

### Aula 3: Qualidade de teste de software

## Apresentação

---

Esta aula tem como proposta discutir a importância dos testes de software, apresentando sua evolução histórica desde as décadas de 1950 e 1960, quando surgiram as primeiras linguagens de programação ditas modernas, algumas delas ainda em uso até hoje, como por exemplo o Cobol.

Neste contexto, apresentaremos algumas das principais características e exigências dos projetos de software baseados em processos de controle da qualidade.

## Objetivos

---

- Esclarecer por que é necessário ter qualidade nos testes;
- Identificar nos softwares as suas características e utilizar um teste adequado para eles;
- Relatar o que deve ser entregue como garantia para o teste com o cliente.



# Histórico

No início do desenvolvimento dos softwares, quando só existia a função de programador, que era exercida por poucos, não havia atividades de testes no processo do seu desenvolvimento. Na verdade, antes dos anos 1970, não havia nem processo definido de desenvolvimento de software.

## Comentário

O que ocorria nessa época?

- Tendo em vista que os erros ocorriam, após o software estar pronto, o próprio programador percorria o código para solucionar possíveis erros;
- Os testes eram feitos pelo próprio usuário.

Já no final dos 1960 e início dos 1970, surgiu a "programação estruturada", que recomendava, basicamente, evitar o uso de **goto**, ou seja, desvio incondicional. Há um consenso entre os programadores que o desvio incondicional é um mau estilo de programação, que gera códigos com baixa qualidade.

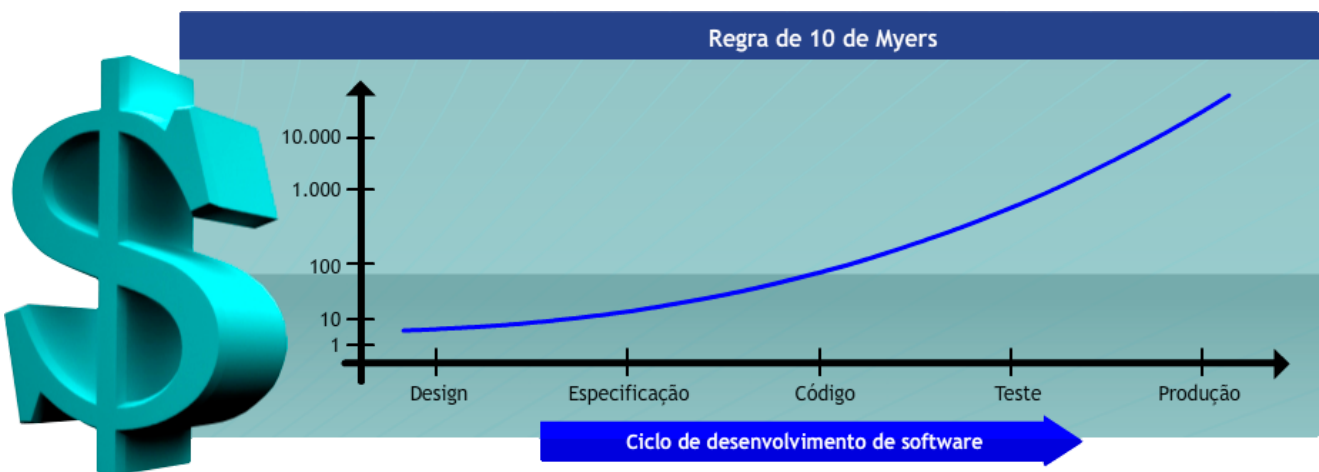
## Comentário

Nessa época, surgiram, então, os primeiros conceitos de engenharia de software, adotados, principalmente, como modelo nos cursos de Exatas nas universidades em todo o mundo. Assim, os primeiros procedimentos de testes passaram a ser usados, porém de forma bastante tímida.

Em 1979, Genford J. Myers, autor do livro *The Art of Software Testing*, apresentou um trabalho pioneiro e profundo sobre um processo de teste de software. Foi criador de termos muito usados como “caixa branca”, “caixa preta” e “caso de teste”.

