

# Resumão

## MD395

### Qualidade de Software

### Prof. Washington Almeida



#### ATENÇÃO!

Este é um **resumo do módulo**. Trata-se de um texto que também é baseado na transcrição consolidada da fala do professor(a), mas tem foco em resumir a parte técnica tratada ao longo do módulo.

Use-o caso tenha muito pouco tempo para estudar todo o conteúdo ou para revisá-lo rapidamente. Valem aqui as mesmas observações: não se trata de uma apostila técnica, mas um resumo baseado na transcrição consolidada da fala do professor.

Versão 1.0 (texto básico inicial).

## Sumário

Introdução.....	5
Capítulo I Conceitos Iniciais.....	6
1. Definição de qualidade de software .....	6
2. Definição de engenharia de software .....	6
3. Definição de qualidade.....	6
4. Aspecto organizacional.....	6
5. Aspecto de projeto .....	6
Conclusão.....	6
Capítulo II Técnicas de gerenciamento da qualidade .....	7
1. Técnicas de gerenciamento da qualidade .....	7
2. Garantia da qualidade .....	7
3. Controle da qualidade .....	7
4. Controle versus garantia .....	7
Conclusão.....	8
Capítulo III Qualidade de software.....	9
1. Processos baseados em planos .....	9
2. Equipe de qualidade .....	9
3. Equipe independente .....	9
Conclusão.....	9
Capítulo IV Plano de Qualidade.....	10
1. Introdução.....	10
2. Planos do produto .....	10
3. Descrições de processos .....	10
4. Metas de qualidade .....	10
5. Riscos.....	10
Conclusão.....	10
Capítulo V Padrões de Software.....	11
1. Definição de padrões de software .....	11
2. Aspecto organizacional.....	11
3. Padrões de produto .....	11
4. Padrões de processo.....	11
Conclusão.....	11
Capítulo VI Padrões ISO 9001.....	12
1. Processos principais.....	12
2. ISO 9001 e gerenciamento de qualidade .....	12

<b>3. Certificação ISO 9001</b>	12
<b>Conclusão</b>	12
<b>Capítulo VII Revisões</b>	13
1. Definição de revisão	13
2. Finalidade da revisão	13
3. Revisão de qualidade	13
4. Processo de revisão	13
5. Inspeção de programa	13
6. Checklist inspeção	13
7. Relevância das inspeções	14
<b>Conclusão</b>	14
<b>Capítulo VIII Métodos Ágeis</b>	15
1. Definição dos métodos ágeis	15
2. Características dos métodos ágeis	15
3. Qualidade de software nos métodos ágeis	15
4. Qualidade versus agilidade	15
<b>Conclusão</b>	15
<b>Capítulo IX Boas práticas</b>	16
1. Verificar antes do check-in	16
2. Nunca quebre a construção	16
3. O código não tem dono	16
4. Revisão em pares	16
5. Revisão por equipe externa	16
6. Vários times distribuídos geograficamente	16
7. Sistemas de longa vida útil	16
8. Adaptação da qualidade nos métodos ágeis	16
<b>Conclusão</b>	17
<b>Capítulo X Medição de Software</b>	18
1. Definição de medida de software	18
2. Com que se preocupa a medição de software	18
3. Comparando os valores medidos	18
4. Medidas diretas	18
5. Medidas indiretas	18
<b>Conclusão</b>	18
<b>Capítulo XI Métricas de Software</b>	19
1. Definição de métricas de software	19

<b>2. Exemplos de métricas</b>	19
<b>3. Métricas</b>	19
De controle (ou processo)	19
De previsão (ou produto)	19
<b>4. Tipos de métricas de produto</b>	19
Métricas dinâmicas	19
Métricas estáticas	19
<b>5. Medida versus métrica</b>	19
<b>Conclusão</b>	19
<b>Capítulo XII Atributos da Qualidade</b>	20
<b>1. Definição de atributo da qualidade</b>	20
<b>2. Manutenibilidade</b>	20
<b>3. Confiabilidade</b>	20
<b>4. Reusabilidade</b>	20
<b>5. Usabilidade</b>	20
<b>Conclusão</b>	20



# Introdução

Este resumo do módulo aborda os fundamentos e práticas essenciais relacionados à qualidade de software. Iniciamos com uma exploração dos Conceitos Iniciais, estabelecendo a base para entender a importância da qualidade no desenvolvimento de software. Em seguida, mergulhamos nas Técnicas de Gerenciamento da Qualidade, destacando a necessidade de uma abordagem estruturada para garantir produtos de alta qualidade. A Qualidade de Software é discutida em profundidade, enfatizando sua natureza dinâmica e complexa, influenciada por fatores técnicos, contextuais e humanos.

