Capítulo 24

Gerenciamento de Qualidade

Tópicos abordados

engenharia de SOFTWARE

- Qualidade de software
- Padrões de software
- Revisões e inspeções
- Medições e métricas de software

Gerenciamento de qualidade de software

engenharia de SOFTWARE

- Preocupados em garantir que o nível necessário de qualidade seja alcançado em um produto de software.

- Três principais preocupações:

  - ✓ No nível organizacional, o gerenciamento de qualidade se preocupa em estabelecer um quadro de processos organizacionais e padrões que irão gerar um software de alta qualidade.
  - ✓ No nível de projeto, o gerenciamento de qualidade envolve a aplicação de processos de qualidade específicos e a verificação de que esses processos planejados sejam seguidos.
  - ✓ No nível de projeto, o gerenciamento de qualidade também está preocupada com o estabelecimento de um plano de qualidade para um projeto. O plano de qualidade deve estabelecer as metas de qualidade para o projeto e definir os processos e padrões a serem usados.

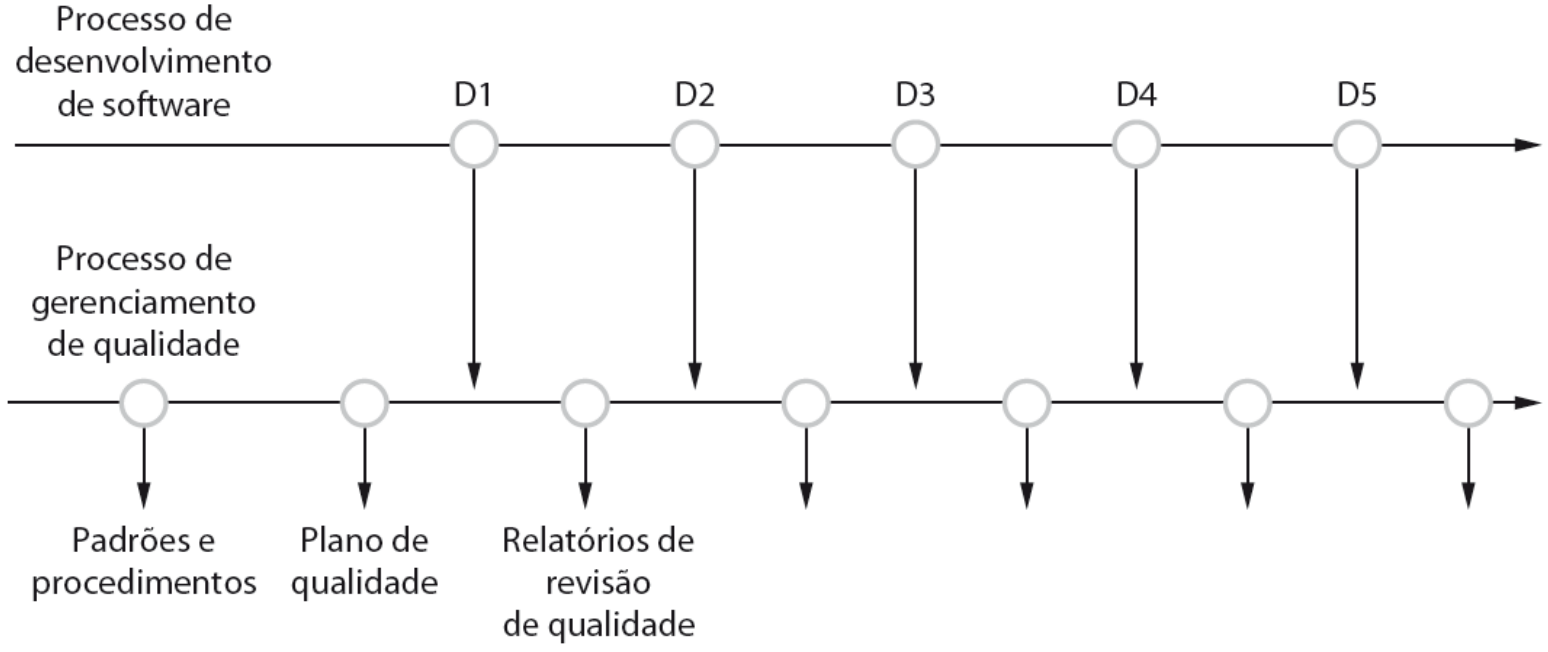
Atividades de gerenciamento de qualidade

engenharia de SOFTWARE

- O gerenciamento de qualidade fornece uma verificação independente a respeito do processo de desenvolvimento de software.
- O processo de gerenciamento da qualidade verifica as entregas do projeto para garantir que sejam consistentes com os objetivos e padrões organizacionais.
- A equipe de qualidade deve ser independente da equipe de desenvolvimento para que possa ter uma visão objetiva do software. O que permite que façam relatórios sobre a qualidade do software, que não sejam influenciados por questões de desenvolvimento de software.

Gerenciamento de qualidade e desenvolvimento de software

engenharia de SOFTWARE



Planejamento de qualidade

engenharia de SOFTWARE

- Um plano de qualidade define as qualidades desejadas do produto e como esses são avaliados, além de definir os atributos de qualidade mais significativos.
- O plano de qualidade deve definir o processo de avaliação da qualidade.
- Ele deve estabelecer quais padrões da organização devem ser aplicadas e, se necessário, definir os novos padrões a serem usados.

Planos de qualidade

- Estrutura do plano de qualidade:
  - ✓ Introdução ao produto;
  - ✓ Planos de produto;
  - ✓ Descrições de processo;
  - ✓ Metas de qualidade;
  - ✓ Riscos e gerenciamento de riscos.
- Os planos de qualidade devem ser documentos curtos, sucintos.
  - ✓ Se são muito longos, ninguém vai lê-los.

Âmbito do gerenciamento de qualidade

engenharia de SOFTWARE

- O gerenciamento de qualidade é particularmente importante para sistemas grandes e complexos.
- A documentação de qualidade é um registro do progresso e apoia a continuidade do desenvolvimento na medida em que a equipe de desenvolvimento muda.
- Para sistemas menores, o gerenciamento de qualidade necessita de menos documentação e deve se concentrar em estabelecer uma cultura de qualidade.

Qualidade de software

engenharia de SOFTWARE

- De uma maneira simplista, a qualidade significa que um produto deve corresponder às suas especificações.

