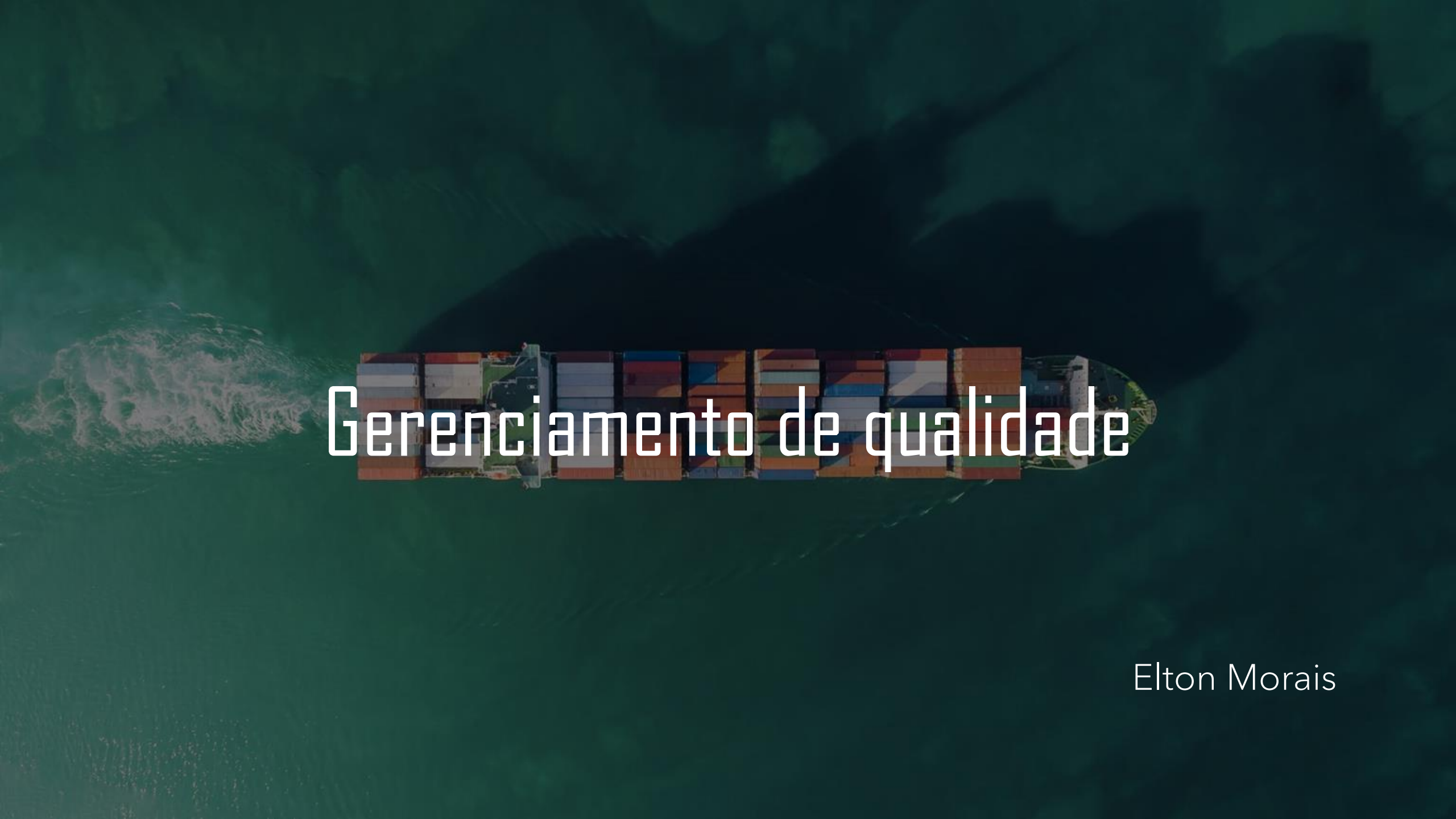


Gerenciamento de qualidade

Elton Morais

Objetivos

Objetivos

- Conhecer o processo de gerenciamento de qualidade;
- Identificar o por que o planejamento de qualidade é importante;
- Compreender que a qualidade do software é afetada pelo processo usado em seu desenvolvimento;
- Tomar ciência da importância dos padrões no processo de gerenciamento de qualidade;

Objetivos

- Saber como os padrões são usados a fim de garantir a qualidade;
- Compreender de que maneira revisões e inspeções são usadas como um mecanismo de garantia de qualidade de software;
- Compreender de que maneira a medição é útil na avaliação de alguns atributos de qualidade de software e as limitações atuais da medição de software.

Introdução

Os problemas com a qualidade de software foram inicialmente descobertos na década de 1960 com o desenvolvimento do primeiro grande sistema e continua a incomodar a engenharia de software desde então.

Introdução

O **gerenciamento de qualidade** iniciou com métodos usados na **indústria manufatureira**. Essas técnicas, em conjunto com **novas tecnologias** de software e melhores **testes de software**, conduziram a melhorias significativas no nível geral da **qualidade de software**.

O gerenciamento de qualidade tem três principais preocupações, sendo a primeira, no **nível organizacional** e as outras duas no **nível de projeto**.

Introdução

No **nível organizacional**, o gerenciamento de qualidade está preocupado com o estabelecimento de um *framework* de **processos organizacionais** e **padrões** que elevem a softwares de alta qualidade.

