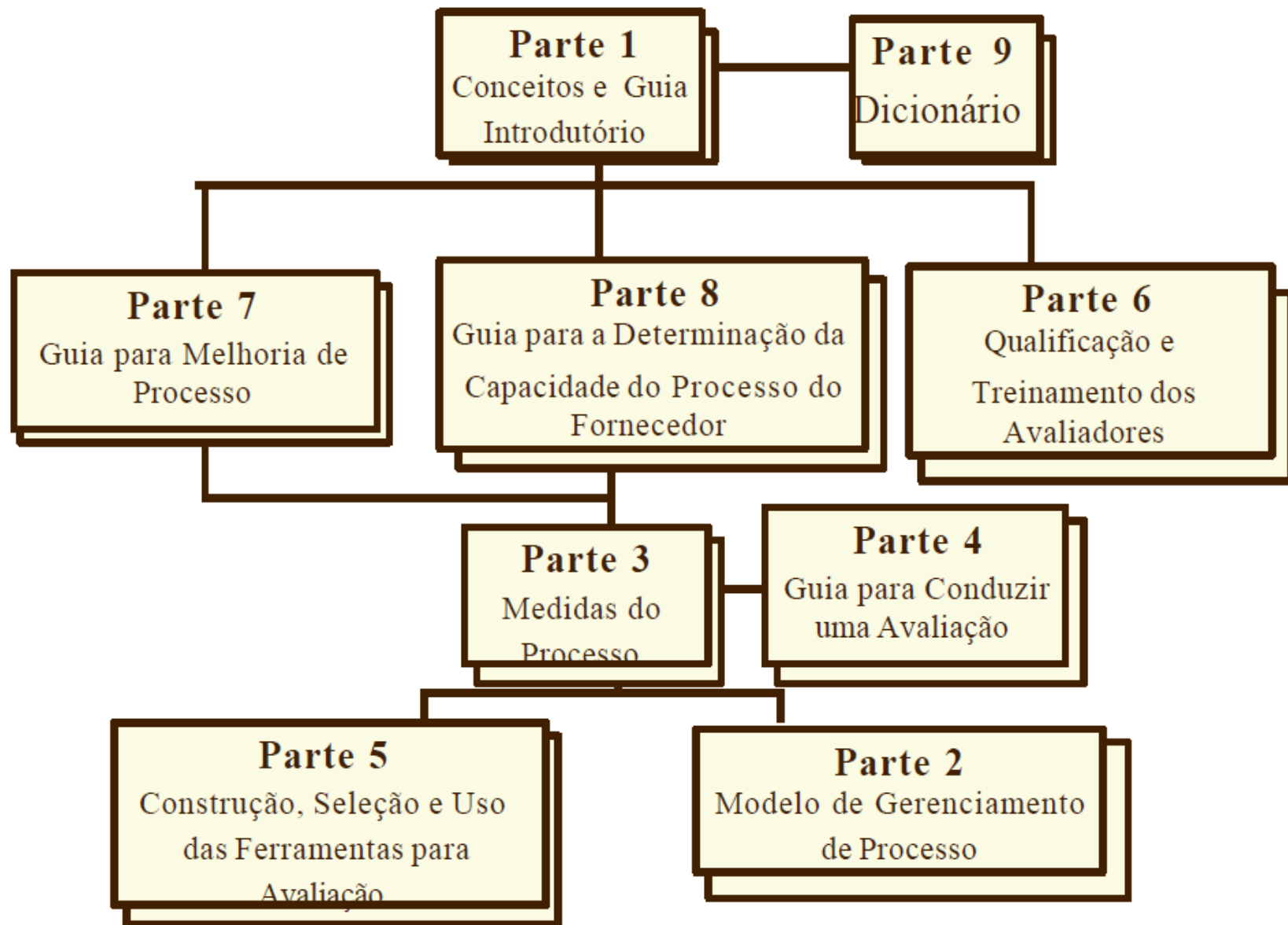
Para os compradores:

- Determinar a capacidade atual e potencial de um processo de software do fornecedor.