- O que é problemático para os sistemas de software.
  - ✓ Existe uma tensão entre os requisitos de qualidade do cliente (eficiência, confiabilidade, etc.) e os requisitos de qualidade do desenvolvedor (reúso, de manutenção, etc.);
  - ✓ Alguns requisitos de qualidade são difíceis de se especificar de forma inequívoca;
  - ✓ Geralmente as especificações de software são incompletas e muitas vezes inconsistentes.

- O foco pode ser "adequação à finalidade" em vez de conformidade à especificação.

Adequação do software à finalidade

engenharia de SOFTWARE

- Durante o processo de desenvolvimento os padrões de programação e documentação foram seguidos?
- O software foi devidamente testado?
- O software é confiável o suficiente para ser colocado em uso?
- O desempenho do software é aceitável para uso normal?
- O software é usável?
- O software bem é compreensível e estruturado ?

Atributos de qualidade de software

engenharia de SOFTWARE

Segurança	Compreensibilidade	Portabilidade
Proteção	Testabilidade	Usabilidade
Confiabilidade	Adaptabilidade	Reusabilidade
Resiliência	Modularidade	Eficiência
Robustez	Complexidade	Capacidade de aprendizado

Conflitos de qualidade

- Não é possível para qualquer sistema ser otimizado para todos esses atributos – por exemplo, melhorar a robustez poderá levar à perda de desempenho.

- Portanto, o plano de qualidade deve definir os atributos de qualidade mais importantes para o software que está sendo desenvolvido.

- O plano também deve incluir uma definição do processo de avaliação de qualidade, uma forma acordada de avaliar se alguma qualidade, como a de manutenção ou robustez, está presente no produto.

Qualidade de processo e produto

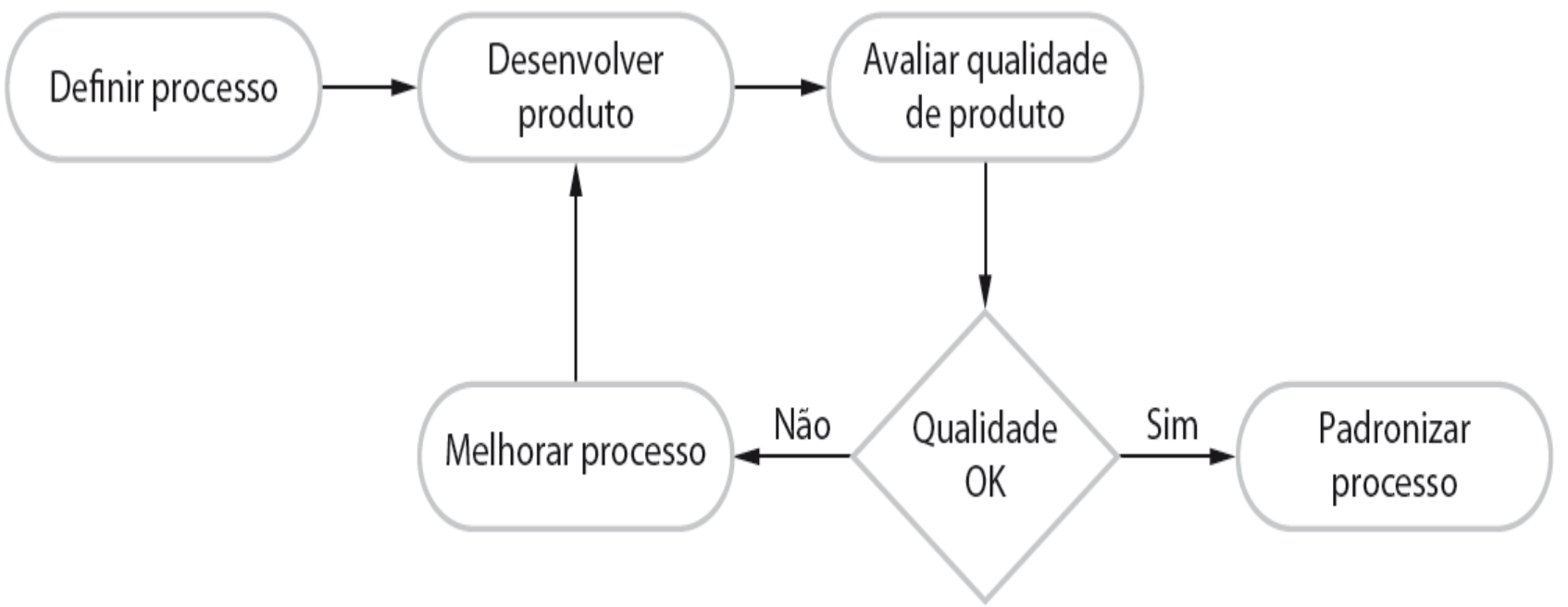
- A qualidade de um produto desenvolvido é influenciada pela qualidade do processo de produção.

- Isso é importante no desenvolvimento de software pois alguns atributos de qualidade de produto são difíceis de avaliar.

- No entanto, existe uma relação muito complexa e mal compreendida entre os processos de software e a qualidade de produto.

  - ✓ A aplicação das habilidades individuais e experiência é particularmente importante no desenvolvimento de software;
  - ✓ Fatores externos, tais como a novidade de uma aplicação ou a necessidade de um cronograma de desenvolvimento acelerado pode prejudicar a qualidade de produto.

Qualidade baseada em processos



Padrões de software

engenharia de SOFTWARE

- O padrões definem os atributos necessários de um produto ou processo. Eles desempenham um papel importante no gerenciamento de qualidade.
- Os padrões podem ser internacionais, nacionais, padrões organizacionais ou de projeto.
- Os padrões de produto definem características que todos os componentes de software devem exibir, por exemplo, um estilo de programação comum.
- Os padrões de processo definem como o processo de software deve ser seguido.

Importância dos padrões

engenharia de SOFTWARE

- Uma síntese das melhores práticas evita a repetição de erros do passado.
- São um framework para definir o que significa a qualidade de uma determinada configuração, ou seja, a visão de qualidade daquela organização.
- Eles fornecem a continuidade – pessoas novas podem compreender a organização através da compreensão dos padrões usados.

Padrões de produto e de processo

Padrões de produto	Padrões de processo
Formulário de revisão de projeto	Condução de revisão de projeto
Estrutura de documento de requisitos	Apresentação do novo código para a construção de sistema
Formato de cabeçalho de método	Processo de versão e *release*
Estilo de programação Java	Processo de aprovação de plano de projeto
Formato de plano de projeto	Processo de controle de mudança
Formulário de solicitação de mudança	Processo de registro de teste

Problemas com os padrões

engenharia de SOFTWARE

- Podem não ser percebidos como relevantes e atualizados pelos engenheiros de software.
- Muitas vezes eles envolvem o preenchimento de formulários muito burocráticos.
- Se não recebem apoio de ferramentas de software, geralmente, o preenchimento tedioso de formulários é acrescentado para manter a documentação associada com os padrões.