A **equipe** de gerenciamento de qualidade deve assumir a responsabilidade de **definir os processos** de desenvolvimento do software que serão usados e os **padrões** que devem ser usados no software, bem como a **documentação** relacionada, incluindo os **requisitos de sistema**, **projeto** e **código**.

Introdução

No **nível de projeto**, o gerenciamento de qualidade verifica se os **processos planejados** foram seguidos, e as **saídas** de projeto estejam em conformidade com os padrões aplicáveis ao projeto.

Também está preocupado com o estabelecimento de um **plano de qualidade**, que define as metas de qualidade para o projeto e quais processos e padrões devem ser usados.

Introdução

Garantia de qualidade (*Quality Assurance – QA)* é a definição de processos e padrões que devem conduzir a produtos de alta qualidade e a introdução de processos de qualidade na fabricação.

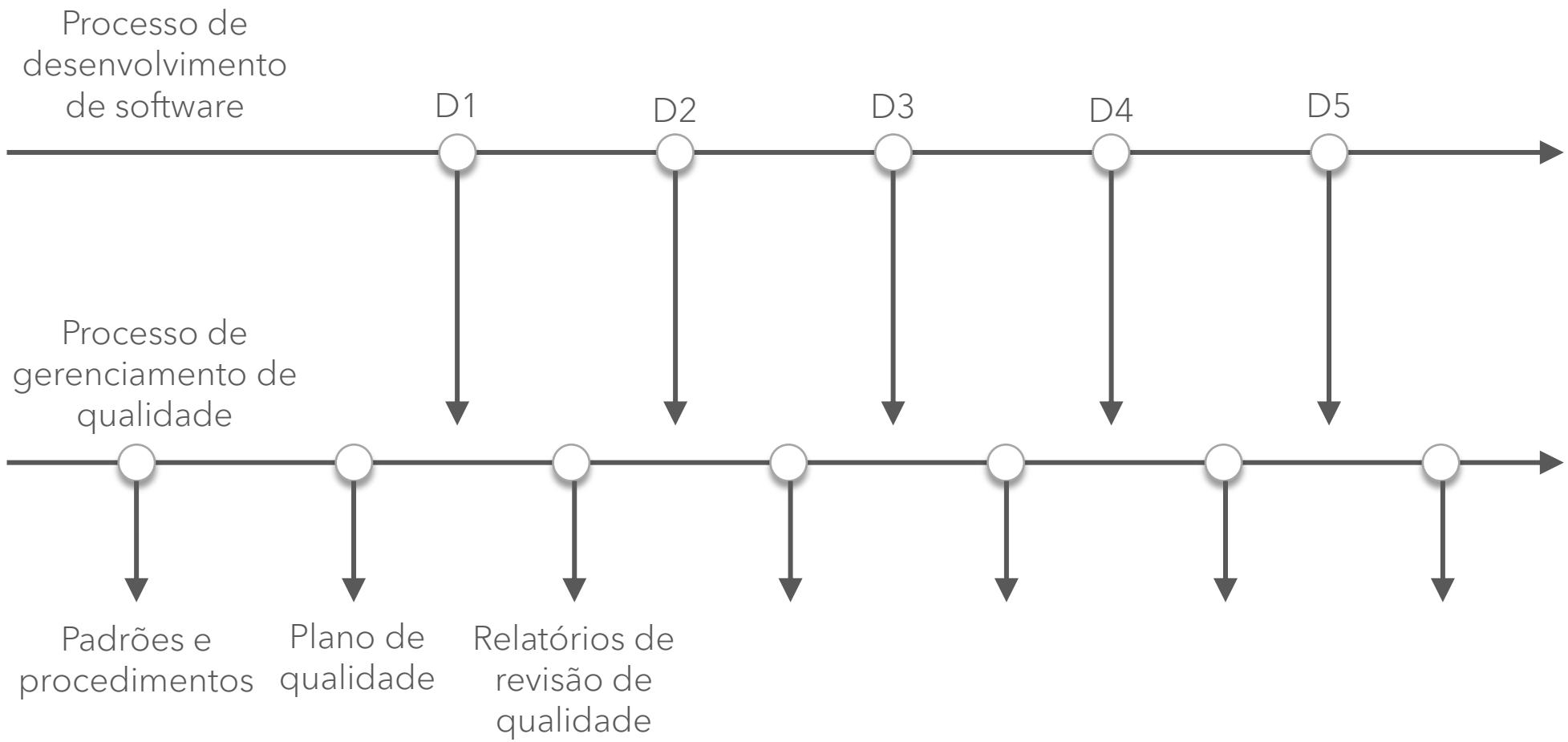
O **controle de qualidade** é a aplicação desses processos de qualidade visando eliminar os produtos que não atingiram o nível de qualidade exigido.

Introdução

O processo de gerenciamento de qualidade verifica os entregáveis de projeto para garantir que eles sejam consistentes com os padrões e objetivos organizacionais.

A equipe de QA deve ser independente da equipe de desenvolvimento, e preferencialmente não deve ser associada a qualquer grupo de desenvolvimento, mas deve ter a responsabilidade de toda a organização para o gerenciamento de qualidade.

Gerenciamento de qualidade e desenvolvimento de software



FONTE: SOMMERVILLE (2011)

Introdução

O planejamento de qualidade é o processo de desenvolvimento de um plano de qualidade para um projeto. Esse plano estabelece as qualidades desejadas, e como elas serão avaliadas.

