

# FIAP

Compliance & Quality Assurance

2025

# Normas na Qualidade de Software

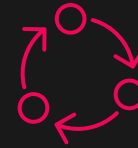
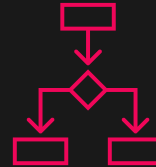
**Prof. Braz**

# Conteúdo

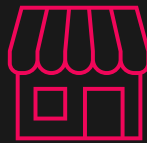
1. O que são Padrões?
2. ISO/IEC 25010 – Modelo de Qualidade de Produto de Software
3. ISO/IEC 29119 – Teste de Software

# O que são Padrões

De acordo com a **ISO**, padrões são “documentações de diretrizes (*guidelines*) que refletem acordos sobre produtos, práticas ou operações



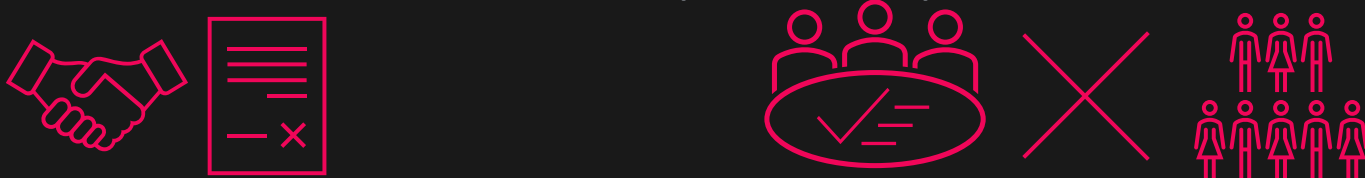
Feitos por associações industriais, profissionais, comerciais ou órgãos governamentais reconhecidos nacional ou internacionalmente”



Eles são “*guidelines*” (documentos de orientação) pois não são obrigatórios, a menos que exigidos por um indivíduo ou uma organização

# O que são Padrões

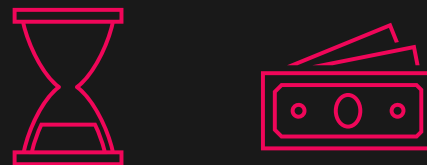
Os padrões são baseados em acordos porque refletem um certo nível de consenso. O consenso não precisa implicar unanimidade



Consumidores instruídos têm maior confiança em organizações que podem mostrar conformidade com os padrões



E essas mesmas organizações se beneficiam ao basear suas práticas nos padrões acordados do setor



# Certificações em Normas ISO

É possível certificar empresa em alguma ISO?

Sim

Toda ISO é certificável?

Não

Nem toda isso é certificável. Por mais que as ISOs sejam documentos de orientação com boas práticas.

# Certificações em Normas ISO

Me da exemplos de ISOs certificaveis?

ISO/IEC 15504 (SPICE)  
ISO 27001

# Órgãos Padronizadores

A criação de padrões é gerenciada por um grande número de órgãos de padronização.

Existem vários órgãos internacionais (ISO, IEC, ITU, CEN) e nacionais (ANSI, ABNT), que são normalmente representantes nos órgãos internacionais.

## **ISO (International Standards Organization):**

compreende uma rede de mais de 160 órgãos nacionais de padronização e publicou mais de 25.000 padrões até 2023.

O Plano Estratégico da ISO para 2021-2030 expressa o desejo de ter “padrões ISO espalhados por todos os lugares, ir de encontro com as necessidades globais e ouvir todas as vozes”.



# Eficiência e Eficácia

**Eficiente** é o que executa uma tarefa com qualidade, competência, excelência, com nenhum ou com o mínimo de erros.

**A eficiência está ligada ao modo de fazer uma tarefa.**

**Eficaz** é o que cumpre perfeitamente determinada tarefa ou função, atingindo o objetivo proposto.

**A eficácia está diretamente ligada ao resultado.**

O eficiente faz com qualidade, mas nem sempre atinge um objetivo.

O eficaz faz o que é certo para atingir o objetivo inicialmente planejado.

<https://www.trf3.jus.br/emag/emagconecta/conexaoemag-lingua-portuguesa/eficaz-eficiente-efetivo>

# Efetividade

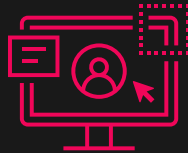
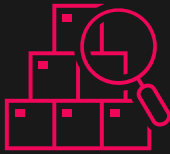
Efetivo é o que tem a habilidade de ser Eficiente e Eficaz ao mesmo tempo

Efetividade consiste em:

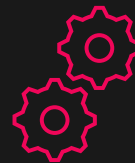
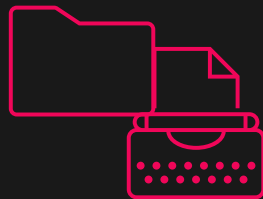
- Fazer o que deve ser feito,
- Com qualidade,
- Utilizando os recursos da melhor maneira possível
- Para atingir o objetivo inicialmente proposto.