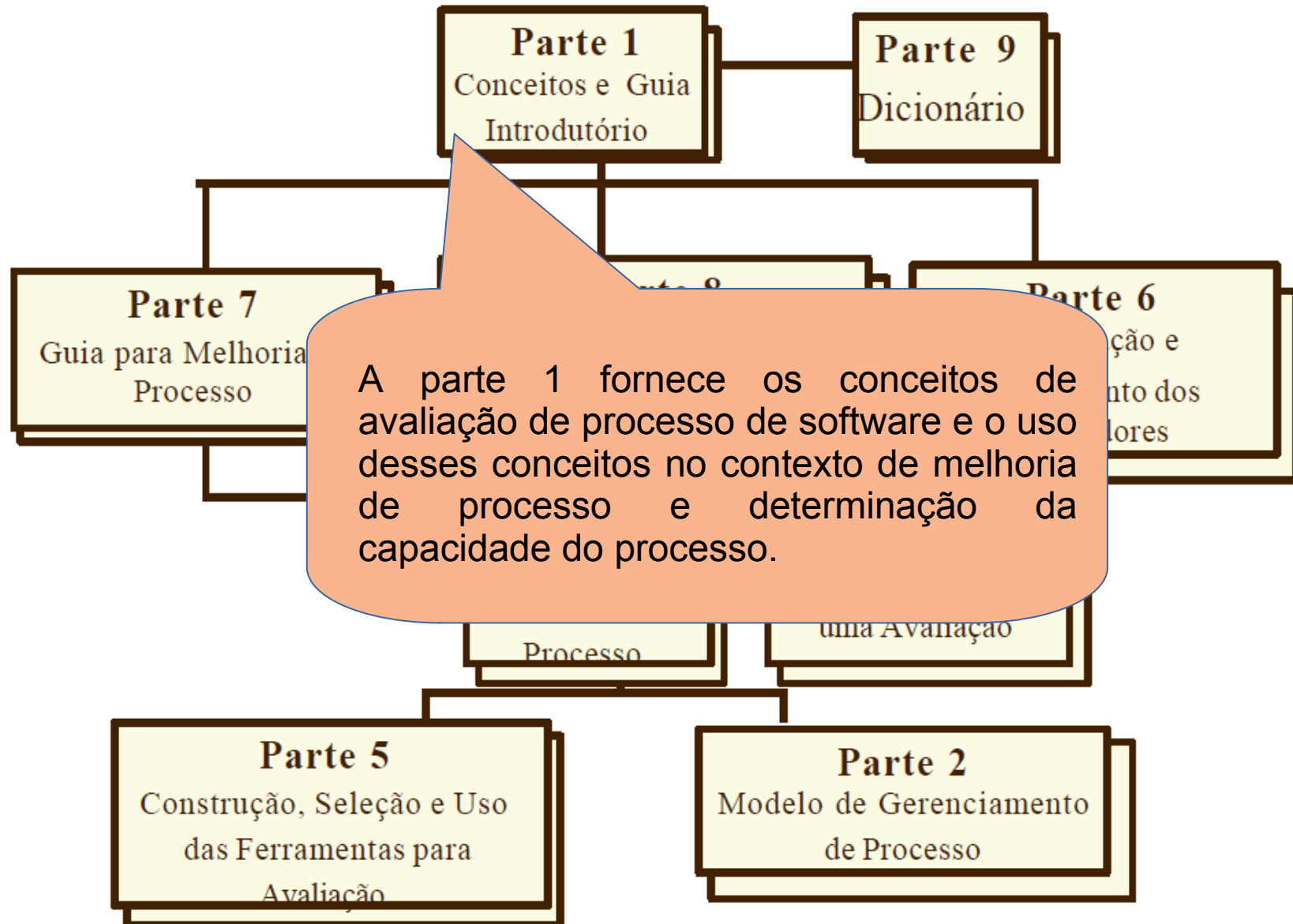
Processo de Avaliação



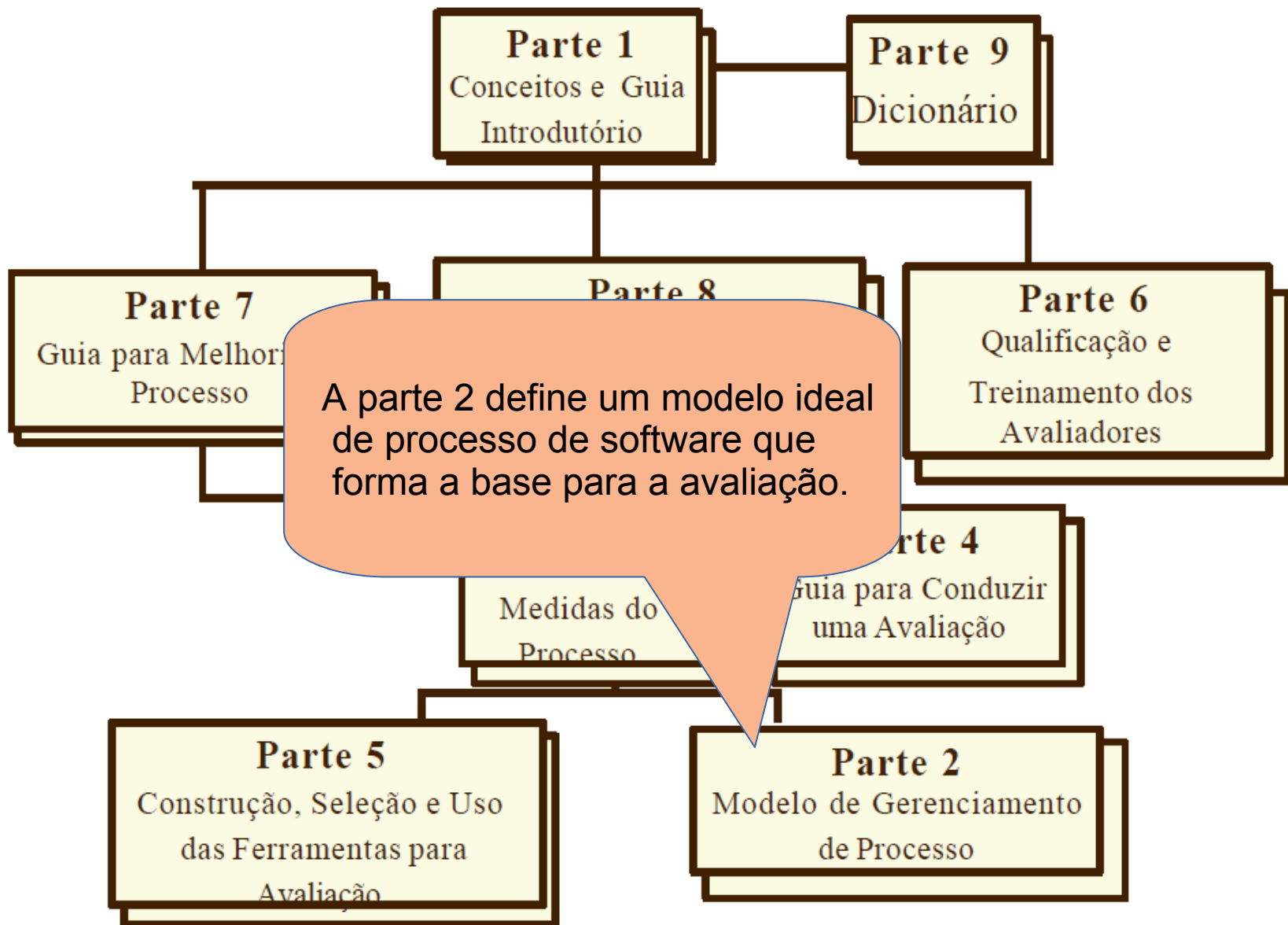
Componentes do Modelo SPICE



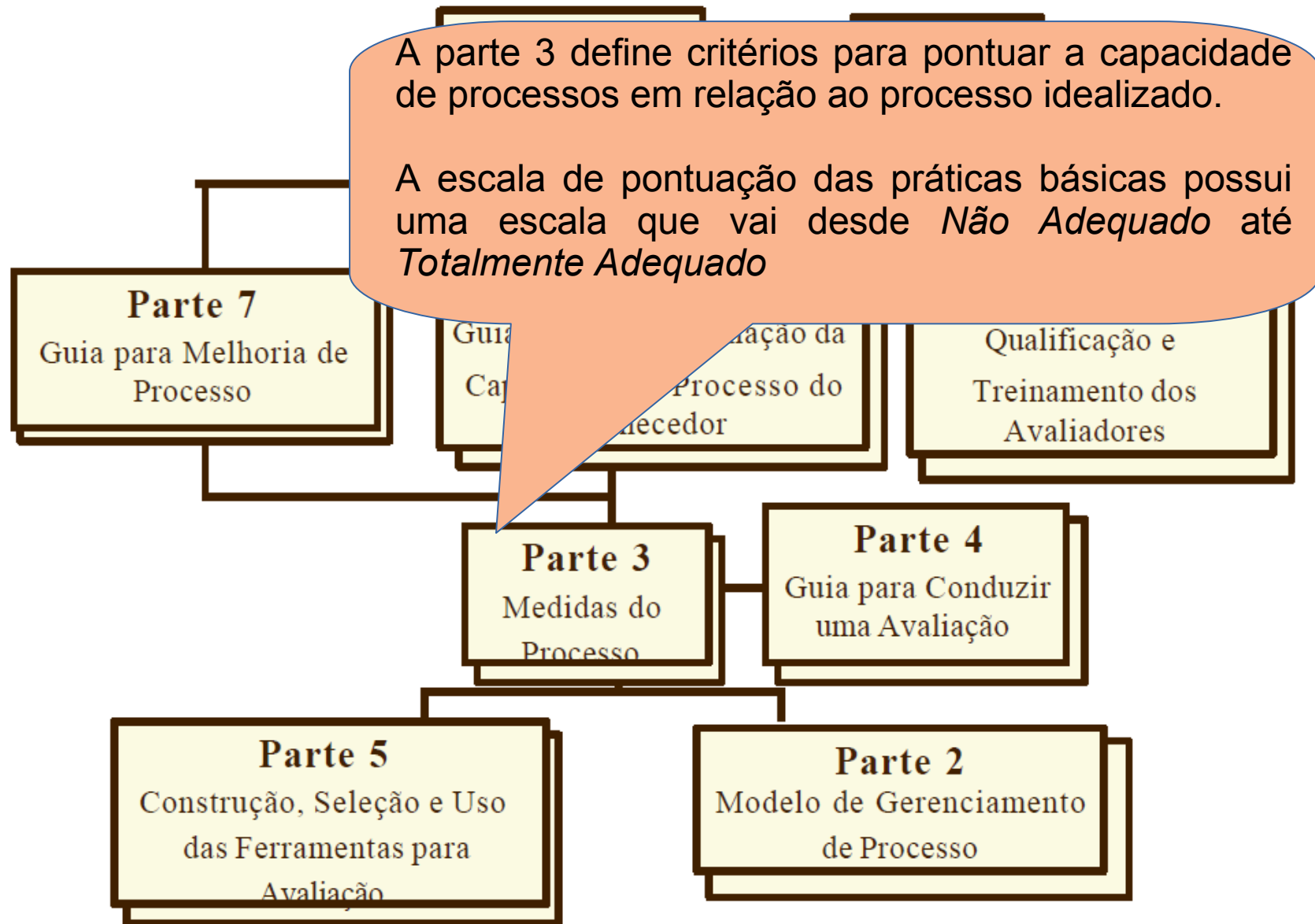
Componentes do Modelo SPICE



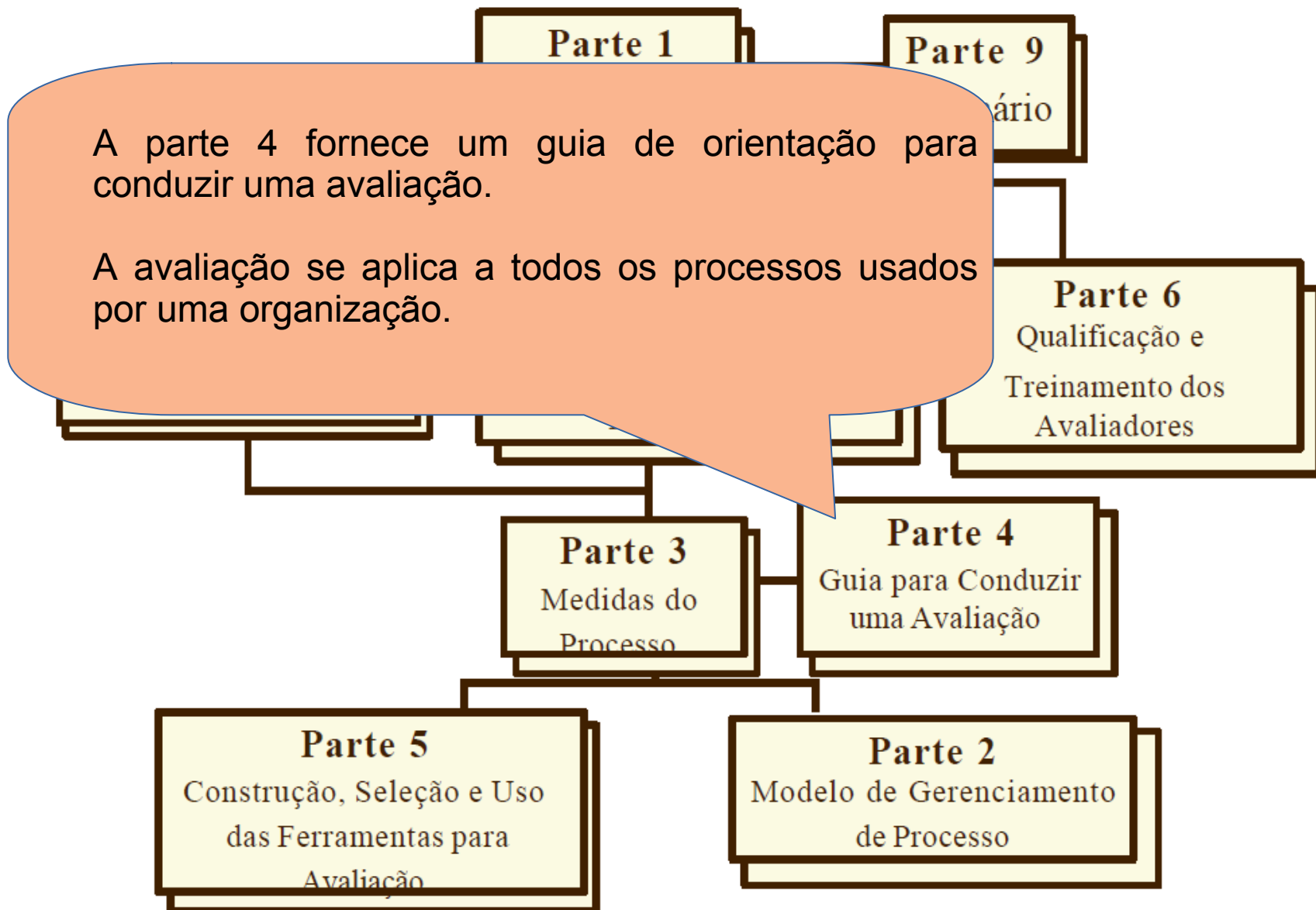
Componentes do Modelo SPICE



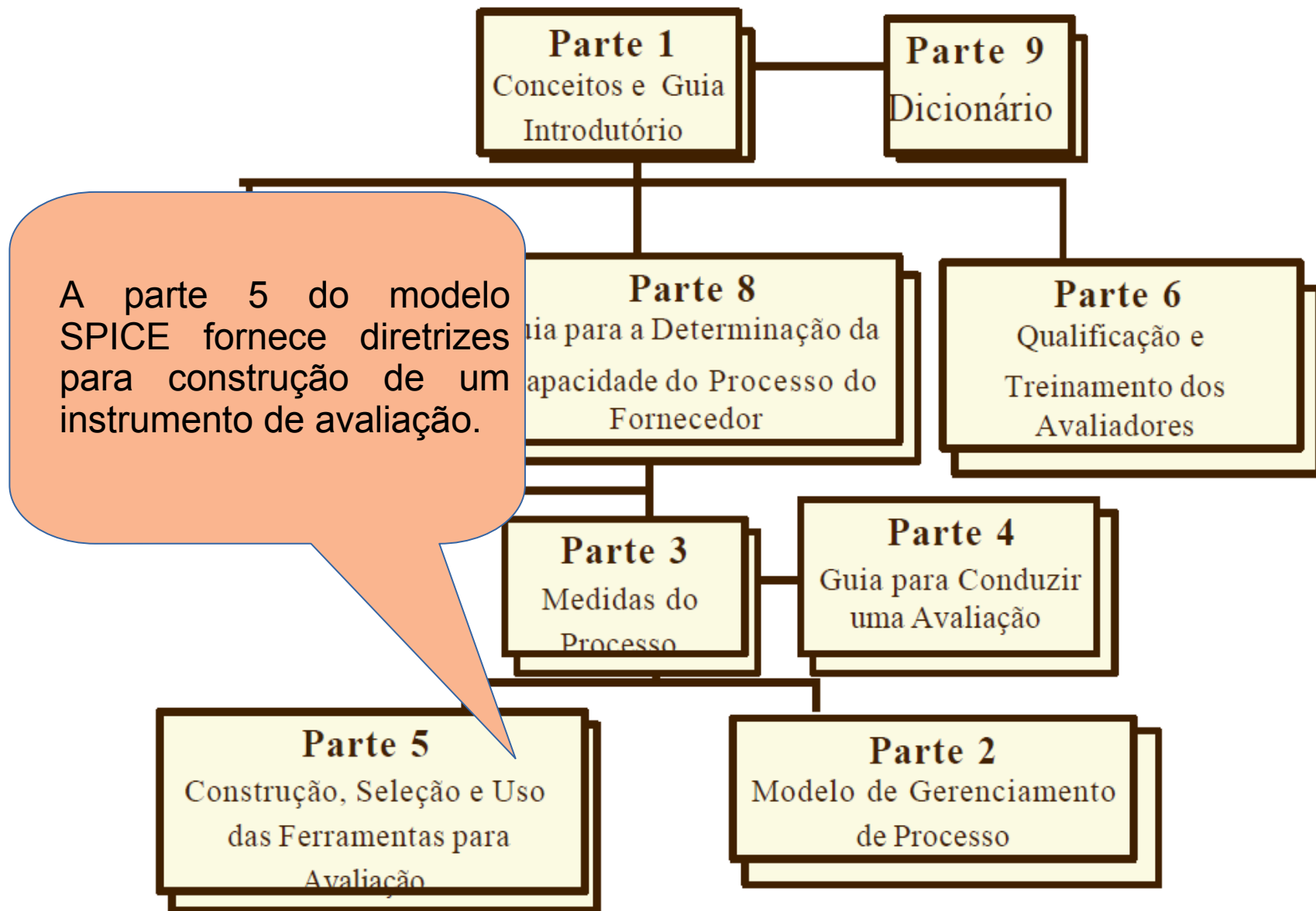
Componentes do Modelo SPICE



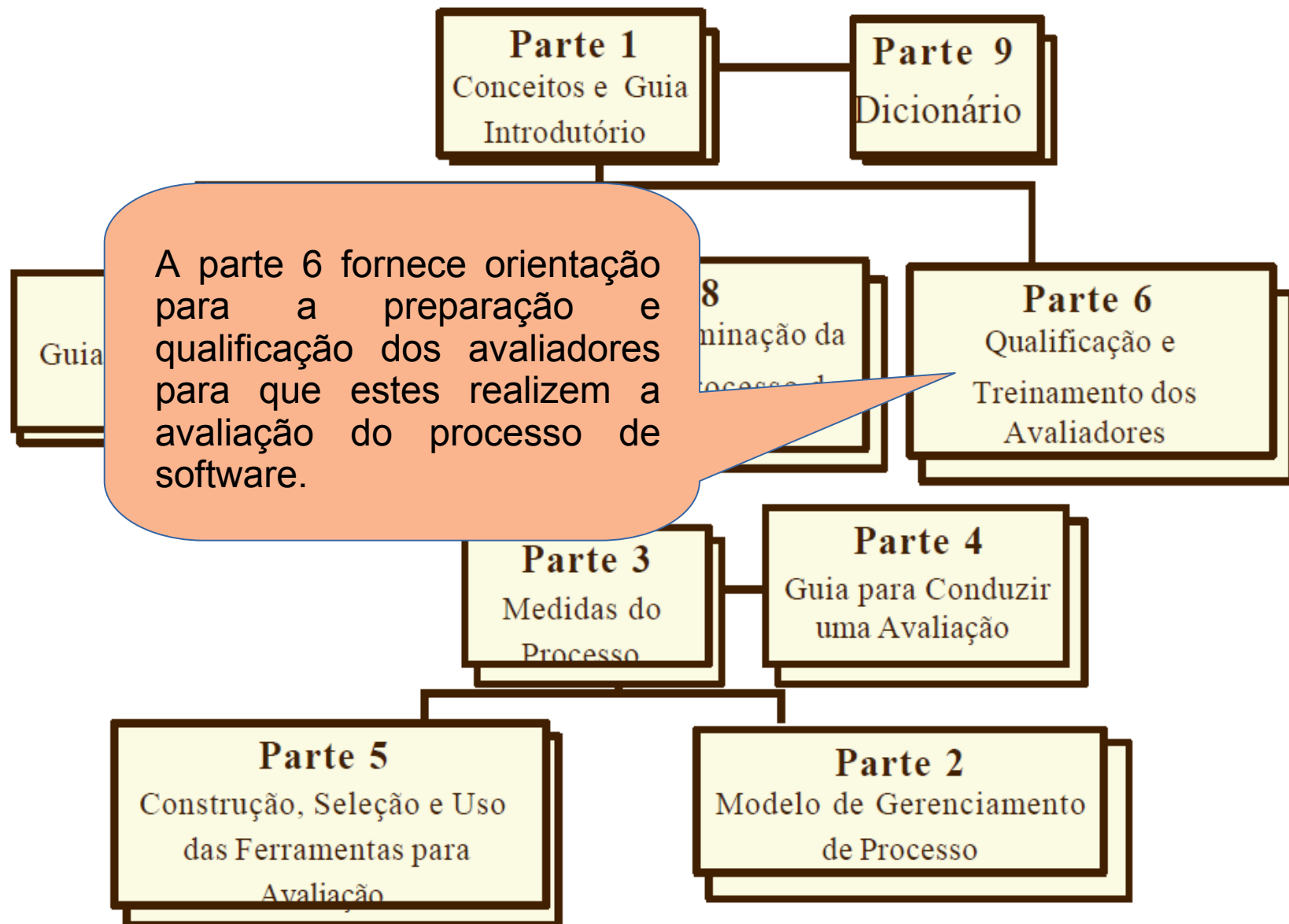
Componentes do Modelo SPICE



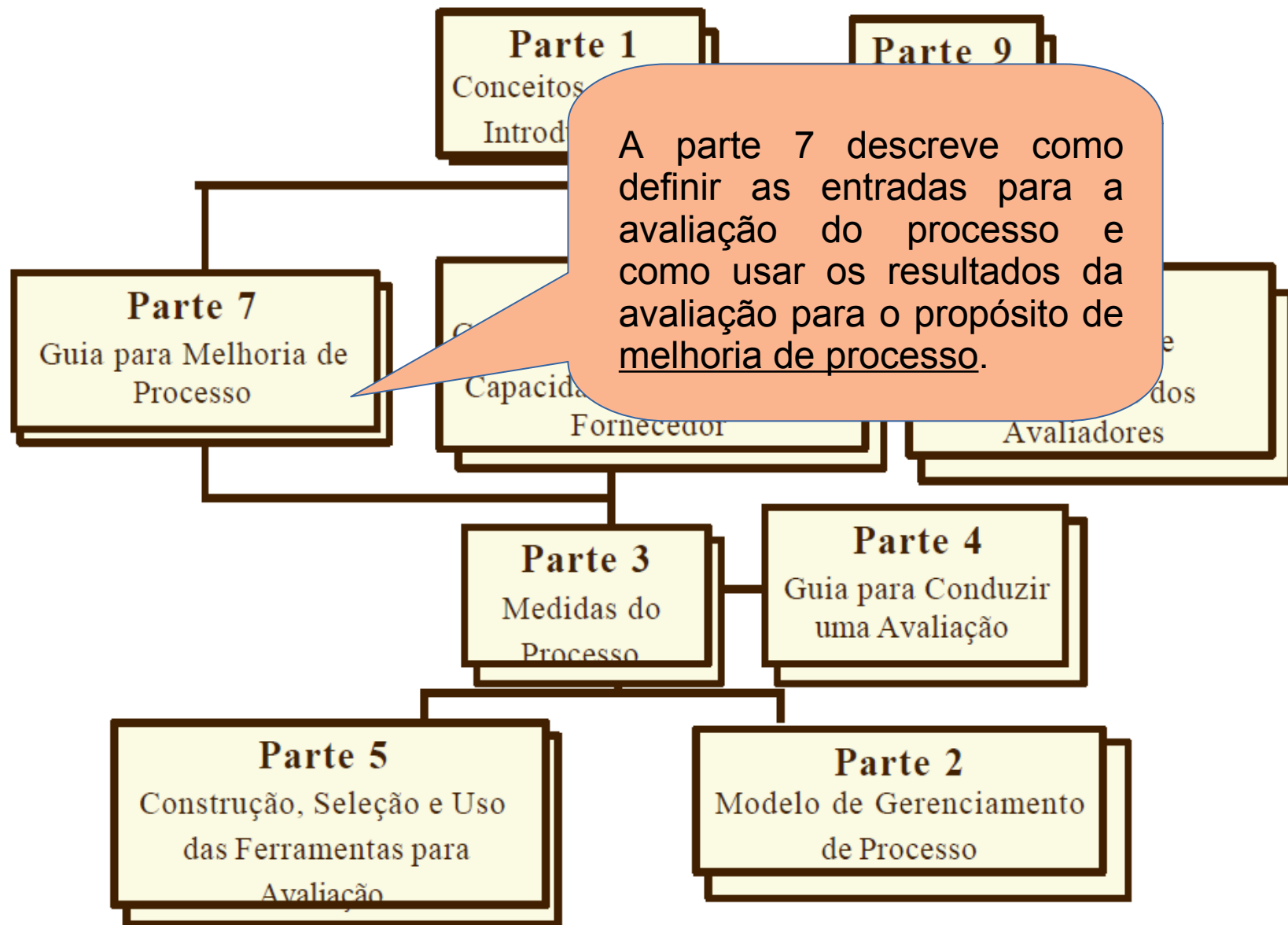
Componentes do Modelo SPICE



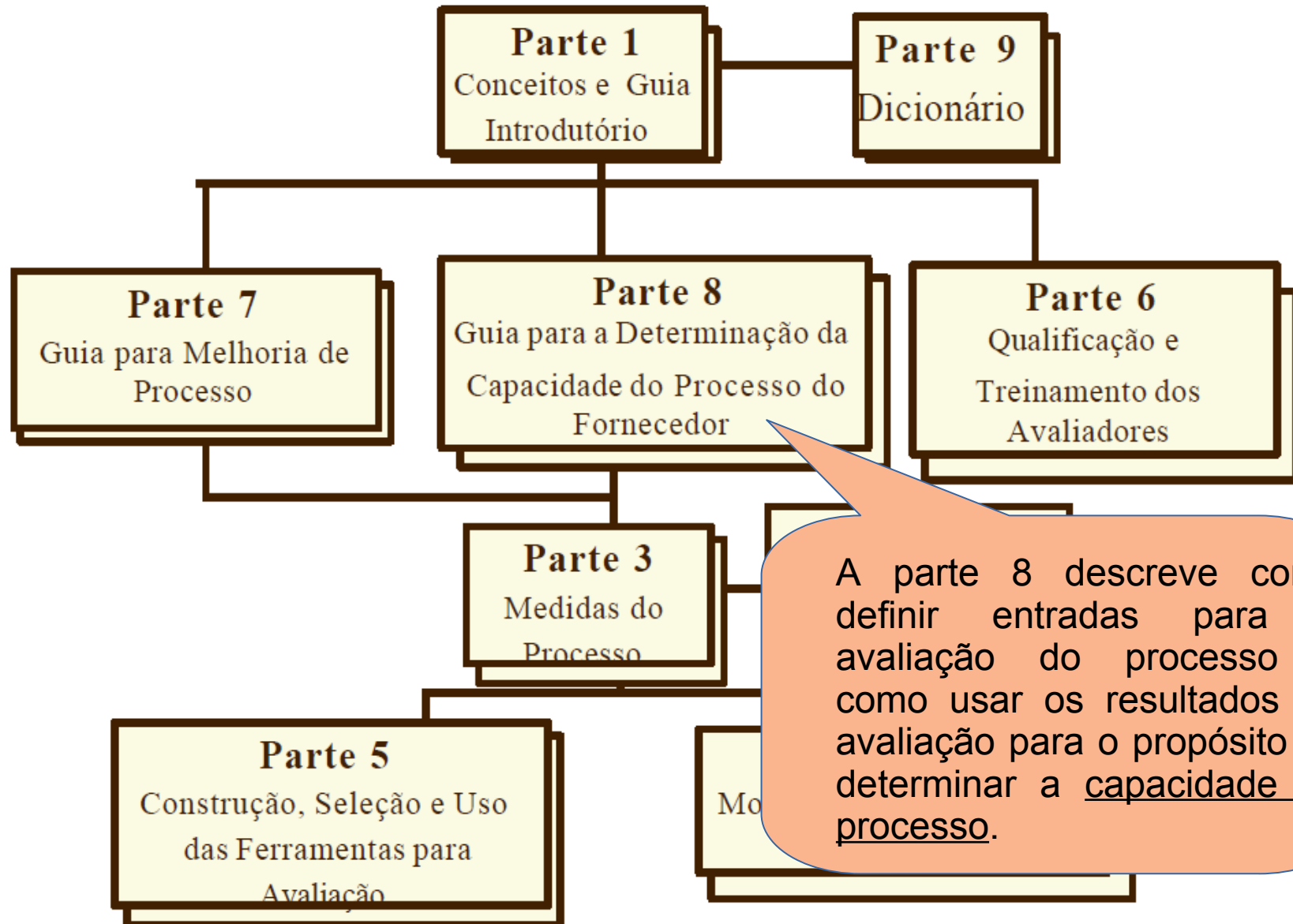
Componentes do Modelo SPICE



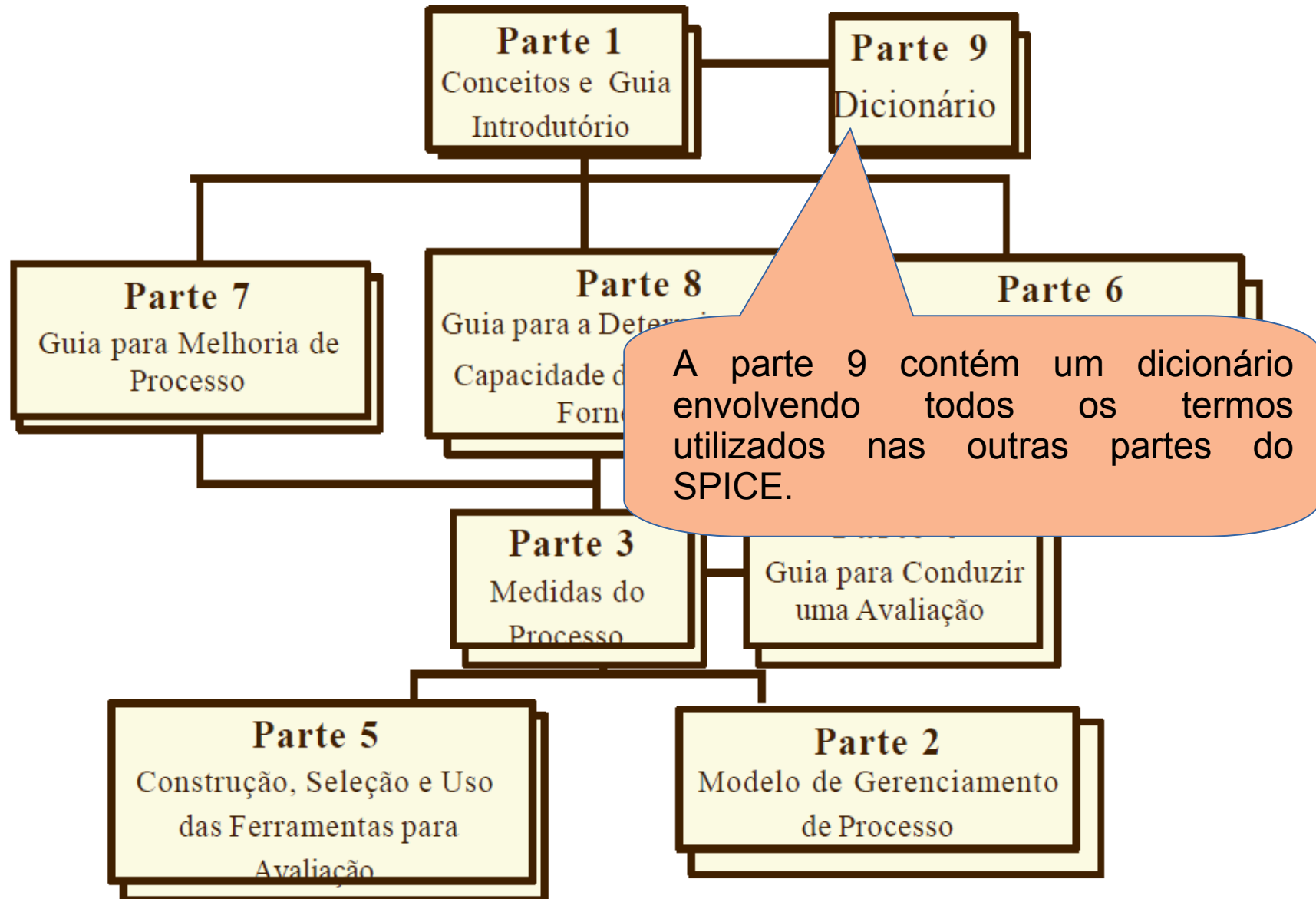
Componentes do Modelo SPICE



Componentes do Modelo SPICE



Componentes do Modelo SPICE



Dimensão de capacidade de processo – Níveis

Nível 0: processo incompleto