Um plano de qualidade inclui:

- Introdução ao produto;
- Planos de produto;
- Descrições de processo;
- Metas de qualidade;
- Riscos e gerenciamento de riscos.

Qualidade de software

Os fundamentos do gerenciamento de qualidade foram estabelecidos pela indústria manufatureira em um esforço para melhorar a qualidade dos produtos em produção.

Qualidade de software

Na qualidade de software não é aplicável a ideia de tolerância. Ainda assim, pode ser impossível concluir objetivamente se um sistema de software cumpre ou não suas especificações.

Isso ocorre, por exemplo, na especificação de requisitos, onde cliente e desenvolvedores podem interpreta-los de maneiras diferentes.

Além disso, *stakeholders* que não participaram diretamente da especificação de requisitos pode concluir que o sistema é de baixa qualidade.

Qualidade de software

A avaliação de software é um processo subjetivo, ficando a cargo da equipe de gerenciamento de qualidade julgar e decidir se foi alcançado um nível aceitável de qualidade.

Basicamente, trata-se de responder perguntas sobre as características do sistema. Por exemplo:

1. Durante o processo de desenvolvimento os padrões de programação e documentação foram seguidos?
2. O software foi devidamente testado?
3. O software é suficientemente confiável para ser colocado em uso?
4. O desempenho do software é aceitável para uso normal?
5. O software é útil?
6. O software é bem estruturado e compreensível?

Qualidade de software

Geralmente, no gerenciamento de qualidade, os testes de sistema serão baseados em seus requisitos.

A decisão sobre se o software oferece ou não a funcionalidade necessária deve basear-se nos resultados desses testes.

A equipe de QA deve revisar os testes desenvolvidos e analisar os registros de testes, com intuito de verificar se eles foram devidamente realizados.

A qualidade subjetiva de um sistema, em grande parte, baseia-se em suas características não funcionais.

Atributos de qualidade de software

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

FONTE: SOMMERVILLE (2011)

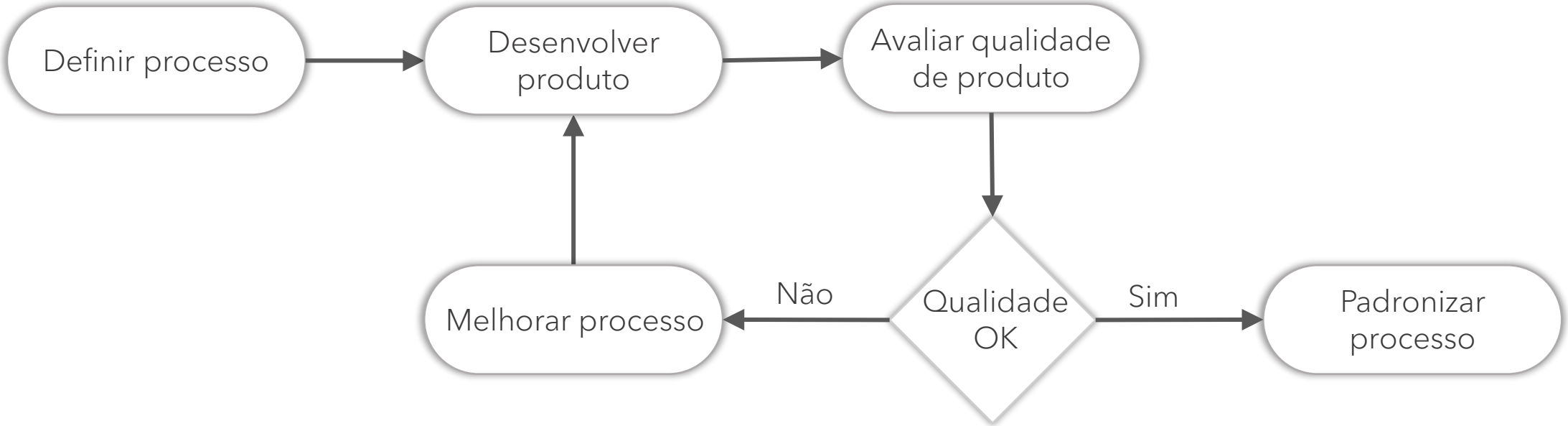
Qualidade de software

Um pressuposto do gerenciamento de qualidade de software é que a qualidade do software é diretamente relacionada à qualidade do processo de desenvolvimento de software.

Qualidade de produto é intimamente relacionada ao processo de produção.

No desenvolvimento de software a relação entre a qualidade de processo e de produto tem relação com competências e experiências individuais.

Qualidade baseada em processos



FONTE: SOMMERVILLE (2011)

Padrões de software

Os padrões de software desempenham um papel muito importante no gerenciamento de qualidade de software.

Padrões de software

Importância dos padrões de softwares:

1. Capturam sabedoria, que é valiosa para a organização;
2. Fornecem um *framework* para a definição do significado de "qualidade" em uma determinada organização;
3. Ajudam a dar continuidade ao trabalho realizado por uma pessoa, quando retomado e continuado por outra.

Tipos de padrões no gerenciamento de qualidade de software:

1. Padrões de produto;
2. Padrões de processo;

Padrões de software

Os padrões devem entregar valor.

Padrões de produto precisam ser projetados para serem aplicados e verificados de forma efetiva.

Padrões de processos devem incluir uma definição de processos que verifique se os padrões de produto foram seguidos.

O desenvolvimento de padrões para uma empresa devem, em geral, basear-se em padrões nacionais e internacionais. Deve-se elaborar um ‘manual’ de padrões.