Myers também ficou conhecido pela Regra de 10 de Myers, que mostra que **“quanto mais tarde os defeitos forem encontrados, tanto mais caro será corrigi-los”**.

Entenda graficamente a regra de 10 de Myers:

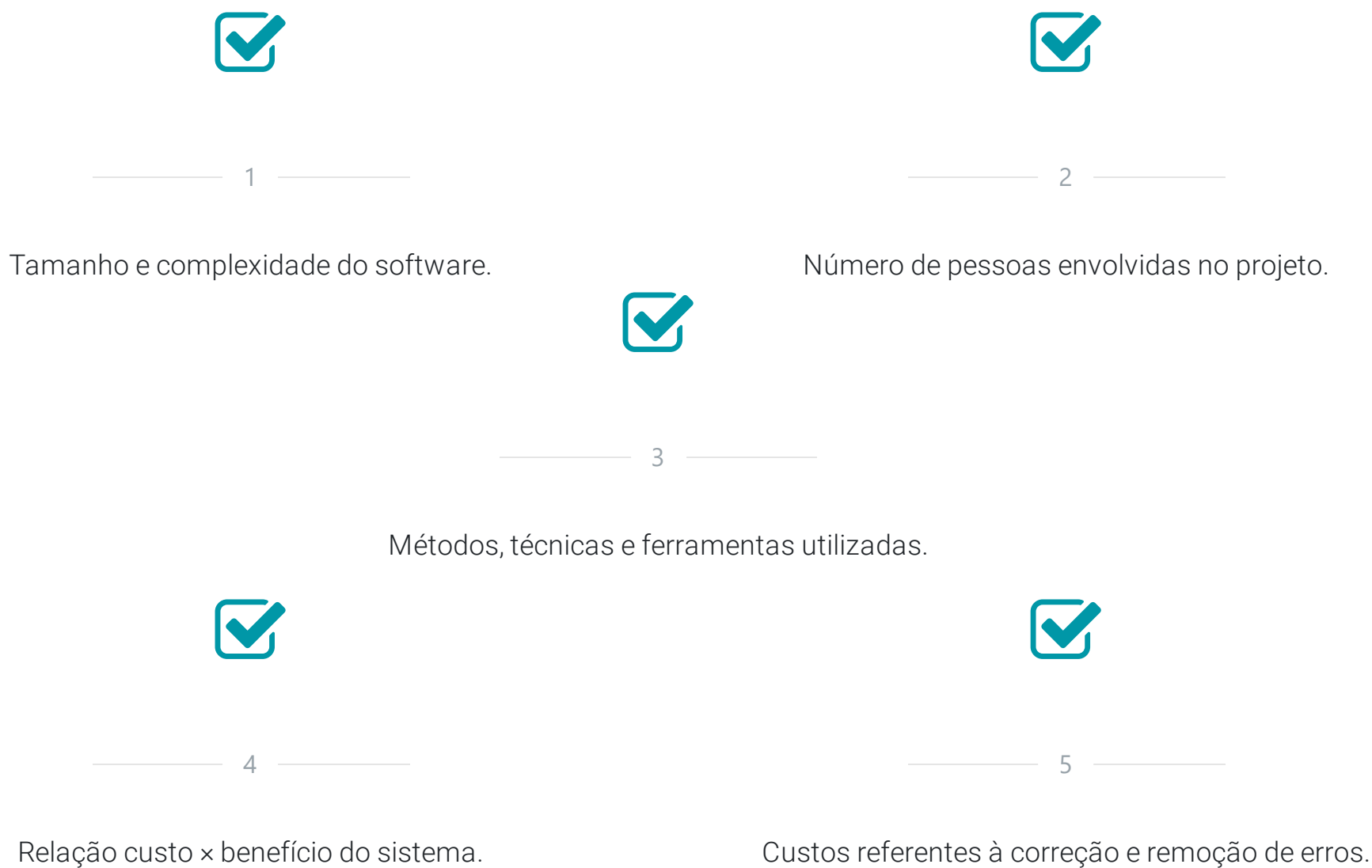


# Qualidade de software

---

Nos anos 1980, surgiu o conceito de qualidade de software, processo fortemente relacionado à conformidade com requisitos e à satisfação do cliente, que envolve a delimitação do escopo de um sistema e a volatilidade dos requisitos, lugar-comum no desenvolvimento de software.