Processo não implementado

Processo não gera os produtos esperados

Não há atributos de processo

Dimensão de capacidade de processo – Níveis

Nível 1: processo executado

Processo consegue alcançar os objetivos de alguma forma

Processo gera os produtos esperados

Atributo:

PA 1.1 → atributo de execução de processo: transformação de produtos de entrada em produtos de saída e os resultados esperados do processo são alcançados

Dimensão de capacidade de processo – Níveis

Nível 2: processo gerenciado

Processo executado de modo planejado, controlado, acompanhado, verificado e corrigido

Atributos:

PA 2.1 → atributo de execução de processo: mede até que ponto o processo é gerenciado para produzir os produtos que satisfazem ao seu objetivo

Características: objetivos do processo em termos de qualidade, prazo e uso de recursos identificados, atribuição de responsabilidade pelo produto de trabalho feita, execução do processo é gerenciada

Dimensão de capacidade de processo – Níveis

Nível 2: processo gerenciado

Atributos:

PA 2.2 → atributo de gestão dos produtos de trabalho: mede até que ponto os produtos de trabalho são documentados, controlados e verificados

Características: requisitos dos produtos de trabalho documentados, requisitos para documentação e controle de produtos de trabalho definidos, dependências entre produtos de trabalho estão definidas, os produtos de trabalho têm suas mudanças controladas e são distribuídos em *baselines*, os produtos são verificados

Dimensão de capacidade de processo – Níveis