O Plano de Qualidade é fundamental, definindo os atributos mais importantes de qualidade para o software em desenvolvimento. Este plano considera aspectos como descrição do produto, mercado-alvo, processos e padrões de desenvolvimento, metas de qualidade e gerenciamento de riscos. A seção sobre Padrões de Software destaca a importância de seguir diretrizes estabelecidas para garantir a consistência e a confiabilidade do software. Os Padrões ISO 9001 são abordados, ressaltando sua relevância no cenário global de qualidade de software.

As Revisões são essenciais para garantir que o software atenda aos requisitos de qualidade esperados. O módulo também aborda os Métodos Ágeis, destacando sua flexibilidade e adaptabilidade, especialmente em cenários de desenvolvimento dinâmico. As Boas Práticas são enfatizadas, garantindo que os desenvolvedores sigam diretrizes que promovam a qualidade em todas as etapas do ciclo de vida do software.

A Medição de Software é discutida, sublinhando a importância de quantificar atributos para avaliar a qualidade e eficácia dos processos, ferramentas e métodos. Isso nos leva ao tópico de Métricas de Software, onde características mensuráveis de um sistema de software são exploradas, desde medidas diretas, como custo e esforço, até medidas indiretas relacionadas à funcionalidade e qualidade. Finalmente, concluímos com uma discussão sobre os Atributos da Qualidade, enfatizando a avaliação de características como confiabilidade e manutenibilidade.

Este resumo do módulo oferece uma visão abrangente dos principais tópicos e práticas relacionados à qualidade de software, preparando os leitores para uma jornada profunda e enriquecedora neste campo essencial da engenharia de software.

# Capítulo I

## Conceitos Iniciais

O resumo do módulo inicia-se com os conceitos fundamentais que formam a base para a compreensão da qualidade de software. Estes conceitos são essenciais para entendermos o panorama geral da engenharia de software e como a qualidade se encaixa nesse contexto. Além disso, abordaremos a importância dos aspectos organizacionais e de projeto na garantia da qualidade do software.

### 1. Definição de qualidade de software

A qualidade de software refere-se à medida em que um software atende aos requisitos funcionais e de desempenho definidos, é livre de defeitos e é adequado para seu uso pretendido.

### 2. Definição de engenharia de software

A engenharia de software é uma disciplina que envolve todos os aspectos da produção de software, desde a concepção inicial até a manutenção do sistema.

### 3. Definição de qualidade

A qualidade é uma característica intrínseca ou propriedade de algo que pode ser avaliada em termos de sua capacidade de satisfazer as necessidades estabelecidas ou implícitas.

### 4. Aspecto organizacional

No âmbito organizacional, a qualidade do software é amplamente defendida, especialmente em projetos complexos que seguem abordagens baseadas em planos. Aqui, há uma preocupação constante em estabelecer processos e padrões para garantir produtos de alta qualidade. A equipe de gerenciamento da qualidade desempenha um papel central, definindo os processos a serem seguidos, os padrões a serem aplicados à documentação e as metas de qualidade do projeto.

### 5. Aspecto de projeto

A qualidade do projeto é avaliada em cada etapa, garantindo que o resultado final esteja em conformidade com os padrões estabelecidos. Essa abordagem é especialmente relevante em projetos complexos, onde a organização é fundamental para garantir a qualidade do processo e do produto.