Alguns fatores afetam o desenvolvimento do software e influenciam na avaliação do usuário, como:



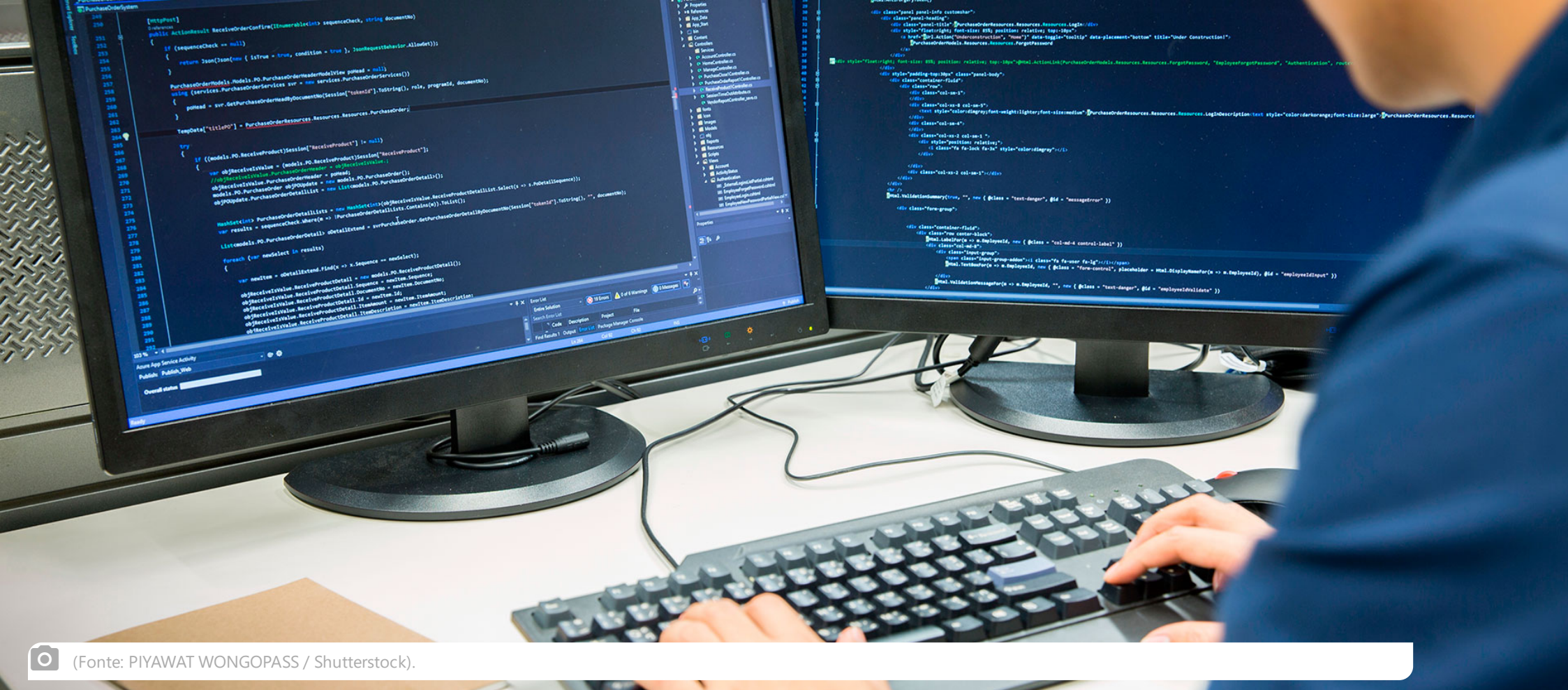
Os testes aconteciam desde a fase inicial do projeto de software até a fase de encerramento e entrega do produto final.

Alguns padrões foram criados para a medição e avaliação do processo de desenvolvimento, e o modelo que ganhou maior credibilidade e importância para as empresas desenvolvedoras foi o [Capability Maturity Model \(CMM\)](#)<sup>1</sup>, apresentado pelo SEI.