Nível 3: processo estabelecido

Processo executado e gerenciado, é definido com base em princípios de engenharia de software

Atributos:

PA 3.1 → atributo de definição de processo: mede até que ponto o processo é definido com base em um processo padronizado

Características: existe um processo padronizado na empresa, no qual as adaptações podem ser feitas; o processo é executado conforme um processo padronizado ou adaptado de acordo com orientações para adaptações; dados históricos do processo são coletados para auxiliar na compreensão do comportamento estatístico do processo; a experiência acumulada é usada para refinar o processo

Dimensão de capacidade de processo – Níveis

Nível 3: processo estabelecido

Atributos:

PA 3.2 → atributo de recursos de processo: mede até que ponto o processo faz uso de recursos humanos e materiais para ser executado com sucesso

Características: os papéis, responsabilidades e competências necessárias para execução do processo são identificados e documentados; a infra-estrutura necessária para a execução do processo é identificada e documentada; os recursos necessários são alocados e utilizados na execução do processo

Dimensão de capacidade de processo – Níveis

Nível 4: processo previsível

Processo executado, gerenciado e definido, é executado dentro de limites quantitativos bem definidos

Atributos:

PA 4.1 → atributo de medida: mede até que ponto métricas e objetivos de processo/produto são usados para assegurar que a execução do processo é efetiva, alcançando os objetivos de negócio da empresa