Desenvolvimento de padrões

engenharia de SOFTWARE

- Envolver os profissionais no desenvolvimento. Os engenheiros devem compreender as razões subjacentes a um padrão.

- Revisões regulares dos padrões e seu uso. Os padrões podem se tornar desatualizados rapidamente o que reduz a sua credibilidade entre os profissionais.

- Padrões detalhados devem ter suporte de ferramentas especializadas. O trabalho burocrático excessivo é a queixa mais significativa aos padrões.

  - ✓ Formulários baseados na web não são bons o suficiente.

O *framework* de normas ISO 9001

- Um conjunto de padrões internacionais que podem ser usados como base para o desenvolvimento de sistemas de gerenciamento de qualidade.

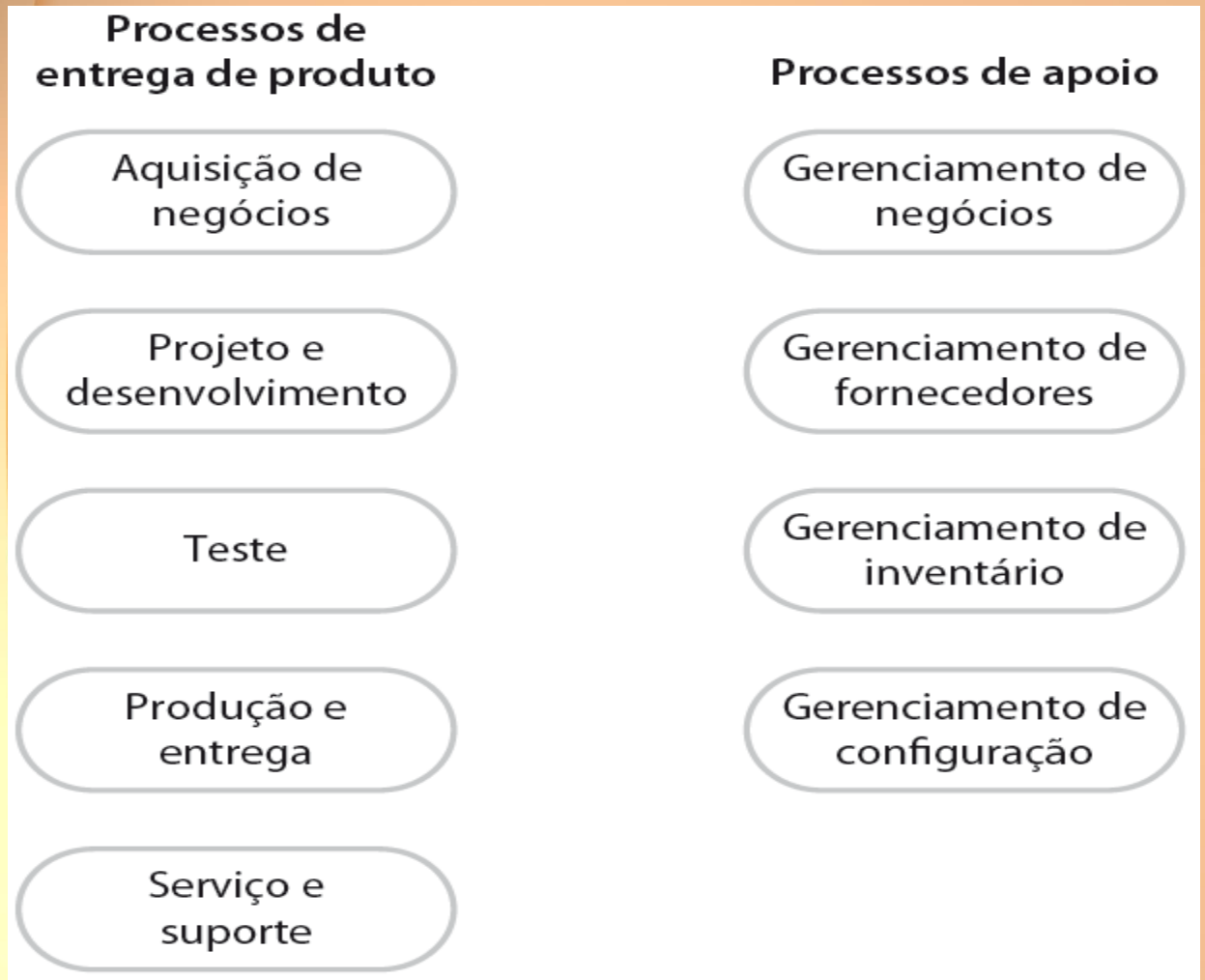
- ISO 9001, a mais geral dessas normas, aplica-se a organizações que projetam, desenvolvem e mantém produtos, incluindo software.

- A norma ISO 9001 é um framework para desenvolvimento de padrões de software.