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

FONTE: SOMMERVILLE (2011)

Padrões de software

O uso de padrões podem ser encorajados quando:

- Envolve os engenheiros de software na seleção de padrões de produto;
- Há revisão e mudanças regulares nos padrões para refletir mudanças nas tecnologias;
- Uso de ferramentas de software que fornecem suporte aos padrões.

O gerente de projeto e o gerente de qualidade podem decidir quais padrões organizacionais devem ser usados sem alterações, quais devem ser modificados e quais devem ser ignorados, isso logo no início do projeto.

Há possibilidade de novos padrões a serem incorporados para atender a novas necessidades.

O framework de normas ISO 9001

A ISO 9000 são normas para desenvolvimento de sistemas de gerenciamento de qualidade em diversos setores.

ISO 9001 – aplica-se a organizações que projetam, desenvolvem e mantêm produtos, incluindo software.

É um *framework* que define princípios gerais de qualidade, descreve os processo gerais de qualidade e estabelece os padrões organizacionais e os procedimentos que devem ser definidos.

Os processos essenciais da ISO 9001

Processos de entrega de produto	Processos de apoio
Aquisição de negócios	Gerenciamento de negócios
Projeto e desenvolvimento	Gerenciamento de inventário
Teste	Gerenciamento de fornecedores
Produção e entrega	Gerenciamento de configuração
Serviço e suporte	

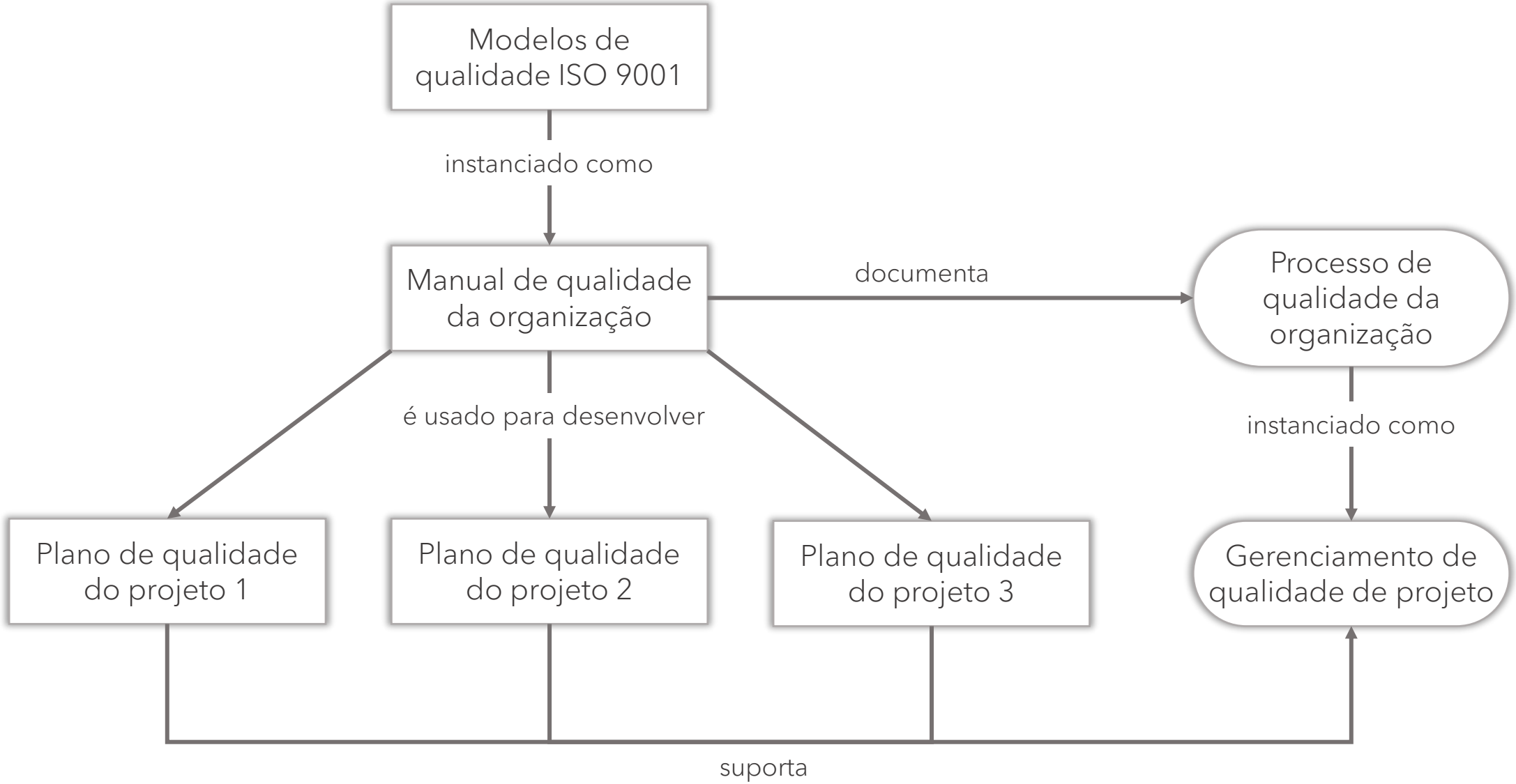
FONTE: SOMMERVILLE (2011)

O framework de normas ISO 9001

As normas ISO 9001 não definem ou prescrevem os processos de qualidade específicos que devem ser usados em uma empresa.