Características: métricas são identificadas e coletadas; as tendências observadas são analisadas; a capacidade de processo é medida

Dimensão de capacidade de processo – Níveis

Nível 4: processo previsível

Atributos:

PA 4.2 → atributo de controle de processo: mede até que ponto o processo é controlado por intermédio da coleta, análise e uso de medidas para servir de base para ações corretivas, quando necessário

Características: técnicas de medida são estabelecidas; características de produto e processo são medidas e usadas como insumo para permitir o controle de processo dentro de limites de variabilidade; o processo é gerenciado de forma quantitativa

Dimensão de capacidade de processo – Níveis

Nível 5: processo em otimização

Processo executado, gerenciado, definido e executado dentro de limites quantitativos, pode ser mudado e evoluído de maneira dinâmica e sob controle

Atributos:

PA 5.1 → atributo de mudança de processo: mede até que ponto mudanças na definição, gerência e execução do processo são controladas

Características: impacto de mudança *versus* objetivos do processo e de seus produtos é avaliado; mudanças realizadas de forma controlada e prevenida; eficácia das mudanças é medida em relação ao planejado

Dimensão de capacidade de processo – Níveis

Nível 5: processo em otimização

Atributos:

PA 5.2 → atributo de melhoria contínua: mede até que ponto as mudanças de processo contribuem para melhoria contínua

Características: metas de melhoria de processo são estabelecidas conforme objetivos de negócio; possíveis fontes de risco e de problemas são identificadas; estratégia de melhoria contínua é estabelecida

Mecanismos de Pontuação

Baseia-se na verificação do grau de satisfação dos atributos de processo

Escala ordenada de quatro valores:

N (*not achieved*) ou não atendido: 0% a 15%

P (*partially achieved*) ou parcialmente atendido: 16% a 50%

L (*largely achieved*) ou largamente atendido: 51% a 85%

F (*fully achieved*) ou totalmente atendido: 86% a 100%

Nível de capacidade

Processo pontuado em determinado nível se todos os atributos de processo dos níveis inferiores estiverem totalmente atendidos e que os atributos de processo desse nível sejam pelo menos largamente atendidos

Nível de capacidade

Exemplo:

PA 1.1 – F PA 3.2 – P

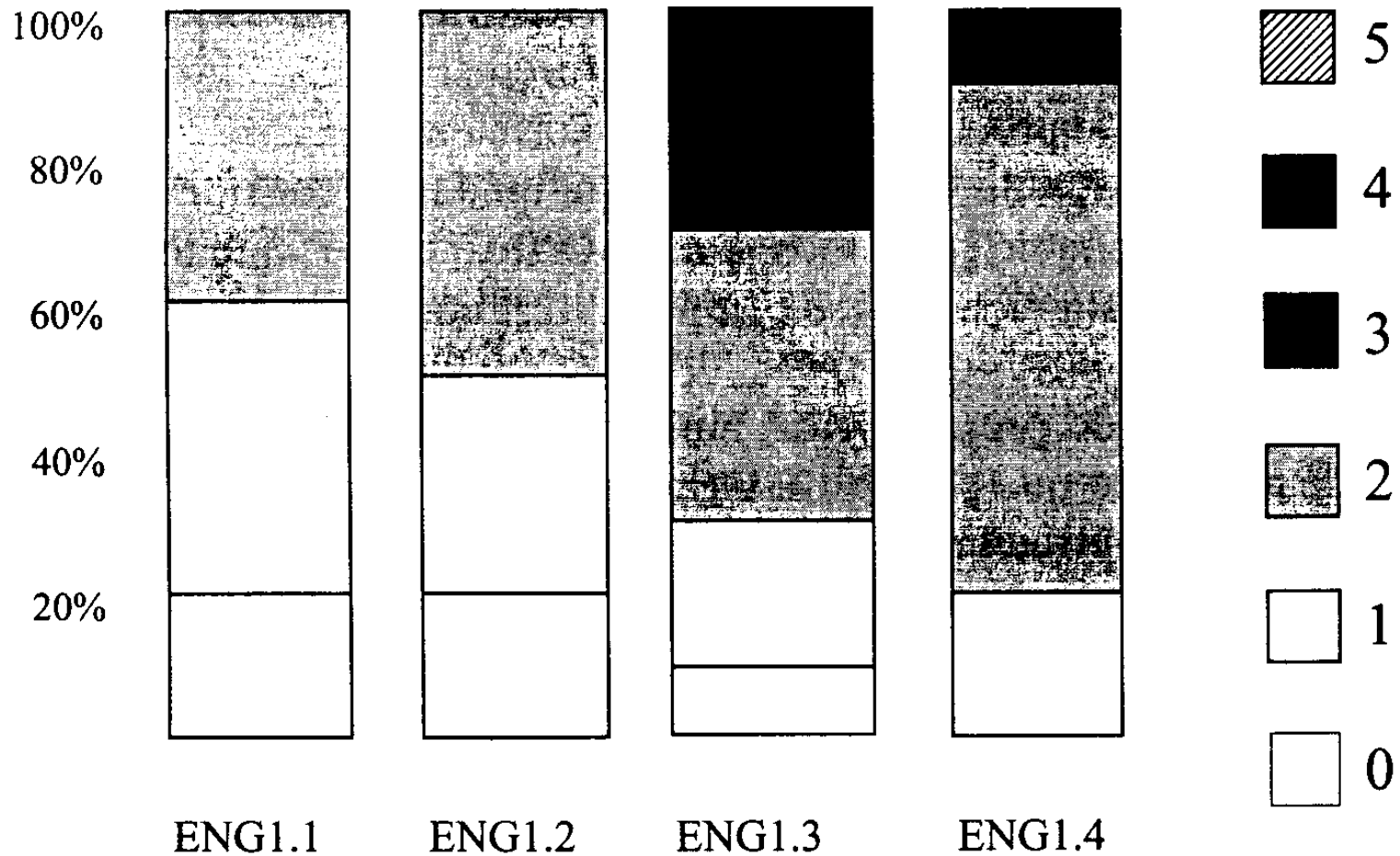
PA 2.1 – F PA 4.1 – L

PA 2.2 – L PA 4.2 – L

PA 3.1 – F PA 5.1 – N

Qual o nível desse processo?