Nos anos 1990, surgiram algumas ferramentas de teste que proporcionaram alta produtividade e qualidade no processo.

Assim, determinados tipos de testes, que antes não eram possíveis de serem executados, tornaram-se de fato uma realidade, proporcionando alta produtividade e qualidade no processo de teste e, consequentemente, na qualidade do software.





(Fonte: PIYAWAT WONGOPASS / Shutterstock).

## Cenário atual do desenvolvimento de software

A era digital exigiu que os softwares fossem se adaptando à realidade. Hoje eles fazem parte do nosso cotidiano, estando presentes, por exemplo, nos aplicativos das mais diversas áreas das atividades humanas, como alimentação, transações bancárias, compra e aquisição de produtos, contratação de serviços etc.

A **Globalização** é o processo que proporciona a integração entre diversas sociedades, países e comunidades em todo o mundo, aproximando pessoas, empresas e seus departamentos, clientes, fornecedores etc., seja no âmbito político, cultural, financeiro ou comercial.

O destaque maior da **Globalização** está na integração de mercado existente entre os países.

Assim, a exigência pela qualidade, funcionalidade, portabilidade e tantas outras características fizeram com que esses aplicativos atingissem um grau elevado de complexidade e integridade.

Por outro lado, provavelmente você já teve experiência com algum software ou aplicativo que não funcionasse a contento.

O que pode acontecer se:

Bancos perderem milhões?

Clientes virem saldos de suas contas sumirem de repente?

Telefones pararem de funcionar?

Aviões tiverem suas rotas desviadas?

Vários trens do metrô forem colocados no mesmo trilho?

Atenção

Softwares que apresentam bugs podem acarretar diversos problemas, como supressão de negócio, prejuízos financeiros, além de perda de tempo e, principalmente, queda na reputação das empresas.

Assim sendo, os processos de gerenciamento de testes ganham cada vez mais importância no contexto do desenvolvimento do software, uma vez que vulnerabilidades ou falhas podem gerar riscos e causar perdas, em muitos casos, irreversíveis.

Apesar disso, algumas empresas ainda resistem em investir na área de segurança e testes, por considerar seu custo alto e desnecessário.

Essas empresas precisaram quebrar paradigmas e considerar que a implantação de um processo que garanta a qualidade do software é estratégico e vital para a continuidade dos negócios, em um mercado cada vez mais exigente e competitivo.

Principais demandas de software atuais e a evolução do processo de qualidade e de teste, segundo Bartié (2002)

Características	1960	1980	2000
Tamanho do software	Pequeno	Médio	Muito Grande
Complexidade do Software	Baixa	Média	Alta
Tamanho da Equipe de Desenvolvimento	Pequeno	Médio	Grande
Padrões de Metodologia de Desenvolvimento	Interno	Moderado	Sofisticado
Padrões e Metodologias de Qualidade e Testes	Interno	Emergente	Sofisticado
Organização de Qualidade e Testes	Bem poucas	Algumas	Muitas
Reconhecimento da Importância da Qualidade	Pequeno	Algum	Significante
Tamanho da Equipe de Qualidade e Testes	Pequeno	Pequeno	Grande

Tabela: Evolução das empresas desenvolvedoras de softwares.

Qual a realidade dos softwares atuais?

Você deve ter percebido que toda a sociedade tornou-se dependente de sistemas de software, que hoje são parte integrante do nosso dia a dia.