As empresas precisam estabelecer seus processos de qualidade, desde que atendam aos processos estabelecidos pela norma ISO 9001.

ISO 9001 e gerenciamento de qualidade



FONTE: SOMMERVILLE (2011)

Revisões e inspeções

Revisões e inspeções são atividades de controle de qualidade que verificam a qualidade dos entregáveis de projeto.

Revisões e inspeções

Revisões e inspeções são usadas junto com testes de programa, como parte do processo geral de validação e verificação de software.

Examina o software, sua documentação e os registros do processo para descobrir erros e omissões e verifica se os padrões de qualidade foram seguidos.

As avaliações de qualidade são baseadas em documentos que foram produzidos durante o processo de desenvolvimento de software.

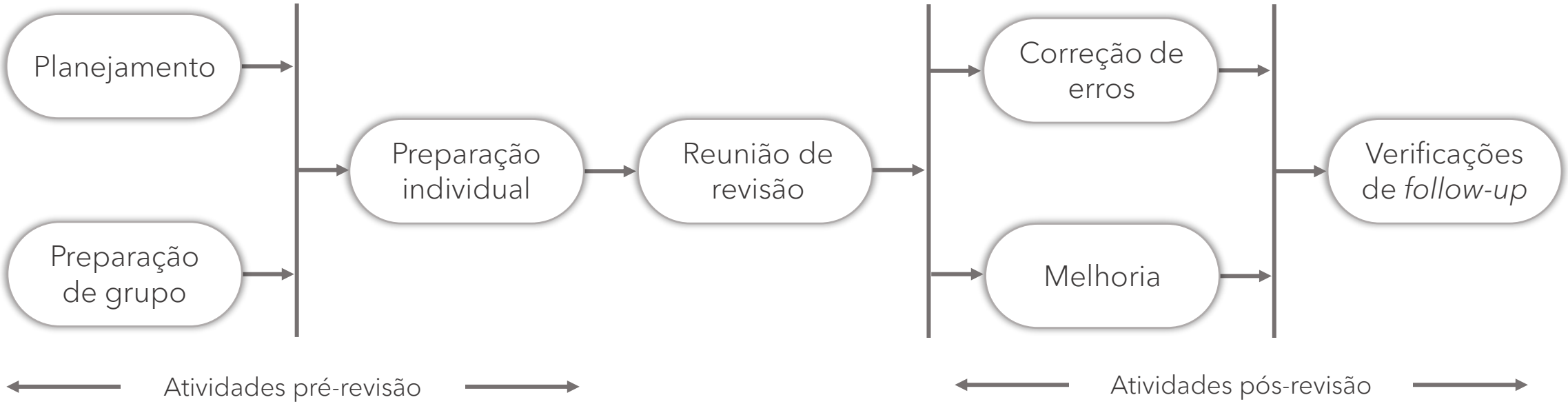
O objetivo das revisões e inspeções é melhorar a qualidade de software.

O processo de revisão

Fases da revisão:

- Atividades pré-revisão – planejamento e preparação da revisão;
- Reunião de revisão – autor do documento deve explanar sobre o mesmo para toda a equipe de revisão;
- Atividades pós-revisão – abordagem dos problemas levantados na reunião de revisão.

O processo de revisão de software



FONTE: SOMMERVILLE (2011)

Inspeções de programa

As inspeções são "revisões em pares" para encontrar *bugs* no programa que está sendo desenvolvido.

As inspeções podem descobrir problemas com testes e, assim, melhorar a eficácia desses testes em detectar *bugs* no programa.

Os defeitos podem ser erros lógicos, anomalias no código que podem indicar uma condição errada ou recursos que foram omitidos do código.

Frequentemente usa um *checklist* de erros comuns de programação.

Um *checklist* de inspeção

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 requirido após cada *case* em declarações *case*, este foi incluído?

FONTE: SOMMERVILLE (2011)

Um *checklist* de inspeção

Classe de defeito	Verificação de inspeção
Defeitos de entrada/saída	• Todas as variáveis de entrada são usadas? • Todas as variáveis de saída recebem um valor antes de serem emitidas? • Entradas inesperadas podem causar corrupção de dados?
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 compartilha?

FONTE: SOMMERVILLE (2011)

Um *checklist* de inspeção

Classe de defeito	Verificação de inspeção
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?

FONTE: SOMMERVILLE (2011)

Medições e métricas de software

A medição de software preocupa-se com a derivação de um valor número ou o perfil para um atributo de um componente de software, sistema ou processo.

Medições e métricas de software

O objetivo de medição de software é usá-la no lugar de revisões para fazer julgamentos sobre a qualidade de software.

A medição pode ser avaliada usando uma variedade de métricas.

Uma métrica é uma característica de um sistema, documentação ou processo de desenvolvimento que pode ser medido.

Exemplos de métricas:

- Tamanho de um produto em linhas de código;
- Número de defeitos relatados em um produto de software entregue;
- Número de pessoas/dia requerido para desenvolver um componente de sistema.