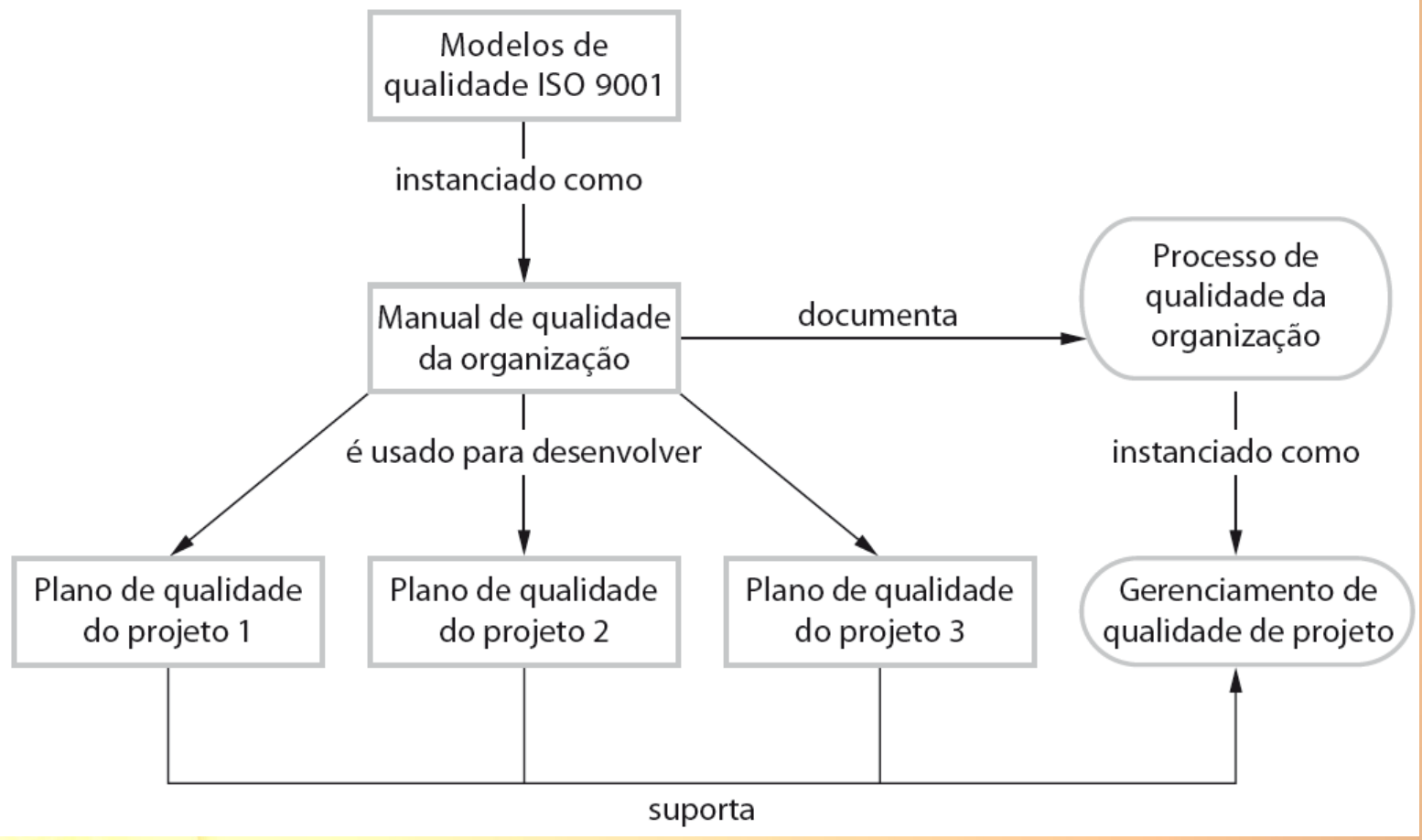
  - ✓ Estabelece princípios de qualidade geral, descreve os processos de qualidade em geral e estabelece os padrões de organização e procedimentos que devem ser definidos. Esses devem ser documentados em um manual de qualidade da organização.

Processos essenciais da ISO 9001

engenharia de SOFTWARE



ISO 9001 e gerenciamento de qualidade



A certificação ISO 9001

engenharia de SOFTWARE

- Os padrões e os procedimentos de qualidade devem ser documentados em um manual de qualidade da organização.
- Um órgão externo pode certificar que um manual de qualidade da organização está em conformidade com a norma ISO 9001.
- Alguns clientes exigem que os fornecedores tenham a certificação ISO 9001, embora a necessidade de flexibilidade aqui seja cada vez mais reconhecida.

Pontos importantes

- O gerenciamento de qualidade de software está preocupada com a garantia de que o software tenha um baixo número de defeitos e que atinja os padrões exigidos de manutenção, confiabilidade, portabilidade e, assim por diante.

- O gerenciamento de qualidade de software inclui a definição de padrões para processos e produtos e o estabelecimento de processos para verificar se esses padrões foram seguidos.

- Os padrões de software são importantes para garantia de qualidade pois representam uma identificação das "melhores práticas“.

- Os procedimentos de gerenciamento de qualidade podem ser documentados em um manual de qualidade da organização, com base no modelo genérico para um manual de qualidade sugerido na norma ISO 9001.

Revisões e inspeções

- Um grupo examina parte ou a totalidade de um processo ou sistema e sua documentação para encontrar potenciais problemas.

- Softwares ou documentos podem ser "assinados" em uma revisão, o que significa que o avanço para a fase seguinte de desenvolvimento foi aprovado pela gerência.

- Existem diferentes tipos de revisão com objetivos diferentes

  - ✓ Inspeções para remoção de defeitos (produto);
  - ✓ Comentários para avaliação de progresso (produto e processo);
  - ✓ Avaliações de qualidade (produto e padrões).

Avaliações de qualidade

engenharia de SOFTWARE

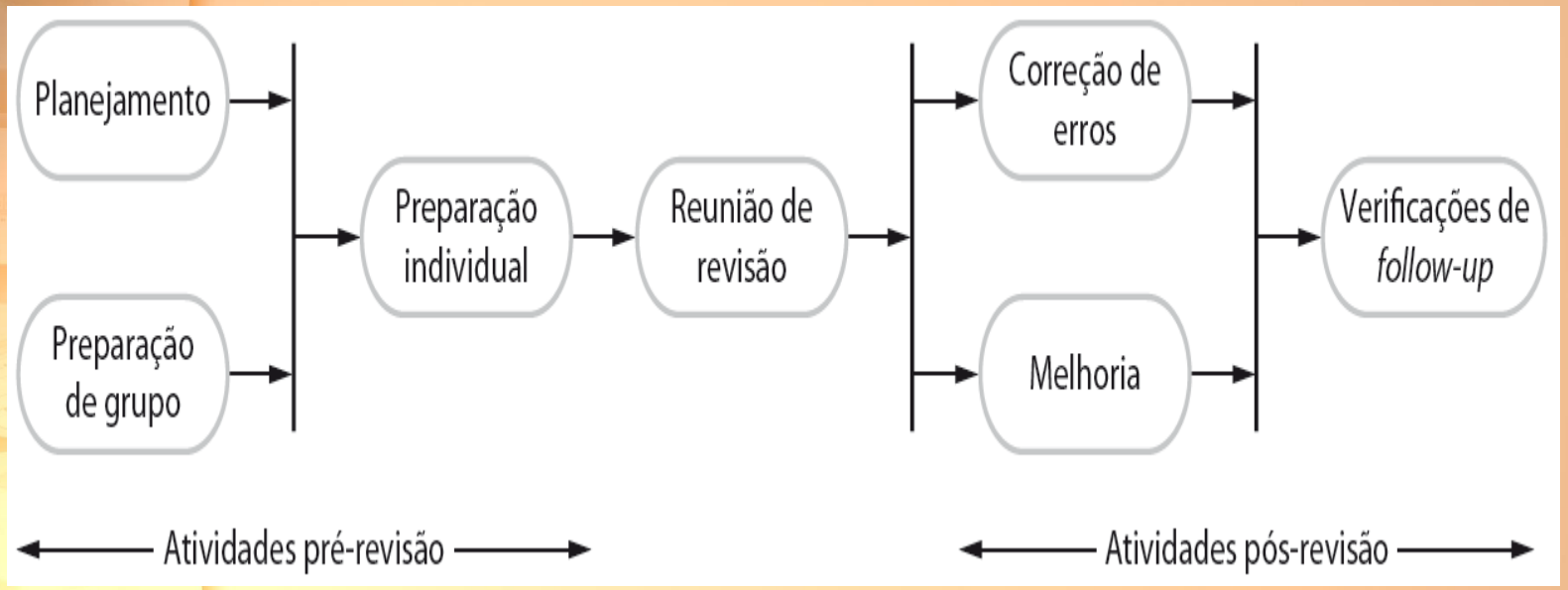
- Um grupo de pessoas examinam cuidadosamente parte ou a totalidade de um sistema de software e sua documentação associada.