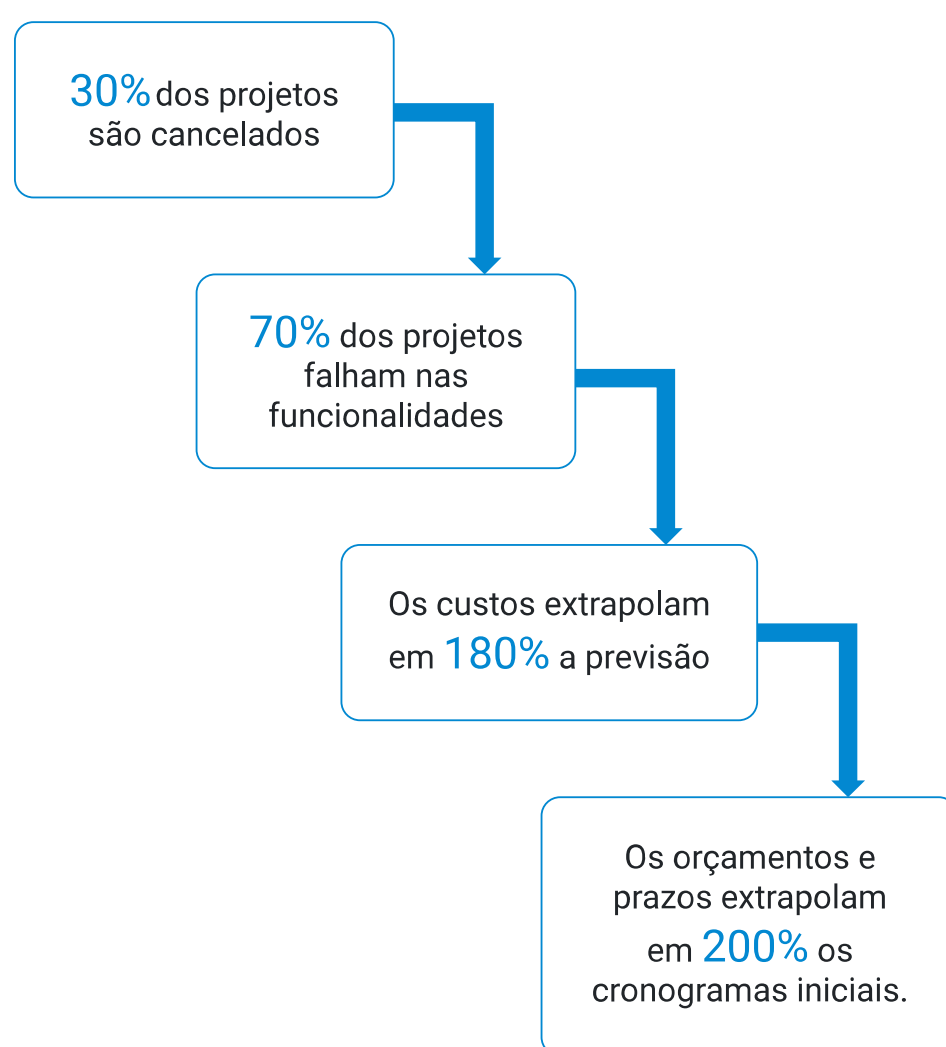


(Fonte: Rawpixel.com / Shutterstock).

As empresas desenvolvedoras estão percebendo que os processos de desenvolvimento são estratégicos e agregam valor aos seus negócios, valorizando seus produtos e serviços.

Na realidade, a indústria de software não está preparada para atender às exigências do mercado em constante evolução porque não investe em seus processos internos.

Um estudo feito recentemente nos EUA mostra o quanto a indústria de softwares está deficitária:



Déficits na indústria de softwares.

**Atenção!** Aqui existe uma videoaula, acesso pelo conteúdo online

## A necessidade de testes no desenvolvimento de softwares

A necessidade e a importância dos testes vão depender do tipo de uso que o software terá. São mais críticos aqueles que podem causar danos à vida humana ou levar a grandes perdas financeiras.

Como vimos com Myers, quanto mais precoce a detecção de falhas ocorre, menores os gastos do projeto com reparos e replanejamento.

**Por esse motivo, a utilização de ferramentas de suporte a testes tem se tornado uma regra no desenvolvimento de softwares.**

## Atenção

A qualidade de um produto ou artefato reúne um conjunto de características e propriedades que devem ser satisfeitas segundo as exigências do usuário, de modo a atender a uma medida de conformidade com as especificações, como defeito zero nos componentes e no produto final, obtendo benefício com o alcance da qualidade.

## Definição sobre qualidade de software

Segundo Pressman (2016), em seu livro *Engenharia de Software*:

"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, 2016.

Esse processo visa garantir a uniformidade dos processos e produtos, eliminando defeitos e melhorando o desempenho de suas funcionalidades.