### Conclusão

Os conceitos iniciais apresentados neste capítulo fornecem uma base sólida para a compreensão da qualidade de software. Eles destacam a importância de se ter uma definição clara de qualidade e como ela se relaciona com a engenharia de software. Além disso, os aspectos organizacionais e de projeto desempenham um papel crucial na garantia da qualidade, estabelecendo padrões e processos que orientam o desenvolvimento de software de alta qualidade.

# Capítulo II

## Técnicas de gerenciamento da qualidade

A gestão da qualidade é uma parte integral da engenharia de software, e sua origem está profundamente enraizada nos métodos de garantia e controle de qualidade utilizados em outras indústrias. No contexto da engenharia de software, a distinção entre garantia de qualidade e controle de qualidade é crucial para entender o processo de desenvolvimento e entrega de software de alta qualidade.

### 1. Técnicas de gerenciamento da qualidade

A qualidade de software tem suas raízes em métodos de garantia e controle de qualidade empregados em outras indústrias, adaptados à realidade da engenharia de software. Essa adaptação é essencial devido à natureza única e complexa do desenvolvimento de software, que difere significativamente da produção tradicional.

### 2. Garantia da qualidade

A garantia de qualidade refere-se a funções gerenciais, como auditorias e relatórios, que acompanham o processo de construção do software para garantir que os padrões estabelecidos estejam sendo seguidos. No contexto do software, a garantia de qualidade estende-se além da fase de desenvolvimento, focando em funções gerenciais que incluem auditoria e relato, buscando garantir que os processos e produtos atendam aos padrões estabelecidos mesmo após o lançamento do software.

### 3. Controle da qualidade

O controle de qualidade, por outro lado, envolve inspeções, revisões e testes que ocorrem durante todo o ciclo de vida do software. Está intimamente relacionado ao processo de produção, com o objetivo de monitorar a qualidade durante sua construção. O controle de qualidade se preocupa com a verificação e validação contínuas para assegurar que o software esteja em conformidade com os padrões de qualidade estabelecidos.

### 4. Controle versus garantia

Embora ambos sejam essenciais para a qualidade do software, a diferença entre controle e garantia de qualidade no setor de software é mais sutil do que em comparação com setores de manufatura. A garantia de qualidade busca assegurar que os processos e produtos ao longo do ciclo de vida do projeto estejam em conformidade com os padrões, enquanto o controle de qualidade se estende durante o desenvolvimento do software, envolvendo inspeções, revisões e testes para verificar se o software atende aos padrões definidos.

### Conclusão

A gestão da qualidade no desenvolvimento de software é uma combinação de garantia e controle de qualidade. Ambos são essenciais para garantir que o software atenda aos padrões estabelecidos e às expectativas dos usuários. A compreensão clara da diferença entre essas duas abordagens e sua aplicação adequada é fundamental para a entrega de software de alta qualidade.





## Capítulo III

# Qualidade de software

A qualidade de software é uma métrica essencial que determina o valor e a eficácia de um produto de software. Para alcançar a qualidade desejada, é fundamental ter processos bem definidos, uma equipe dedicada à qualidade e uma abordagem independente para avaliar o software.

### 1. Processos baseados em planos

Os processos baseados em planos são essenciais para garantir que o software seja desenvolvido de acordo com os padrões estabelecidos. Esses processos envolvem a definição clara de etapas, metas e critérios de aceitação, garantindo que o software atenda às expectativas e requisitos dos usuários.

### 2. Equipe de qualidade

Uma equipe dedicada à qualidade é responsável por garantir que o software atenda aos padrões de qualidade estabelecidos. Esta equipe trabalha em estreita colaboração com os desenvolvedores, conduzindo revisões, testes e validações para assegurar que o software seja de alta qualidade e livre de defeitos.

### 3. Equipe independente

Uma equipe independente, separada da equipe de desenvolvimento, é essencial para avaliar o software de uma perspectiva imparcial. Esta equipe realiza avaliações independentes, inspeções e testes, garantindo que o software atenda aos padrões de qualidade sem qualquer viés.

### Conclusão

A qualidade do software não é apenas uma métrica, mas uma necessidade para garantir a satisfação do usuário e o sucesso do produto no mercado. Com processos bem definidos, uma equipe dedicada à qualidade e avaliações independentes, é possível alcançar um software de alta qualidade que atenda às necessidades e expectativas dos usuários.