- Código, projetos, especificações, planos de teste, padrões, etc., todos podem ser revistos.

- Softwares ou documentos podem ser "assinados" em uma revisão, o que significa que o avanço para a fase seguinte de desenvolvimento foi aprovado pela gerência.

O processo de revisão de software

engenharia de SOFTWARE



Comentários e métodos ágeis

- Geralmente, o processo de revisão no desenvolvimento ágil de software é informal.
  - ✓ Em *Scrum*, por exemplo, há uma reunião de avaliação após cada iteração do software ser concluída (uma revisão *sprint*), na qual as questões de qualidade e problemas podem ser discutidos.
- No *Extreme Programming* (XP), a programação em pares garante que o código seja constantemente examinado e revisado por outro membro da equipe.
- XP se baseia em pessoas que tomam a iniciativa de melhorar e refatorar o código. As abordagens ágeis geralmente não são orientadas por padrões, para que os problemas de compatibilidade com os padrões normalmente não sejam considerados.

Inspeções de programa

- Essas são avaliações em pares, em que os engenheiros examinam a fonte de um sistema com o objetivo de descobrir anomalias e defeitos.
- As inspeções não exigem a execução de um sistema, assim podem ser usadas antes da implementação.
- Elas podem ser aplicadas a qualquer representação do sistema (requisitos, projeto, configuração, dados de teste, etc.)
- Elas têm se mostrado uma técnica eficaz para descobrir bugs de programas.

Checklists de inspeção

- Um *checklist* de inspeção dos defeitos comuns deve ser usada para conduzir a inspeção.
- Os *checklists* de inspeção de defeitos são dependentes da linguagem de programação e refletem os erros característicos que podem surgir na linguagem.
- Em geral, quanto "mais fraca" a verificação de tipos, maior o *checklist*.
- Exemplos: Iniciação, nomeação de constantes, repetição de loop, limites de vetor, etc.

Um *checklist* de inspeção (a)

Classe de defeito	Verificação de inspeção
Defeitos de dados	• Todas as variáveis de programa são iniciadas antes que seus valores sejam usados? • Todas as constantes foram nomeadas? • O limite superior de vetores deve ser igual ao tamanho do vetor ou ao tamanho –1? • Se as *strings* de caracteres são usadas, um delimitador é explicitamente atribuído? • Existe alguma possibilidade de *overflow* de *buffer*?
Defeitos de controle	• Para cada instrução condicional, a condição está correta? • É certo que cada *loop* vai terminar? • As declarações compostas estão posicionadas corretamente entre colchetes? • Em declarações *case*, todos os *cases* possíveis são considerados? • Se um *break* é requerido após cada *case* em declarações *case*, este foi incluído?
Defeitos de entrada/saída	• Todas as variáveis de entrada são usadas? • Todas as variáveis de saída receberam um valor antes de serem emitidas? • Entradas inesperadas podem causar corrupção de dados?

Um *checklist* de inspeção (b)

engenharia de SOFTWARE

Classe de defeito	Verificação de inspeção
Defeitos de interface	• Todas as chamadas de funções e métodos têm o número correto de parâmetros? • Os parâmetros formais e reais correspondem? • Os parâmetros estão na ordem correta? • Se os componentes acessam memória compartilhada, eles têm o mesmo modelo de estrutura de memória compartilhada?
Defeitos de gerenciamento de armazenamento	• Se uma estrutura ligada é modificada, todas as ligações foram corretamente reatribuídas? • Se o armazenamento dinâmico é usado, o espaço foi alocado corretamente? • O espaço é explicitamente desalocado após não ser mais necessário?
Defeitos de gerenciamento de exceção	• Foram levadas em consideração todas as condições possíveis de erro?

Métodos ágeis e inspeções

engenharia de SOFTWARE

- Processos ágeis raramente usam inspeção formal ou processos de revisão em pares.

- Em vez disso, eles contam com membros da equipe cooperando para controlar os código uns dos outros, e as orientações informais, tais como ‘verifique antes do check-in', que sugerem que os programadores devem verificar o seu próprio código.

- Os adeptos do *Extreme Programming* argumentam que a programação em pares é um substituto eficaz para a inspeção, pois é, com efeito, um processo de inspeção contínua.

- Duas pessoas olham para cada linha de código e verificam essa antes que seja aceita.

Medições e métricas de software

engenharia de SOFTWARE

- A medição de software está preocupada com a derivação de um valor numérico para um atributo de um produto de software ou processo.

  - O que permite comparações objetivas entre as técnicas e os processos.