# Normas ISO relacionadas a SQA

- **ISO/IEC 25010** - **Modelo de Qualidade de Produto**: define um modelo de qualidade para produtos de software e sistemas relacionados, especificando características e subcaracterísticas que podem ser usadas para avaliar a **qualidade do software**.

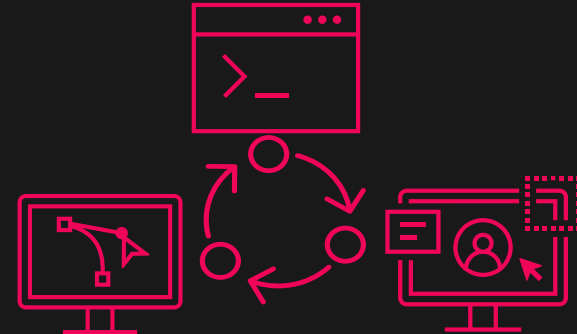


- **ISO/IEC/IEEE 29119** - **Norma de Processos de Teste de Software**: composta por várias partes que abordam diferentes aspectos do processo de teste de software, incluindo planejamento, especificação, execução e avaliação de testes.

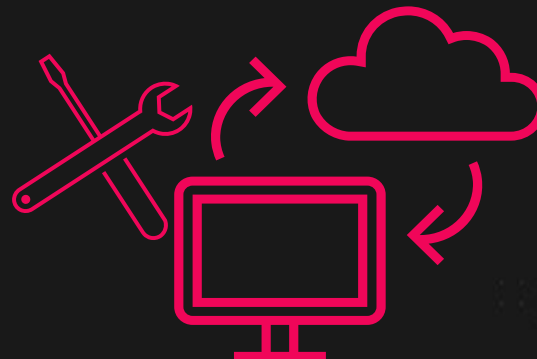
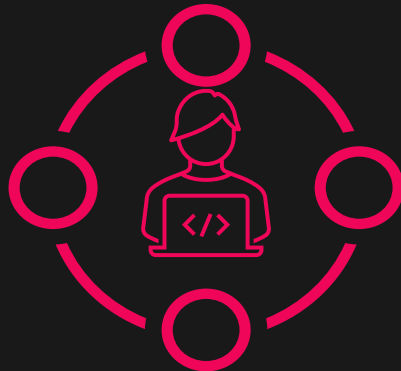


# Normas ISO relacionadas a SQA

**ISO/IEC/IEEE 12207** - **Ciclo de Vida de Software**: aborda o ciclo de vida completo do software, incluindo a fase de teste.



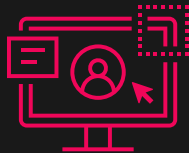
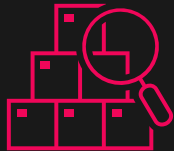
**ISO/IEC 15504** - **Avaliação de Processos**: também conhecida como SPICE (Software Process Improvement and Capability Determination), é focada na avaliação de processos de desenvolvimento e manutenção de software, incluindo atividades de teste.



# ISO/IEC 25010

## Modelo de Qualidade de Produto

O modelo de qualidade é a base de um sistema de **avaliação da qualidade de produto** e determina quais **características de qualidade** serão levadas em consideração ao **avaliar as propriedades de um produto de software**.