## Capítulo IV

# Plano de Qualidade

O Plano de Qualidade é uma ferramenta essencial no desenvolvimento de software, delineando as metas, processos e riscos associados ao projeto. Ele serve como um guia para garantir que o software atenda aos padrões de qualidade desejados.

### 1. Introdução

A introdução ao Plano de Qualidade fornece uma visão geral dos objetivos e do escopo do plano, estabelecendo a base para os detalhes subsequentes.

### 2. Planos do produto

Os planos do produto detalham as especificações, requisitos e critérios de aceitação do software. Eles fornecem uma estrutura clara para o desenvolvimento e teste do software, garantindo que ele atenda às expectativas dos usuários.

### 3. Descrições de processos

As descrições de processos detalham os métodos e procedimentos que serão seguidos durante o desenvolvimento do software. Elas garantem que o software seja desenvolvido de acordo com os padrões estabelecidos e que os processos sejam consistentes e eficazes.

### 4. Metas de qualidade

As metas de qualidade definem os padrões que o software deve atender. Elas fornecem uma métrica clara para avaliar o sucesso do software e garantir que ele atenda às expectativas dos usuários.

### 5. Riscos

Identificar e gerenciar riscos é crucial para o sucesso de qualquer projeto de software. O Plano de Qualidade deve detalhar os riscos potenciais associados ao projeto e as estratégias para mitigá-los.

### Conclusão

O Plano de Qualidade é mais do que apenas um documento; é uma ferramenta vital que guia o desenvolvimento do software, garantindo que ele atenda aos padrões de qualidade desejados. Com metas claras, processos bem definidos e uma compreensão dos riscos, o Plano de Qualidade assegura que o software seja de alta qualidade e atenda às necessidades dos usuários.

# Capítulo V

## Padrões de Software

Os padrões de software desempenham um papel fundamental na busca pela qualidade do software. Eles encapsulam experiências passadas, fornecem um referencial para avaliar a qualidade e facilitam a continuidade nos projetos. A adaptação dos padrões ao contexto é essencial para garantir que as melhores práticas sejam aplicadas de maneira apropriada a cada situação.

### 1. Definição de padrões de software

Os padrões de software capturam a sabedoria acumulada que possui valor para a organização. Eles se baseiam no conhecimento adquirido por meio de tentativa e erro, ao longo do tempo, sobre as melhores práticas e abordagens adequadas para a empresa.

### 2. Aspecto organizacional

Os padrões fornecem um referencial para definir o que se entende por qualidade em um contexto específico. Usando padrões, é possível estabelecer uma base para avaliar se um nível aceitável de qualidade foi alcançado.

### 3. Padrões de produto

Existem dois tipos de padrões na engenharia de software: padrões de produto e padrões de processo. Os padrões de produto estão relacionados às características específicas do software em desenvolvimento, incluindo padrões de documentação, formatação de código, entre outros.

### 4. Padrões de processo

Os padrões de processo definem como as atividades de desenvolvimento devem ser conduzidas, abrangendo desde a especificação do projeto até a validação e lançamento do software, englobando também práticas de controle de qualidade.

### Conclusão

Os padrões de software são essenciais para garantir a qualidade e consistência no desenvolvimento de software. Eles fornecem diretrizes claras e práticas recomendadas que ajudam as organizações a desenvolverem softwares de alta qualidade que atenda às necessidades dos usuários. A compreensão e aplicação adequada desses padrões são cruciais para o sucesso de qualquer projeto de software.

# Capítulo VI

## Padrões ISO 9001

A ISO 9001 é um padrão reconhecido e aplicado em muitas organizações para assegurar o controle de qualidade geral. Embora contribua para o controle de qualidade organizacional, ela não é suficiente para garantir a qualidade de software em si.

### 1. Processos principais

A ISO 9001 aborda processos principais, como os de entrega de produtos e de apoio. Os processos de entrega abrangem desde a aquisição de negócios até a produção, entrega, teste e suporte.