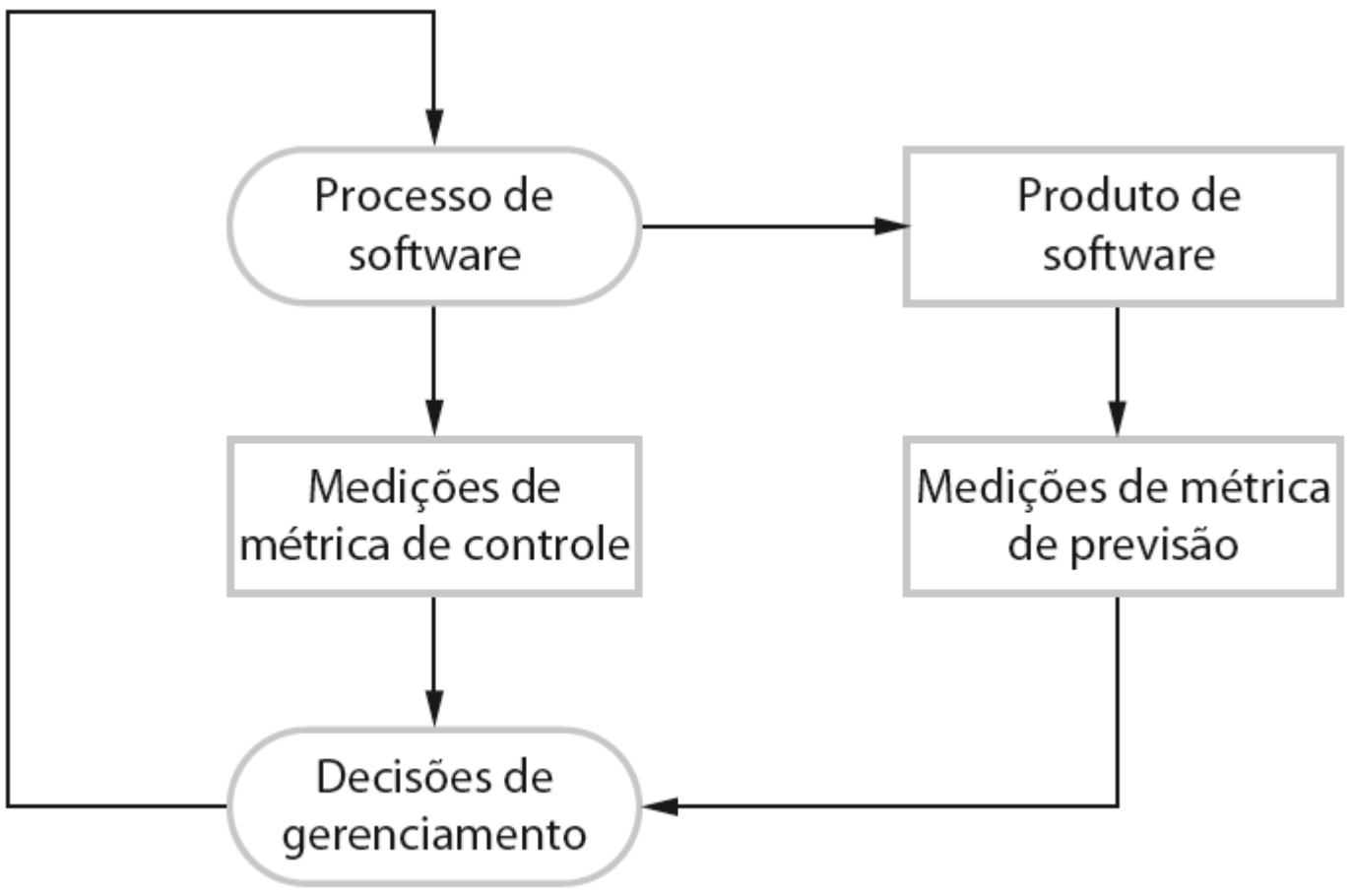
- Embora algumas empresas introduzissem programas de medição, a maioria das organizações ainda não fazem uso sistemático de medição de software.

- Existem poucos padrões estabelecidos nesta área.

Métricas de software

- Qualquer tipo de medida que se relaciona com um sistema de processo ou documentação relacionada ao software.
  - ✓ As linhas de código em um programa, o índice Fog, o número de pessoas-dia necessários para desenvolver um componente.
- Permitem que o software e o processo de software sejam quantificados.
- Podem ser usados para prever os atributos de produto ou para controlar o processo de software.
- As métricas de produto podem ser usadas para previsões gerais ou para identificar os componentes anômalos.

Medições de previsão e controle



Uso de medições

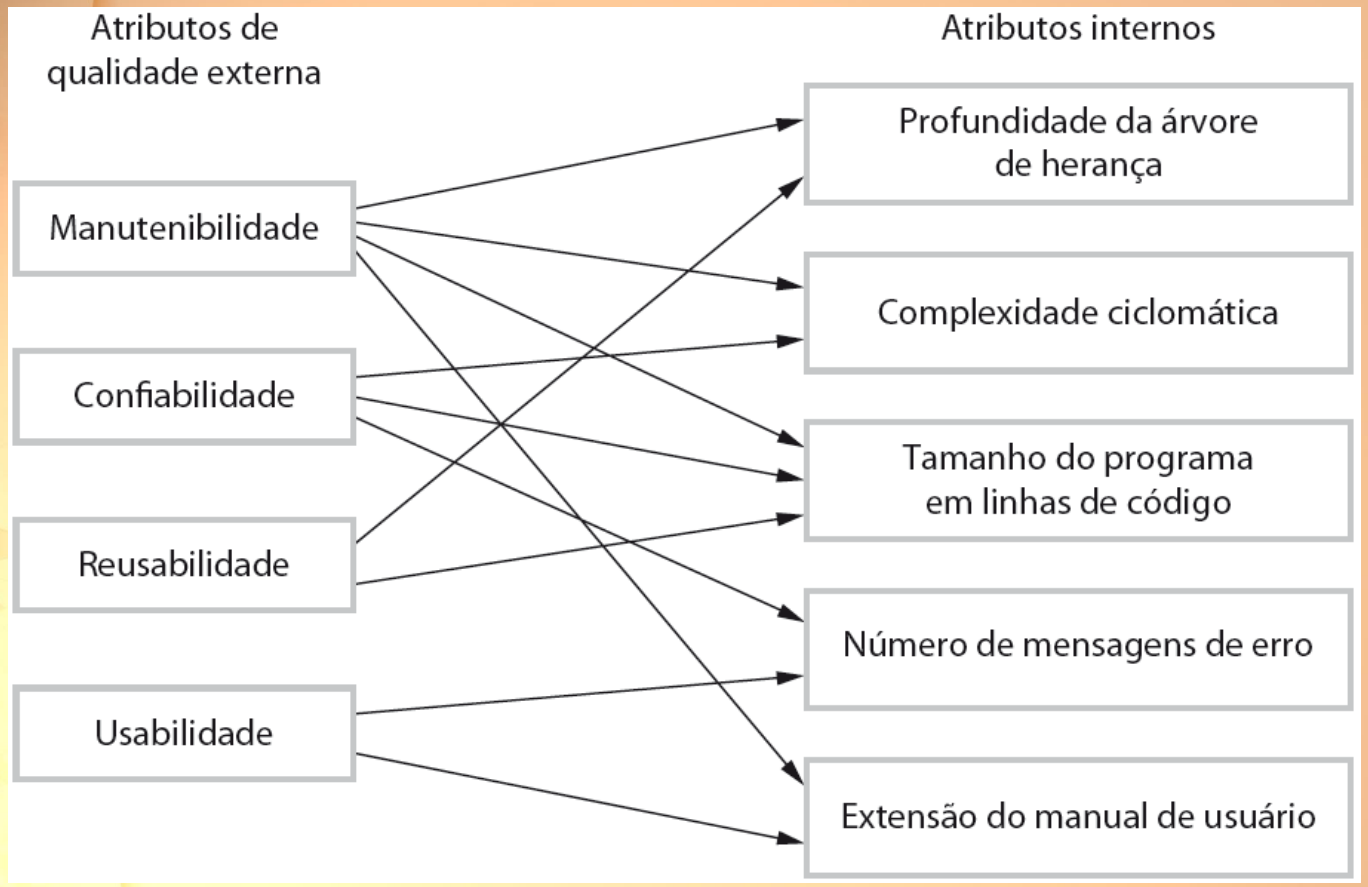
- Para atribuir um valor aos atributos de qualidade de sistema
  - ✓ Ao medir as características dos componentes do sistema, tais como a sua complexidade ciclomática, e depois agregar essas medições, você pode avaliar atributos do sistema de qualidade, tais como a manutenibilidade.
- Para identificar os componentes de sistema cuja qualidade não atingiu o padrão
  - ✓ As medições podem identificar os componentes individuais, com características que se desviam do padrão. Por exemplo, você pode medir componentes para descobrir aqueles com maior complexidade. Esses são mais prováveis de conter bugs pois a complexidade dificulta o entendimento.