Para desenvolver softwares de qualidade, é necessário investir em processos de gestão de qualidade, atuando em todas as fases do ciclo de vida.

Alguns fatores internos e externos podem afetar a qualidade do software. Vejamos alguns exemplos:

1

### Fatores externos

São percebidos tanto pelas pessoas que desenvolvem softwares quanto pelos usuários.

2

### Fatores internos

São percebidos apenas pelas pessoas que desenvolvem softwares.

## Exemplo

São exemplos de **Fatores externos**: confiabilidade, eficiência e facilidade de uso.

São exemplos de **Fatores internos**: modularidade<sup>2</sup> e legibilidade<sup>3</sup>.

É importante notar que a garantia de qualidade de software (Software Quality Assurance) não é algo com que começamos a nos preocupar depois que o código foi gerado, e sim “**ao longo de todo o processo de engenharia de software**”.

# Todos os métodos, ferramentas e procedimentos definidos pela Engenharia de Software buscam um único objetivo: produzir softwares de alta qualidade.

Segundo Philip Crosby (*Quality is Free*): “O problema do gerenciamento da qualidade não é o que as pessoas não sabem. **O problema é o que as pessoas acham que sabem**”.

## Garantia de qualidade

Os testes também fazem parte dos procedimentos seguidos para garantir a qualidade do processo de desenvolvimento de softwares, assegurada por certificações concedidas por organizações que avaliam o processo, considerando modelos de qualidade, como o CMMI (Capability Maturity Model Integration) e a ISO-12207.

**Atenção!** Aqui existe uma videoaula, acesso pelo conteúdo online

## Garantia de qualidade de software (SQA)

Um grande desafio para qualquer programa de qualidade, considerado crítico, é possibilitar que qualquer pessoa faça revisões no trabalho de profissionais experientes.

Os gerentes sempre querem os melhores profissionais para projetar o produto, mas geralmente o SQA não pode tê-los. É necessário concentrar esforços em métodos de SQA que permitam um desenvolvimento que possa ser revisado também por pessoas que não são desenvolvedores. Nesse caso, qual o papel do SQA?

Monitorar os métodos e os padrões que os engenheiros de software usam e verificar se eles estão usando apropriadamente seus conhecimentos.

**É importante compreender que as pessoas podem ser experientes em SQA sem, no entanto, serem experientes em projetos de software.**





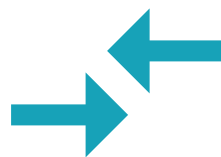
# É impossível obter um software com qualidade com processos de desenvolvimento ineficientes.

Nos processos de gestão de qualidade, o software deverá atender a todas as exigências do cliente/ usuário.

Softwares mal testados causam prejuízos às empresas, como retrabalho, aumento de custo do projeto e informações inconsistentes que podem acarretar decisões equivocadas, além da insatisfação dos usuários.

Temos a aplicação de qualidade em duas dimensões:

**Qualidade do  
Processo**



**Qualidade do  
Produto**


## Qualidade do processo

---

A preocupação com a qualidade deve estar presente em todas as fases do ciclo de desenvolvimento, inclusive no início e na fase de análise de requisitos do sistema.

Quanto mais cedo os problemas forem detectados e corrigidos, mais rápido será o desenvolvimento e com menor custo.

 Como medir?

 Clique no botão acima.

## Como medir?

Métricas de software são usadas para ajudar os desenvolvedores a criar softwares com qualidade reconhecida.

Alguns garantem que o software é incomensurável, porém desconhecem que existem técnicas que colocam as métricas e medições como práticas fundamentais para a determinação do grau de maturidade dos processos de desenvolvimento, conforme definido nas plataformas CMMI e MPS-BR (melhoria do processo de software brasileiro).

O software pode ser medido aplicando-se testes na documentação gerada em cada fase do ciclo de vida. São chamados testes de verificação.

Além disso, existem outras técnicas de medição, como a **Análise de Pontos de Função - APF**, usada para a medição de projetos de software, seguindo alguns parâmetros de medida de tamanho, em **pontos de função - PF**, considerando a funcionalidade implementada, sob o ponto de vista do usuário.