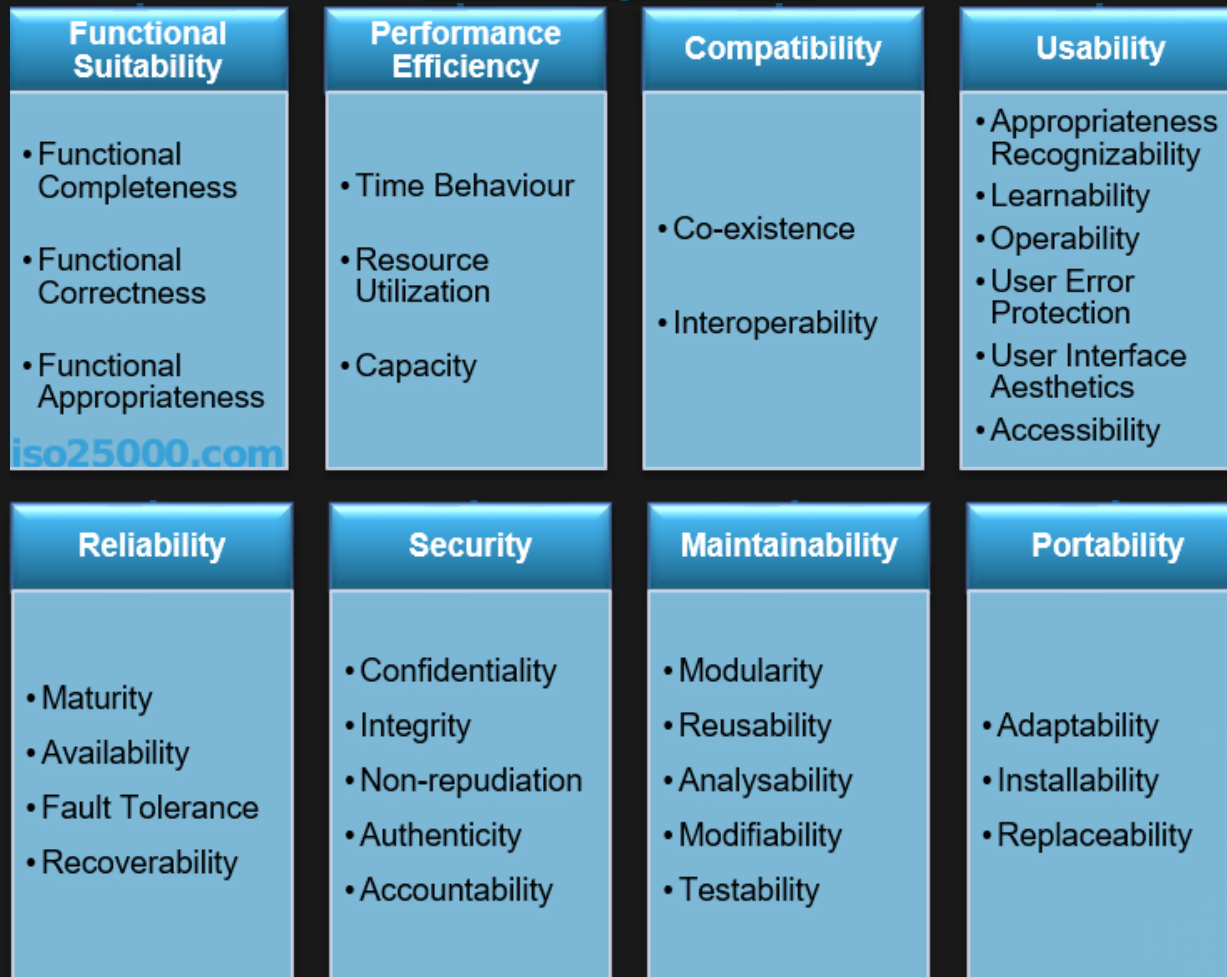


A **ISO/IEC 25010**, também conhecida como **SQuaRE** (*Software Product Quality Requirements and Evaluation*), substituiu a norma ISO/IEC 9126 e define um modelo que compreende oito características de qualidade, cada uma com várias subcaracterísticas, mostradas a seguir:

# ISO/IEC 25010

## Modelo de Qualidade de Produto

### SOFTWARE PRODUCT QUALITY

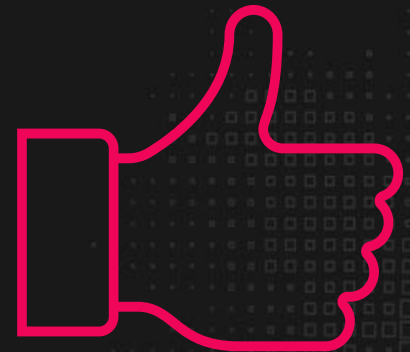
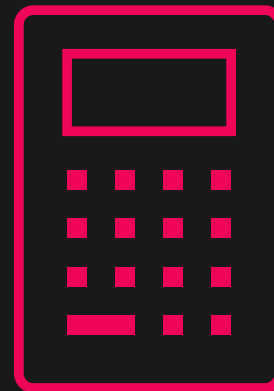
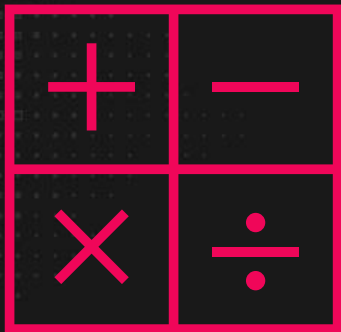


# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 1. Adequação Funcional (*Functional Suitability*)

Esta característica representa o grau em que um produto ou sistema fornece funções que atendem às necessidades explícitas e implícitas, quando usado sob condições especificadas.