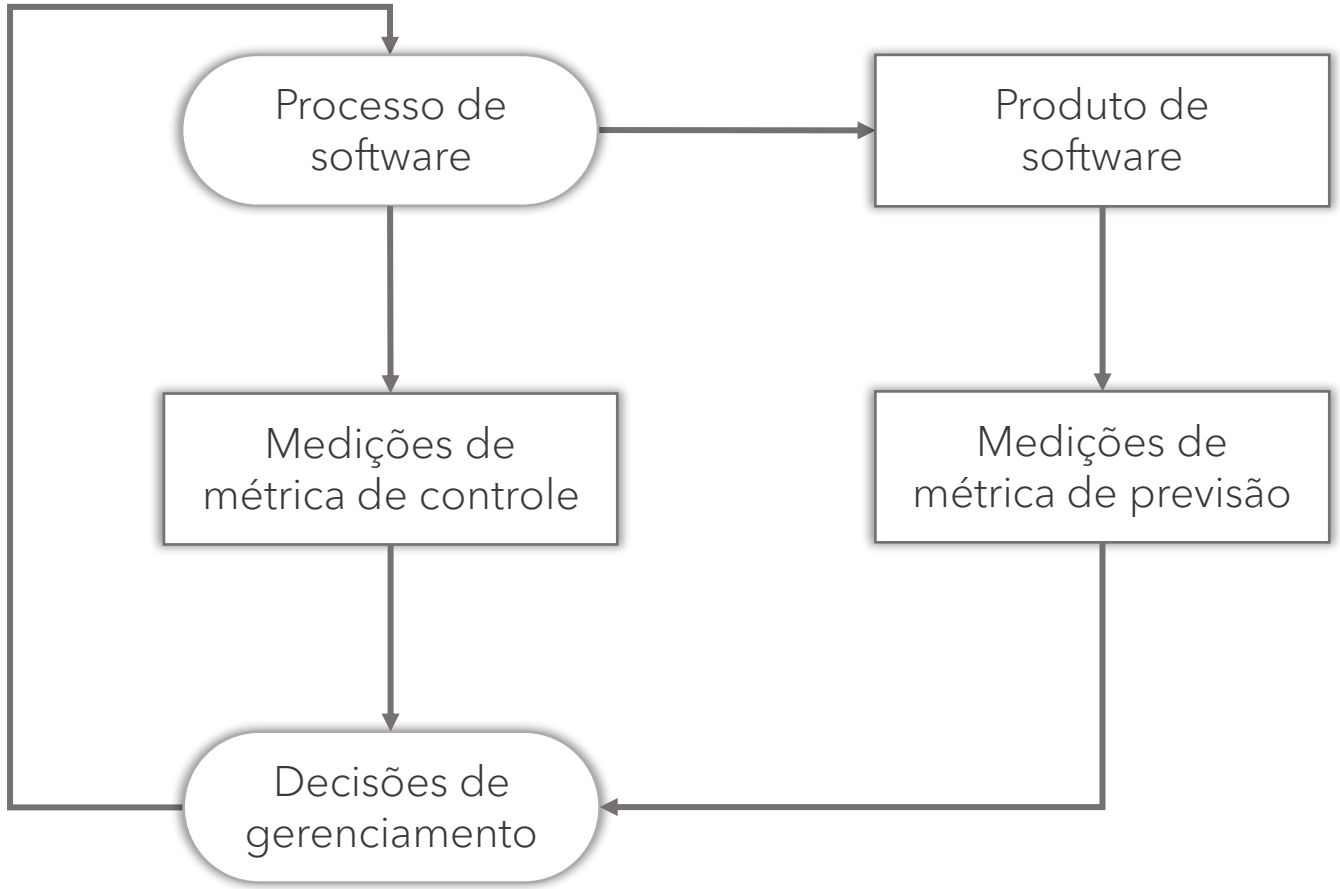
Medições e métricas de software

As métricas podem ser de **controle** ou de **previsão**.

Métricas de controle suportam os processos de gerenciamento, tais como o esforço médio e o tempo necessário para reparar defeitos relatados.

Métricas de previsão ajudam a prever as características do software, tais como o número de atributos e operações associadas com as classes de objeto.