Suposições de métricas

engenharia de SOFTWARE

- Um atributo de software pode ser medido.
- O relacionamento existente entre o que podemos medir e o que queremos saber. Nós só podemos medir atributos internos, mas muitas vezes existe mais interesse nos atributos externos do software.
- Esse relacionamento tem sido formalizado e validado.
- Pode ser difícil relacionar o que pode ser medido com atributos de qualidade externos desejáveis.

Relacionamento entre os atributos internos e externos de software



Problemas com medições na indústria

- É impossível quantificar o retorno sobre o investimento de introduzir um programa de métricas organizacionais.
- Não existe um padrão para métricas de software ou processos padronizados para medição e análise.
- Em muitas empresas, os processos de software não são padronizados e estão mal definidos e controlados.
- A maioria dos trabalhos a respeito da medição de software tem se concentrado em métricas baseadas em códigos e processos de desenvolvimento dirigidos a planos. No entanto, atualmente mais e mais softwares são desenvolvidos pela configuração de sistemas ERP ou COTS.
- A introdução da medição acrescenta um overhead aos processos.

Métricas de produto

- Uma métrica de qualidade deve ser um preditor da qualidade de produto.

- Classes de métricas de produto

  - ✓ As métricas dinâmicas que são coletados através de medições efetuadas em um programa em execução;

  - ✓ Métricas estáticas, as quais são coletadas através de medições efetuadas nas representações do sistema;

  - ✓ Métricas dinâmicas ajudam a avaliar a eficiência e a confiabilidade;

  - ✓ Métricas estáticas ajudam a avaliar a compreensibilidade, a complexidade e a manutenibilidade.

Métricas dinâmicas e estáticas

engenharia de SOFTWARE

- As métricas dinâmicas estão intimamente relacionadas com os atributos de qualidade de software.
  - ✓ É relativamente fácil medir o tempo de resposta de um sistema (atributo de desempenho) ou o número de falhas (atributo de confiabilidade).
- As métricas estáticas têm uma relação indireta com os atributos de qualidade.
  - ✓ Você precisa tentar obter um relacionamento entre essas métricas e suas propriedades, tais como inteligibilidade, complexidade e manutenibilidade.

Métricas estáticas de produto de software

engenharia de SOFTWARE

Métrica de software	Descrição
Fan-in/*Fan-out*	*Fan-in* é a medida do número de funções ou métodos que chamam outra função ou método (digamos X). *Fan-out* é o número de funções que são chamadas pela função de X. Um valor alto para *fan-in* significa que X está fortemente acoplado ao resto do projeto e alterações em X terão repercussões extensas. Um valor alto para *fan-out* sugere que a complexidade geral do X pode ser alta por causa da complexidade da lógica de controle necessário para coordenar os componentes chamados.
Comprimento de código	Essa é uma medida do tamanho de um programa. Geralmente, quanto maior o tamanho do código de um componente, mais complexo e sujeito a erros o componente é. O comprimento de código tem mostrado ser uma das métricas mais confiáveis para prever a propensão a erros em componentes.

Métricas estáticas de produto de software

Métrica de software	Descrição
Complexidade ciclomática	Essa é uma medida da complexidade de controle de um programa. Essa complexidade de controle pode estar relacionada à compreensibilidade de programa. A complexidade ciclomática é discutida no Capítulo 8.
Comprimento de identificadores	Essa é uma medida do comprimento médio dos identificadores (nomes de variáveis, classes, métodos etc.) em um programa. Quanto mais longos os identificadores, mais provável que sejam significativos e, portanto, mais compreensível o programa.
Profundidade de aninhamento condicional	Essa é uma medida da profundidade de aninhamento de declarações *if* em um programa. Declarações *if* profundamente aninhadas são difíceis de entender e potencialmente sujeitas a erros.
Índice *Fog*	Essa é uma medida do comprimento médio de palavras e sentenças em documentos. Quanto maior o valor de um índice *Fog* de um documento, mais difícil a sua compreensão.

O conjunto de métricas de CK orientadas a objetos

Métrica orientada a objeto	Descrição
Métodos ponderados por classe (WMC, do inglês *weighted methods per class*)	É o número de métodos em cada classe, ponderados pela complexidade de cada método. Portanto, um método simples pode ter uma complexidade de 1 e um método grande e complexo pode ter um valor muito superior. Quanto maior o valor para essa métrica, mais complexa a classe de objeto. Geralmente, os objetos complexos são mais difíceis de compreender. Eles podem não ser logicamente coesos, portanto, não podem ser reusados efetivamente como superclasses em uma árvore de herança.
Árvore de profundidade de herança (DIT, do inglês *depth Of inheritance tree*)	Representa o número de níveis discretos na árvore de herança em que as subclasses herdam atributos e operações (métodos) de superclasses. Quanto mais profunda a árvore de herança, mais complexo o projeto. Muitas classes de objeto podem precisar ser compreendidas para que as classes de objeto nas folhas da árvore sejam entendidas.
Número de filhos (NOC, do inglês *number of children*)	É uma medida do número de subclasses imediatas em uma classe. Ele mede a largura de uma hierarquia de classe, considerando que DIT mede sua profundidade. Um valor alto para NOC pode indicar um maior reúso. Isso pode significar que mais esforço deve ser dispendido na validação de classes de base por causa do número de subclasses que dependem delas.