### 2. ISO 9001 e gerenciamento de qualidade

A ISO 9001 estabelece princípios gerais de qualidade, descreve processos de qualidade de maneira geral e define normas e procedimentos organizacionais que devem ser estabelecidos.

### 3. Certificação ISO 9001

A aplicação da ISO 9001 tem sido amplamente reconhecida, com muitas organizações certificadas por ela. No entanto, a certificação ISO 9001 não garante, por si só, que a qualidade do software produzido por uma empresa certificada seja superior à de uma não certificada.

### Conclusão

A ISO 9001 é um padrão essencial para assegurar o controle de qualidade em diversas organizações. No entanto, para garantir a qualidade de software, é necessário combinar a ISO 9001 com padrões específicos de engenharia de software. A compreensão e aplicação adequada da ISO 9001 são cruciais para o sucesso de qualquer projeto de software.

# Capítulo VII

## Revisões

As revisões são uma parte essencial do processo de desenvolvimento de software, permitindo que as organizações identifiquem e corrijam problemas antes que eles se tornem mais graves. Através das revisões, é possível garantir que o software atenda aos padrões de qualidade desejados e que os requisitos dos usuários sejam atendidos.

### 1. Definição de revisão

A revisão é um processo sistemático que envolve a avaliação de um produto de software ou de um processo de desenvolvimento para identificar problemas e áreas de melhoria.

### 2. Finalidade da revisão

A principal finalidade da revisão é garantir que o software atenda aos padrões de qualidade desejados e que os requisitos dos usuários sejam atendidos. Através das revisões, é possível identificar e corrigir problemas antes que eles se tornem mais graves.

### 3. Revisão de qualidade

A revisão de qualidade é um processo que envolve a avaliação do software para garantir que ele atenda aos padrões de qualidade desejados. Este processo é essencial para garantir que o software seja de alta qualidade e que atenda às necessidades dos usuários.

### 4. Processo de revisão

O processo de revisão envolve várias etapas, incluindo a identificação de problemas, a avaliação das soluções propostas e a implementação das correções necessárias. Este processo é essencial para garantir que o software seja de alta qualidade e que atenda às necessidades dos usuários.

### 5. Inspeção de programa

A inspeção de programa é um processo que envolve a avaliação detalhada do código fonte do software para identificar problemas e áreas de melhoria. Este processo é essencial para garantir que o software seja de alta qualidade e que atenda às necessidades dos usuários.

### 6. Checklist inspeção

O checklist de inspeção é uma ferramenta que ajuda os revisores a identificarem problemas e áreas de melhoria no software. Esta ferramenta é essencial para garantir que o software seja de alta qualidade e que atenda às necessidades dos usuários.

### 7. Relevância das inspeções

Tanto as revisões quanto as inspeções são atividades essenciais para garantir a qualidade do software ao longo de todo o ciclo de desenvolvimento. Estas práticas contribuem para a identificação precoce de erros e o aprimoramento contínuo dos processos, resultando em um produto final de maior qualidade e confiabilidade.

### Conclusão

As revisões são uma parte essencial do processo de desenvolvimento de software, permitindo que as organizações identifiquem e corrijam problemas antes que eles se tornem mais graves. Através das revisões, é possível garantir que o software atenda aos padrões de qualidade desejados e que os requisitos dos usuários sejam atendidos.



## Capítulo VIII

# Métodos Ágeis

Os métodos ágeis têm ganhado destaque no desenvolvimento de software devido à sua flexibilidade e foco na entrega contínua de valor. Eles representam uma abordagem adaptativa que prioriza a colaboração entre equipes e clientes, entregas frequentes e a capacidade de responder rapidamente às mudanças.

### 1. Definição dos métodos ágeis

Os métodos ágeis são abordagens de desenvolvimento de software que enfatizam a colaboração entre equipes e clientes, entregas frequentes de software funcional e a capacidade de se adaptar rapidamente às mudanças.

### 2. Características dos métodos ágeis

Os métodos ágeis são caracterizados por iterações curtas, feedback contínuo, foco no cliente e adaptabilidade. Eles buscam entregar valor de forma contínua e responder rapidamente às necessidades do projeto.