Medições de previsão e de controle



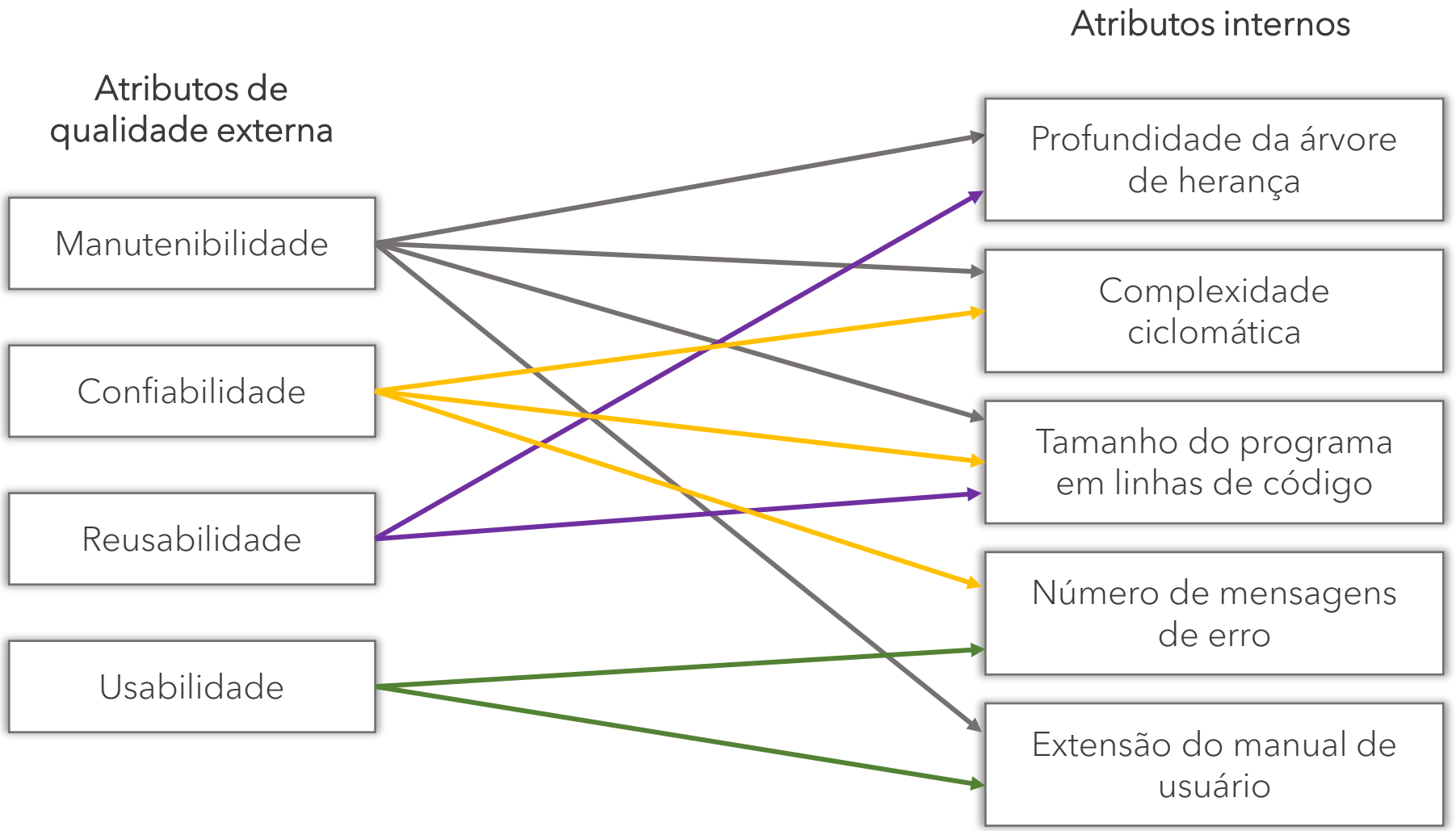
FONTE: SOMMERVILLE (2011)

Medições e métricas de software

As medições pode ser utilizadas:

- Para atribuir um valor aos atributos de qualidade de sistema – avalia os atributos de qualidade do sistema, como a manutenibilidade através da medição das características dos componentes de sistema;
- Para identificar os componentes de sistema cuja qualidade não atingiu o padrão – a medição pode identificar componentes individuais com características que se desviem da norma.

Relacionamentos entre os atributos internos e externos de software



FONTE: SOMMERVILLE (2011)

Medições e métricas de software

Razões que dificultam medições sistemáticas entre produtos e processos de software, avaliações de impacto de mudanças nos processos e ferramentas:

- É impossível quantificar o retorno sobre o investimento da introdução de um programa de métricas em organizações;
- Não existe um padrão para as métricas de software ou processos padronizados para medição e análise;
- Em sua maioria, os processos não são padronizados e/ou são mal definidos e controlados;
- Medição e métricas são baseadas em códigos e processos de desenvolvimento. Outros métodos, com ágeis, não são considerados.

Métricas do produto

As métricas de produto são métricas de previsão usadas para medir atributos internos de um sistema de software.

São divididos em duas classes:

- Métricas dinâmicas – são coletadas durante o teste ou após o sistema estar em uso.
- Métricas estáticas – são coletadas por meio de medições de representações do sistema, como o projeto, o programa ou a documentação.

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. *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.

FONTE: SOMMERVILLE (2011)

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.
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.

FONTE: SOMMERVILLE (2011)