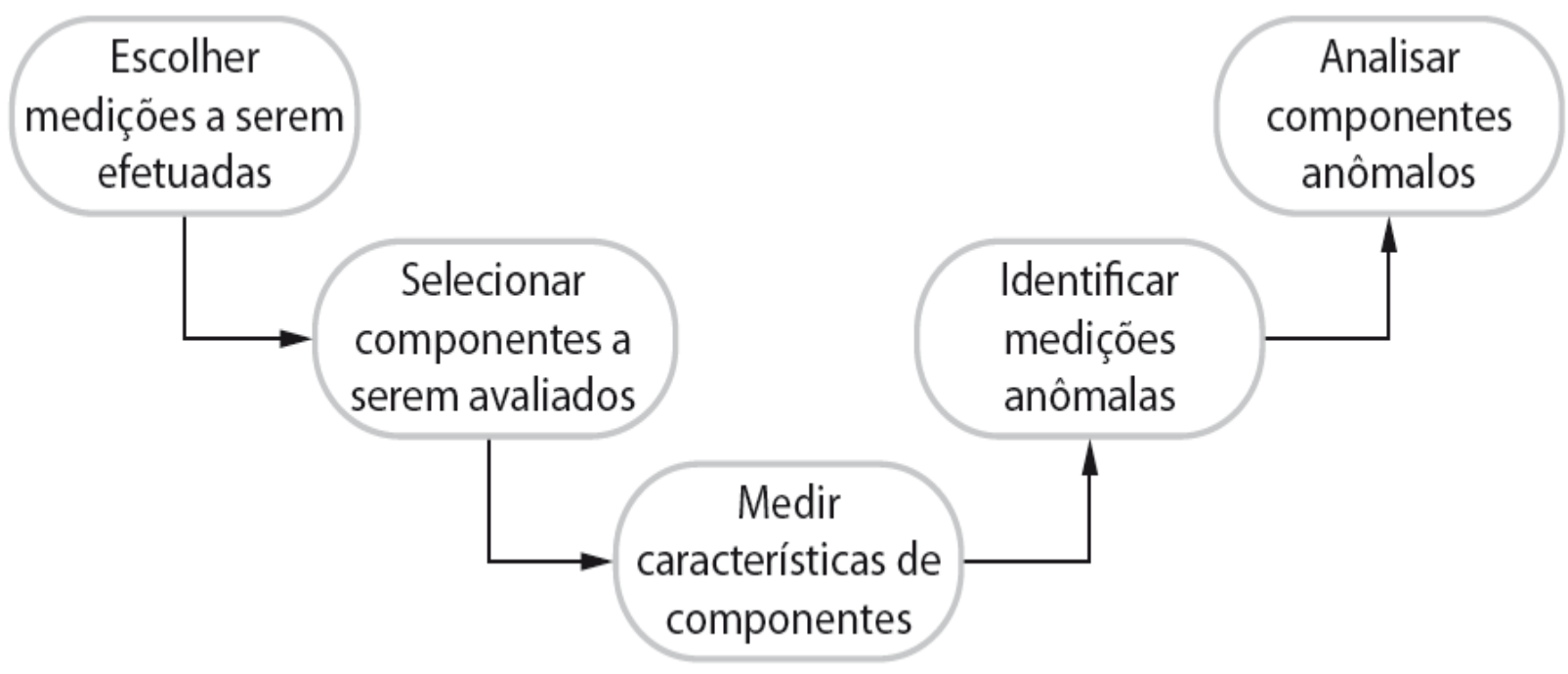
O conjunto de métricas de CK orientadas a objetos

Métrica orientada a objeto	Descrição
Acoplamento entre classes de objeto (CBO, do inglês *coupling between object classes*)	Classes são acopladas quando métodos em uma classe usam métodos ou variáveis de instância definidas em uma classe diferente. CBO é uma medida de quanto acoplamento existe. Um valor alto para o CBO significa que as classes são altamente dependentes e, portanto, é mais provável que a mudança em uma classe afete outras classes do programa.
Resposta para uma classe (RFC, do inglês *response for a class*)	RFC é a medida do número de métodos que poderiam ser executados em resposta a uma mensagem recebida por um objeto dessa classe. Mais uma vez, RFC está relacionada com a complexidade. Quanto maior o valor de RFC, mais complexa é a classe e, portanto, mais provável que inclua erros.
Falta de coesão em métodos (LCOM, do inglês *lack of cohesion in methods*)	LCOM é calculada considerando os pares de métodos em uma classe. LCOM é a diferença entre o número de pares de métodos sem atributos compartilhados e o número de pares de métodos com atributos compartilhados. O valor dessa métrica tem sido amplamente discutido, e ele existe em diversas variações. Não está claro se realmente adiciona qualquer informação útil além das que já são fornecidas por outras métricas.

Análise de componentes de software

- Os componentes de sistema podem ser analisados separadamente, usando uma variedade de métricas.
- Os valores dessas métricas podem, então, ser comparados com diferentes componentes e, talvez, com dados históricos de medição coletados em projetos anteriores.
- Medições anômalas, que se afastem significativamente do padrão, podem implicar na existência de problemas com a qualidade desses componentes.

O processo de medição de produto



Surpresas nas medições

engenharia de SOFTWARE

- Reduzir o número de defeitos em um programa leva a um aumento do número de ligações para o help desk.

  - ✓ Agora, o programa é percebido como mais confiável e por isso mesmo, existe um mercado mais amplo e diversificado. O percentual de usuários que ligam para o help desk pode ter diminuído, mas o total pode aumentar;

  - ✓ Um sistema mais confiável é usado de maneira diferente de um sistema em que os usuários trabalham em torno das falhas. O que ocasiona um maior número de ligações para o help desk.

Pontos importantes

engenharia de SOFTWARE

- Revisões dos resultados do processo de software envolve uma equipe de pessoas que verifica se os padrões de qualidade estão sendo seguidos.
- Em uma inspeção de programa ou revisão por pares, uma pequena equipe verifica sistematicamente o código. Eles leem o código em detalhes e procuram por possíveis erros e omissões
- A medição de softwares pode ser usada para coletar dados sobre o software e sobre os processos de software.
- Métricas de qualidade de produto são particularmente úteis para destacar os componentes anômalos que podem ter problemas de qualidade.