### 1. Adequação Funcional (*Functional Suitability*) subcaracterísticas:

- **Compleitude funcional** (*Functional completeness*): Grau em que o conjunto de funções abrange todas as tarefas especificadas e os objetivos do usuário.
- **Correção funcional** (*Functional correctness*): Grau em que um produto ou sistema fornece os resultados corretos com o grau de precisão necessário.
- **Conveniência funcional** (*Functional appropriateness*): Grau em que as funções facilitam a realização de tarefas e objetivos especificados.

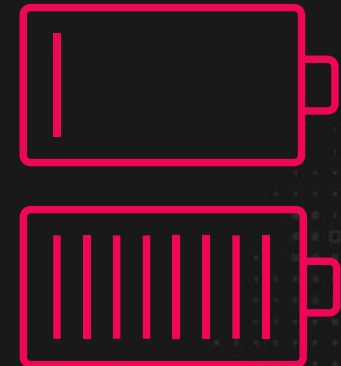
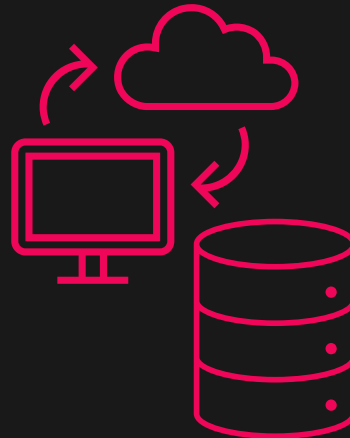


# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 2. Eficiência de desempenho (*Performance efficiency*)

Esta característica representa o desempenho relativo à quantidade de recursos utilizados em determinadas condições.



# ISO/IEC 25010

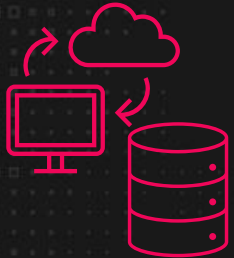
## Modelo de Qualidade de Produto

### 2. Eficiência de desempenho (*Performance efficiency*)

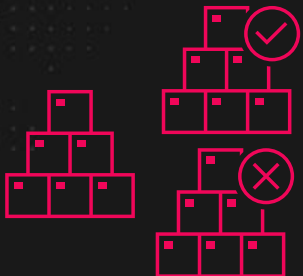
subcaracterísticas:



- **Comportamento do tempo** (*Time behaviour*): Grau em que os tempos de resposta e processamento e as taxas de produtividade de um produto ou sistema, ao executar suas funções, atendem aos requisitos.



- **Utilização de recursos** (*Resource Utilization*): Grau em que as quantidades e tipos de recursos utilizados por um produto ou sistema, ao executar suas funções, atendem aos requisitos.



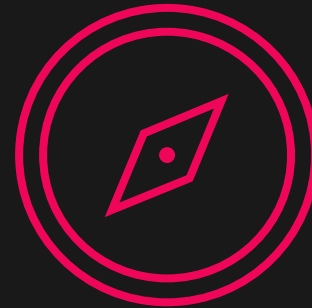
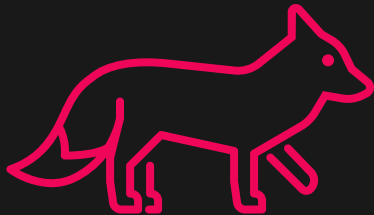
- **Capacidade** (*Capacity*): Grau em que os limites máximos de um parâmetro de produto ou sistema atendem aos requisitos.

# ISO/IEC 25010

## Modelo de Qualidade de Produto

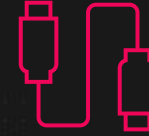
### 3. Compatibilidade (*Compatibility*)

Grau em que um produto, sistema ou componente pode trocar informações com outros produtos, sistemas ou componentes e/ou executar suas funções necessárias enquanto compartilha o mesmo ambiente de hardware ou software.

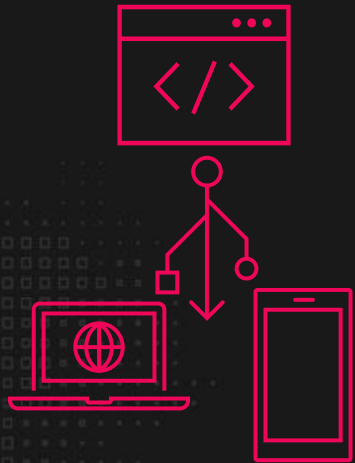


# ISO/IEC 25010

## Modelo de Qualidade de Produto



### 3. Compatibilidade (*Compatibility*) subcaracterísticas:



**Interoperabilidade (*Interoperability*):** Grau em que dois ou mais sistemas, produtos ou componentes podem trocar informações e usar as informações que foram trocadas.