### 3. Qualidade de software nos métodos ágeis

Nos métodos ágeis, a qualidade é integrada ao processo de desenvolvimento. As equipes ágeis podem incorporar um papel de Garantia de Qualidade (QA) para garantir a aderência às práticas de qualidade.

### 4. Qualidade versus agilidade

A abordagem ágil para a gestão da qualidade pode ser altamente eficaz em desenvolvimento interno, onde a empresa controla a especificação do software. No entanto, em cenários mais complexos, a adaptação das práticas ágeis para incluir alguma formalização pode ser necessária.

### Conclusão

Os métodos ágeis oferecem uma abordagem flexível e adaptativa ao desenvolvimento de software, permitindo que as equipes respondam rapidamente às mudanças e entreguem valor contínuo aos clientes. Integrar práticas de qualidade nessa abordagem garante que o software desenvolvido seja robusto e atenda às expectativas dos usuários.

# Capítulo IX

## Boas práticas

As boas práticas no desenvolvimento de software são diretrizes e procedimentos recomendados que visam garantir a qualidade e eficiência do software produzido. Elas são essenciais para garantir que o software seja confiável, eficiente e atenda às expectativas dos usuários.

### 1. Verificar antes do check-in

Antes de fazer o check-in do código em um sistema de controle de versão, é essencial verificar se o código está correto, se atende aos padrões estabelecidos e se não introduz novos erros.

### 2. Nunca quebre a construção

Evitar quebrar a construção é fundamental. Se o código introduzido causar falhas, ele deve ser corrigido imediatamente para não afetar o trabalho de outros desenvolvedores.

### 3. O código não tem dono

O código pertence ao projeto e não a um desenvolvedor específico. Todos devem ter a liberdade de revisar, modificar e melhorar o código conforme necessário.

### 4. Revisão em pares

A revisão em pares é uma prática onde o código escrito por um desenvolvedor é revisado por outro. Isso ajuda a identificar e corrigir erros, melhorar a qualidade do código e compartilhar conhecimento entre a equipe.

### 5. Revisão por equipe externa

Ter uma equipe externa revisando o código pode trazer uma perspectiva fresca e identificar problemas que a equipe interna pode ter perdido.

### 6. Vários times distribuídos geograficamente

Trabalhar com equipes distribuídas geograficamente pode trazer desafios de comunicação e coordenação. No entanto, com as ferramentas e práticas adequadas, é possível garantir a qualidade e eficiência do desenvolvimento.

### 7. Sistemas de longa vida útil

Sistemas que são esperados para ter uma longa vida útil requerem práticas robustas de qualidade para garantir que continuem atendendo às necessidades dos usuários ao longo do tempo.

### 8. Adaptação da qualidade nos métodos ágeis

Nos métodos ágeis, a qualidade é integrada ao processo de desenvolvimento. Em cenários mais complexos, como sistemas distribuídos ou sistemas de longa vida útil, a adaptação das práticas ágeis para incluir alguma formalização pode ser necessária.



### Conclusão

Adotar boas práticas no desenvolvimento de software é essencial para garantir a qualidade, eficiência e confiabilidade do software produzido. Essas práticas ajudam as equipes a evitar erros comuns, melhorar a qualidade do código e garantir que o software atenda às expectativas dos usuários.



# Capítulo X

## Medição de Software

A medição de software é uma área da engenharia de software que se concentra em coletar, analisar e interpretar dados relacionados ao software e seus processos de desenvolvimento. Ela é essencial para entender o desempenho, a qualidade e a eficácia do software e dos processos utilizados para desenvolvê-lo.

### 1. Definição de medida de software

A medida de software refere-se à quantificação de atributos específicos do software ou de seus processos de desenvolvimento. Essas medidas podem ser diretas, como linhas de código, ou indiretas, como complexidade do código.

### 2. Com que se preocupa a medição de software

A medição de software preocupa-se em coletar dados que podem ser usados para avaliar a qualidade do software, a eficiência dos processos de desenvolvimento e a satisfação do usuário. Ela também pode ajudar a identificar áreas de melhoria e a tomar decisões informadas sobre o projeto.