## Atividade

---

1. O teste do software deve acontecer: (assinale a única opção correta)

- a) No início do processo de desenvolvimento
- b) No final do processo de desenvolvimento
- c) No meio do processo de desenvolvimento
- d) No início e no final do processo de desenvolvimento
- e) Em todo o processo de desenvolvimento

2. Com relação ao tamanho e complexidade de software, assinale a opção correta, considerando tais características nos anos 1960, 1980 e 2000.

- a) Em 1980, o tamanho era pequeno e a complexidade alta.
  - b) Em 1960, o tamanho era mínimo e, em 1980, a complexidade era relativa.
  - c) Em 1960, o tamanho era mínimo e, em 2000, a complexidade era média.
  - d) Em 1980, a complexidade era média e, em 2000, a complexidade era alta.
  - e) Em 1960, o tamanho era mínimo e, em 2000, o tamanho é o mesmo.
-

3. O que estabelece a regra de 10 de Myers?

- a) Quanto mais tardiamente descobrimos os erros, mais caros eles ficam.
- b) Os testes tornam-se mais complexos, pois os riscos de os softwares não funcionarem a contento cresce de forma exponencial.
- c) Não existe garantia de que a solução tecnológica contratada será entregue no prazo e nos custos negociados.
- d) A partir de processos uniformes e consistentes, a tendência é que o produto final gerado, ou seja, o software seja eficiente.
- e) Todas as decisões tomadas durante o processo de desenvolvimento do software podem comprometer a sua qualidade final.

4. Podemos conceituar qualidade de software como:

- a) Um processo sistemático que focaliza todas as etapas e artefatos gerados com o objetivo de garantir a conformidade e uniformidade de processos e produtos, prevenindo e eliminado defeitos.
- b) Um processo que foca em todos os produtos de software gerados pela equipe de desenvolvimento.
- c) Um processo que demonstra que algo funciona corretamente.
- d) Um processo para provar que determinadas funções fazem o que devem fazer.
- e) Um processo para demonstrar que os defeitos não estão presentes.

5. Myers concluiu que “zero defeito” é algo inatingível, porém a qualidade de software trabalha com o conceito de zero defeito. O que isto quer dizer?

## Notas

### *Capability Maturity Model*<sup>1</sup>

*Capability Maturity Model* (ou Modelo de Maturidade em Capacitação) para Software é um conjunto de processos desenvolvido pela SEI - Software Engineering Institute ([www.sei.cmu.edu](http://www.sei.cmu.edu)) em 1986 para melhorar o desenvolvimento de aplicações em organizações que trabalham com tecnologias de software.

O processo é dividido em cinco níveis de desenvolvimento: inicial, repetível, definido, gerenciado com métricas e otimizado.

### Modularidade<sup>2</sup>

Conceito no qual o sistema ou software é dividido em partes distintas.

### Legibilidade<sup>3</sup>

Facilidade de leitura em: simplicidade; ortogonalidade; instruções de controle e estruturas e tipos de dados.

## Referências

CROSBY, Philip. ***Quality is free**: the art of making quality certain*. Nova York: McGraw-Hill, 1979.

DELAMARO, M.E.; MALDONADO, J.C.; JINO, M. **Introdução ao teste de software**. 1. ed. Rio de Janeiro: Campus, 2007.

PEZZÉ, Mauro; YOUNG, Michal. **Teste e análise de software**. 1. ed. Porto Alegre: Bookman, 2008.

PRESSMAN, R. S. **Engenharia de software**: uma abordagem profissional. 8. ed. Rio de Janeiro: Campus, 2016.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson, 2011.

## Próxima aula

---

- Teste de software;
- Importância dos testes;
- Papéis e responsabilidades de teste de software.

## Explore mais

---

Leia os textos:

- [Criação e geração de planos de teste de software](#);
- [Capability Maturity Model for Software](#);
- [Qualidade de Software](#);
- [Artigo Engenharia de Software - Introdução à Inspeção de Software](#).

Você poderá aprofundar seus conhecimentos sobre o tema Qualidade de software, acessando os seguintes websites:

- [Associação Latino Americana de Testes de Software \(ALATS\)](#);
- [BSTQB](#);