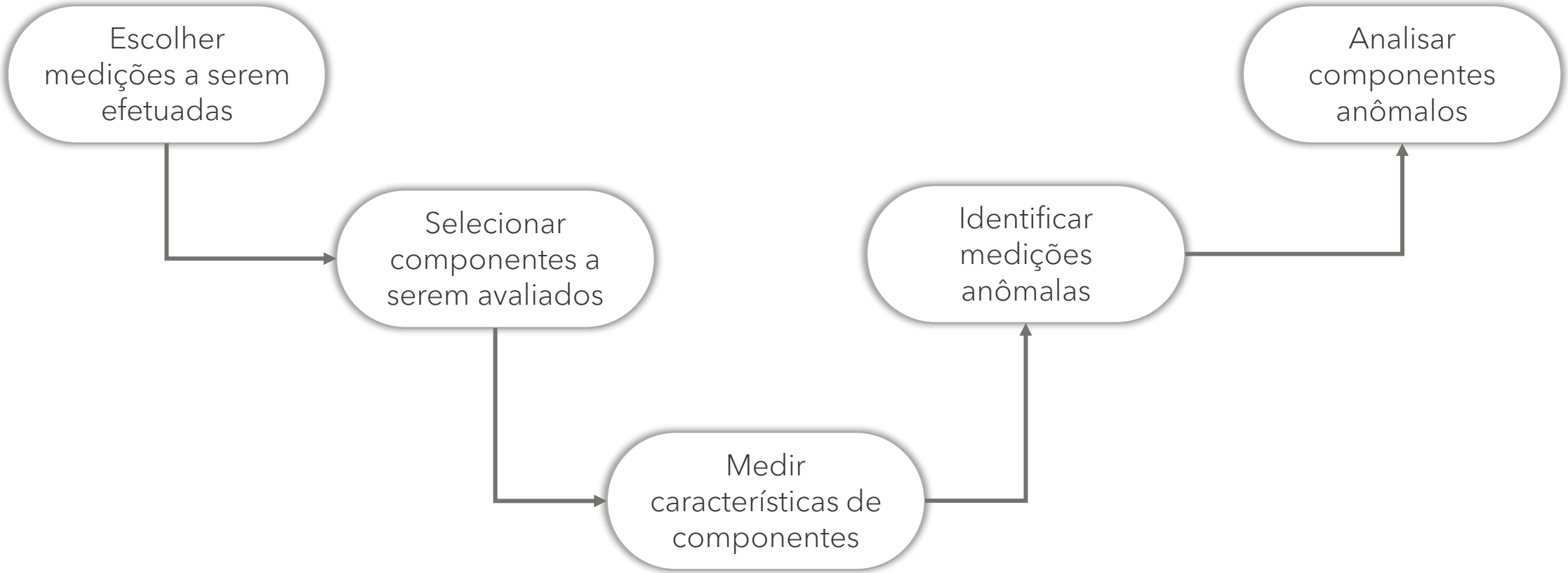
Análise de componentes de software

Cada componente de sistema pode ser analisado separadamente, usando uma variedade de métricas.

Os estágios são:

- Escolher medições a serem efetuadas.
- Selecionar componentes a serem avaliados.
- Medir características de componentes.
- Identificar medições anômalas.
- Analisar componentes anômalas.

O processo de medição de produto



FONTE: SOMMERVILLE (2011)

ISO 25000

É a evolução das ISO 9126, que define um modelo de qualidade para avaliação de produtos de softwares, e da ISO 14528, que define o processo de avaliação de produtos de software.

ISO 25000

A ISO 25000 possui cinco divisões:

- ISO 2500n – Divisão de Gestão da Qualidade
  - ISO 25000 – Fornece o modelo de arquitetura SQuaRe, terminologia, visão geral de documentos, usuários pretendidos e partes associadas da série, bem como modelos de referência.
  - ISO 25001 – Planejamento e Gerenciamento: gerenciamento da especificação e avaliação dos requisitos do produto de software.

- ISO 2501n – Divisão de Modelos de Qualidade
  - ISO 25010 – Modelos de qualidade de sistema e software;
  - ISO 25012 – Modelo de qualidade de dados;

ISO 25000

A ISO 25000 possui cinco divisões:

- ISO 2502n – Divisão de Medição de Qualidade
  - ISO 25020 – Guia e modelo de referência de medição;
  - ISO 25021 – Elementos de medida de qualidade;
  - ISO 25022 – Medição da qualidade em uso;
  - ISO 25023 – Medição da qualidade do sistema e do produto de software;
  - ISO 25024 – Medição da qualidade dos dados;

- ISO 2503n – Divisão de Requisitos de Qualidade
  - ISO 25030 – Requisitos de qualidade;

ISO 25000

A ISO 25000 possui cinco divisões:

- ISO 2504n – Divisão de Avaliação de Qualidade
  - ISO 25040 – Modelo e guia de referência de avaliação;
  - ISO 25041 – Guia de avaliação para desenvolvedores, adquirentes e avaliadores independentes;
  - ISO 25042 – Módulos de avaliação;
  - ISO 25045 – Módulos de avaliação para recuperabilidade;

Resumo

- O gerenciamento de qualidade de software visa assegurar que o software tenha um pequeno número de defeitos e que se adeque aos padrões de manutenibilidade, confiabilidade, portabilidade etc. em vigor.

- Inclui a definição de padrões para produtos e processos e o estabelecimento de processos para verificar se tais padrões foram seguidos.

- Os padrões de software são importantes para a garantia de qualidade, pois representam uma identificação das “melhores práticas”. Durante o desenvolvimento de software, os padrões fornecem uma base sólida para a construção de software de boa qualidade.

Resumo

- O conjunto de procedimentos de garantia de qualidade deve ser documentado em um manual de qualidade organizacional. Este pode ser baseado no modelo genérico para um manual de qualidade sugerido na norma ISO 9001.

- Revisões dos entregáveis de processo de software envolvem uma equipe de pessoas que verifica se os padrões de qualidade estão sendo seguidos. Revisões são técnicas largamente usadas para avaliação de qualidade.

- Em uma inspeção de programa ou revisão empares, uma pequena equipe verifica o código sistematicamente. Ela lê o código em detalhes procurando por possíveis erros e omissões. Em seguida, os problemas detectados são discutidos em uma reunião de revisão de código.

Resumo

- A medição de software pode ser usada para coletar dados quantitativos sobre o software e o processo de software. Você pode ser capaz de usar os valores das métricas de software que são coletadas para fazer inferências sobre a qualidade de produto e de processo.

- Métricas de qualidade de produto são particularmente úteis para realçar componentes anômalos que podem ter problemas de qualidade. Em seguida, esses componentes devem ser analisados em mais detalhes.

Atividade sugerida

Suponha que você trabalhe para uma organização que desenvolve produtos de banco de dados para indivíduos e empresas de pequeno porte. Essa organização está interessada na quantificação de seu desenvolvimento de software. Escreva um relatório sugerindo métricas adequadas e sugira como estas podem ser coletadas.