### 3. Comparando os valores medidos

Comparar os valores medidos permite às equipes entenderem o desempenho do software em relação a benchmarks ou padrões estabelecidos. Isso pode ajudar a identificar áreas de melhoria e a garantir que o software atenda aos padrões de qualidade desejados.

### 4. Medidas diretas

Medidas diretas são quantificações diretas de atributos do software, como o número de linhas de código, o número de erros ou o tempo de resposta.

### 5. Medidas indiretas

Medidas indiretas são derivadas de outras medidas e podem incluir métricas como complexidade do código, eficiência do processo ou satisfação do usuário.

### Conclusão

A medição de software é uma ferramenta poderosa que permite às equipes entender o desempenho e a qualidade do software. Ao coletar e analisar dados, as equipes podem tomar decisões informadas, melhorar a qualidade e a eficiência do desenvolvimento e garantir que o software atenda às expectativas dos usuários.

# Capítulo XI

## Métricas de Software

As métricas de software são ferramentas essenciais para avaliar a qualidade, eficiência e desempenho do software e dos processos utilizados para desenvolvê-lo. Elas fornecem uma quantificação objetiva de atributos específicos do software, permitindo que as equipes avaliem o desempenho em relação a padrões ou benchmarks.

### 1. Definição de métricas de software

Métricas de software são medidas quantitativas que representam atributos específicos do software ou de seus processos de desenvolvimento. Elas podem ser usadas para avaliar a qualidade, eficiência e desempenho do software.

### 2. Exemplos de métricas

Exemplos comuns de métricas incluem linhas de código, complexidade ciclomática, taxa de defeitos e tempo de resposta.

### 3. Métricas

#### De controle (ou processo)

Métricas de controle são usadas para avaliar e melhorar os processos de desenvolvimento. Elas podem incluir métricas como eficiência do processo, tempo de ciclo e satisfação do cliente.

#### De previsão (ou produto)

Métricas de previsão são usadas para prever atributos específicos do software, como sua qualidade, desempenho ou confiabilidade.

### 4. Tipos de métricas de produto

#### Métricas dinâmicas

Métricas dinâmicas são coletadas quando o software está em execução e podem incluir métricas como tempo de resposta, taxa de defeitos ou utilização de recursos.

#### Métricas estáticas

Métricas estáticas são coletadas sem executar o software e podem incluir métricas como linhas de código ou complexidade ciclomática.

### 5. Medida versus métrica

Enquanto uma medida é uma quantificação direta de um atributo, uma métrica é uma interpretação ou análise dessa medida. Por exemplo, o número de linhas de código é uma medida, enquanto a complexidade ciclomática, que pode ser derivada dessa medida, é uma métrica.

### Conclusão

Ao utilizar métricas de forma eficaz, as equipes podem garantir que o software atenda às expectativas dos usuários e alcance padrões de qualidade elevados.

## Capítulo XII

# Atributos da Qualidade

Os atributos da qualidade referem-se às características específicas do software que determinam sua qualidade. Eles são essenciais para entender como o software atende às expectativas dos usuários e aos padrões de qualidade estabelecidos.

### 1. Definição de atributo da qualidade

Atributos da qualidade são características específicas do software que podem ser medidas e avaliadas para determinar a qualidade do software.

### 2. Manutenibilidade

A manutenibilidade refere-se à facilidade com que o software pode ser modificado para corrigir defeitos, melhorar seu desempenho ou adaptá-lo a um ambiente alterado.

### 3. Confiabilidade

A confiabilidade é a capacidade do software de funcionar corretamente e sem falhas em condições especificadas.

### 4. Reusabilidade

A reusabilidade refere-se à capacidade de usar componentes de software em diferentes aplicações ou em diferentes partes da mesma aplicação.

### 5. Usabilidade

A usabilidade é a facilidade com que os usuários podem usar o software para alcançar objetivos específicos com eficácia, eficiência e satisfação.

### Conclusão

Os atributos da qualidade são essenciais para avaliar e garantir a qualidade do software. Ao entender e medir esses atributos, as equipes podem garantir que o software atenda às expectativas dos usuários e alcance padrões de qualidade elevados.