Perfil de nível de capacidade por projeto



Perfil de nível de capacidade por projeto

Situação dos processos ENG1.1, ENG 1.2, ENG 1.3 e ENG 1.4 em vários projetos de uma organização

40% dos projetos têm o processo em ENG 1.3 no nível 2

20% dos projetos têm o processo em ENG 1.4 no nível 1,
70% no nível 2 e 10% no nível 3

PSP: Personal Software Process

Já foram vistas

ISO/IEC 9126 – foco no produto

ISO 9001 e CMM – foco no processo de desenvolvimento

Critica a essas abordagens

Difícil aplicação em pequenas empresas ou no nível individual

PSP: Personal Software Process

Surgimento do PSP

SEI (Software Engineering Institute), por Watts
Humphrey

Objetivo → ajudar as pessoas a serem melhores
engenheiros de software

PSP: Personal Software Process

Visa melhorar, no nível pessoal:

- Capacidade de planejamento

- Acompanhamento

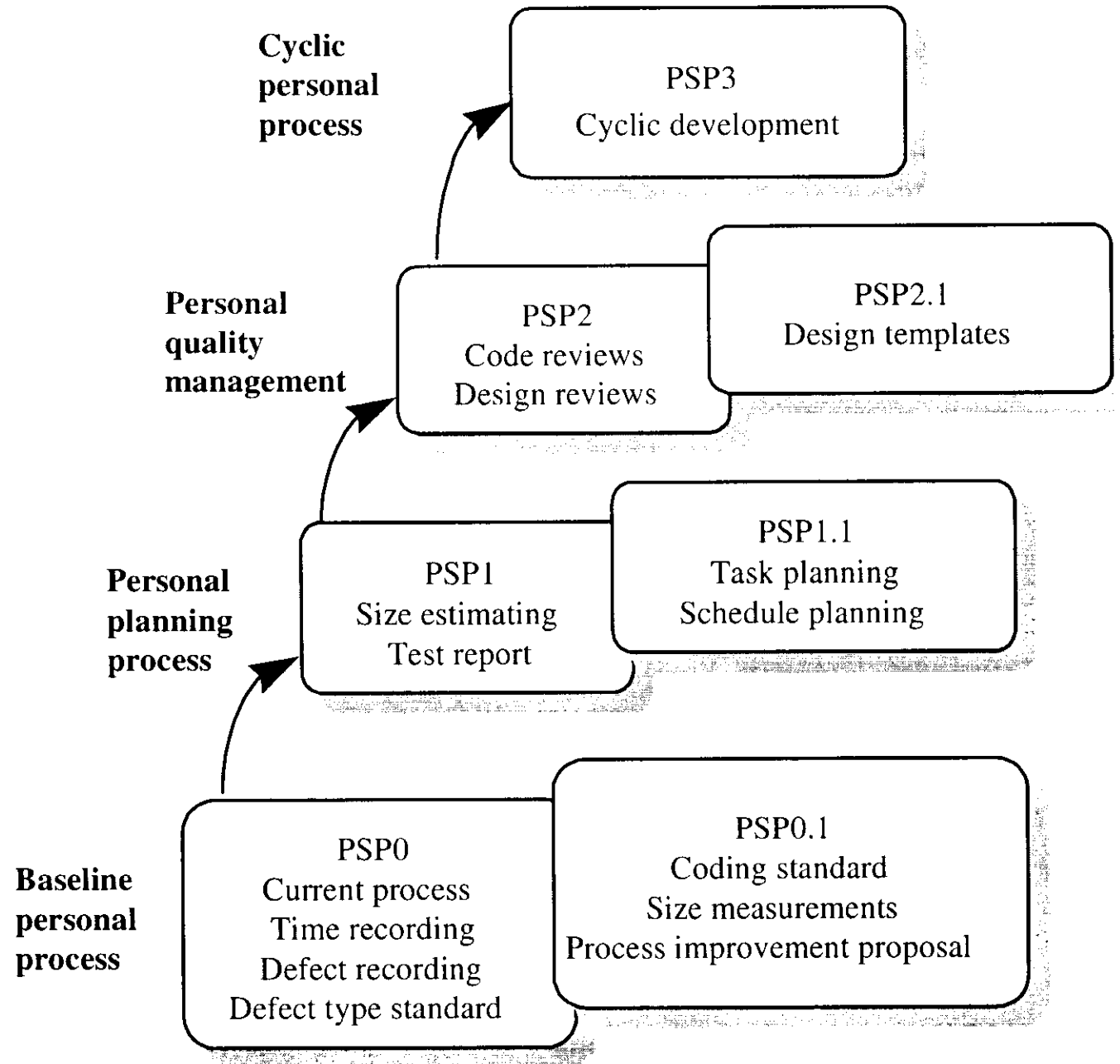
- Qualidade dos resultados

Benefícios

- Melhoria de produtividade

- Melhoria do perfil de qualidade dos produtos

Os 4 níveis do PSP



PSP0: Processo referencial (Baseline process)

Estabelecimento de práticas de medidas

Formatos de relatórios – base para melhoria
contínua pessoal

PSP0: Processo referencial (Baseline process)

PSP0.1:

Padrões de programação

Práticas de medida de tamanho de produto de trabalho

Proposta de melhoria de processo (PIP – Process Improvement Proposal) – registro de problemas nos processos, experiências e propostas de melhoria

PSP1: Processo de planejamento pessoal

Práticas de planejamento

Relatório de teste

Práticas de estimativa de tamanho e recurso

PSP1: Processo de planejamento pessoal

PSP1.1:

Planejamento de tarefas

Elaboração de cronogramas

O planejamento no nível pessoal permite:

- Melhor compreensão do tamanho do programa e tempo gasto no seu desenvolvimento

- Assumir compromissos com certeza de seu cumprimento

- Organizar o trabalho

- Acompanhar melhor o *status* do desenvolvimento

PSP2: Processo de gestão pessoal de qualidade

Gerenciamento dos próprios erros

Técnicas de inspeção e revisão para detecção de defeitos

Coleta e análise de dados de defeitos de compilação e teste detectados em programas anteriores

Avaliação da evolução do nível de qualidade do programador

PSP2: Processo de gestão pessoal de qualidade

PSP2.1:

Processo de design

Auxilia no estabelecimento de critérios de completitude e de técnicas de verificação e consistência

PSP3: Processo pessoal cíclico

Subdividir o programa em módulos

Desenvolvimento incremental

Cada módulo é um ciclo completo de design, codificação e teste (PSP2)

Controle de qualidade de cada módulo

Uso de relatórios

Testes de regressão

Utilização do PSP

Benefícios concretos para os desenvolvedores
[Humphrey 95]:

- Melhor entendimento do trabalho, através da definição, medição e acompanhamento

- Uso de uma estrutura de processos definida e critérios mensuráveis → avaliação de experiências anteriores

- Possibilidade de selecionar métodos e técnicas que melhor se adaptem ao trabalho

- Maior produtividade e eficácia nas equipes de desenvolvimento

Utilização do PSP

Princípios que influenciam a produtividade e qualidade do processo de desenvolvimento

Um processo bem definido e estruturado pode melhorar a eficiência no trabalho

O processo pessoal deve ser ajustado ao conhecimento e preferência de cada um

O desenvolvedor se sente à vontade com o processo se participar de sua definição

Na medida que o conhecimento e habilidade de um profissional evoluem, o processo utilizado deve evoluir

A melhoria contínua fica facilitada com um processo de realimentação permanente

PSP e CMM

PSP foi criado a partir do CMM

A idéia é trazer ao nível de programador os conceitos de processo tratados pelo CMM

Conclusão

- Qualidade do Produto
- Qualidade do Processo
- Maturidade do Processo
- Avaliação de Qualidade