**Coexistência (*Co-existence*):** Grau em que um produto pode executar suas funções necessárias de forma eficiente enquanto compartilha um ambiente e recursos comuns com outros produtos, sem impacto prejudicial sobre qualquer outro produto.

Como nos exemplos a seguir:

# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 3. Compatibilidade – Exemplos de Coexistência:

Em um projeto temos duas APIs instaladas no mesmo servidor, uma desenvolvida em Java e outra desenvolvida em C#.

Apesar de usarem SDKs diferentes e configurações específicas para cada tipo de linguagem, não temos problemas para executar as funções necessárias de cada uma de forma eficiente.

Fazendo analogia com uma instalação elétrica residencial: não é possível ligar dois chuveiros de um mesmo disjuntor. Isso porque quando os dois chuveiros são ligados ao mesmo tempo é a sobrecarga do sistema elétrico gerando a "queda do disjuntor".

Significando então que esses dois itens não coexistem, dado que temos problemas para executar as funções necessárias de cada um de forma eficiente.

Se é possível instalar e mas não é possível usar as funções necessárias de cada um de forma eficiente, então temos um problema de coexistência.

# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 3. Compatibilidade – Exemplos de Coexistência:

No seu aparelho você pode baixar os seguintes aplicativos que usam a localização do device:

Maps;

- Waze;
- Instagram;
- Tinder;
- Happn;
- Banco;
- App de corrida
- Et al.

Ou seja,

Apesar de compartilhar um mesmo recurso em uma mesmo device, todos funcionam adequadamente.

# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 4. Usabilidade (*Usability*)

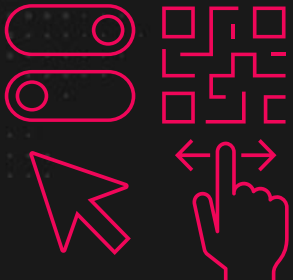
Grau em que um produto ou sistema pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso especificado. Subcaracterísticas:

- **Reconhecimento de adequação** (*Appropriateness recognizability*): Grau em que os usuários podem reconhecer se um produto ou sistema é apropriado para suas necessidades.
- **Operabilidade** (*Operability*): Grau em que um produto ou sistema possui atributos que o tornam fácil de operar e controlar.

Canva

X

Word



# ISO/IEC 25010

## Modelo de Qualidade de Produto

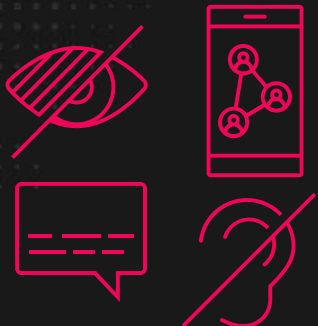
### 4. Usabilidade (*Usability*) subcaracterísticas:



- **Proteção contra erros do usuário (*User error protection*):** Grau em que um sistema protege os usuários contra erros.



- **Estética da interface do usuário (*User interface aesthetics*):** Grau em que a interface do usuário permite uma interação agradável e satisfatória.



- **Acessibilidade (*Accessibility*):** Grau em que um produto ou sistema pode ser usado por pessoas com a mais ampla gama de características e capacidades.



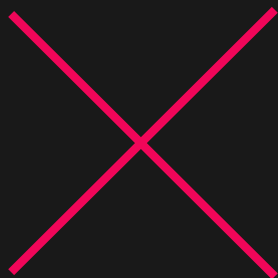
# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 5. Confiabilidade (*Reliability*)

Grau em que um sistema, produto ou componente executa funções especificadas, sob condições especificadas, por um período de tempo especificado. Subcaracterísticas:

- **Maturidade (*Matiruty*)**: Grau em que um sistema, produto ou componente atende às necessidades de confiabilidade em operação normal.



# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 5. Confiabilidade (*Reliability*) subcaracterísticas:



- **Disponibilidade (*Availability*)**: Grau em que um sistema, produto ou componente está operacional e acessível para uso quando necessário.



- **Tolerância a falhas (*Fault Tolerance*)**: Grau em que um sistema, produto ou componente opera conforme pretendido, apesar da presença de falhas de hardware ou software.



- **Recuperabilidade (*Recoverability*)**: Grau em que, em caso de interrupção ou falha, um produto ou sistema pode recuperar os dados diretamente afetados e restabelecer o estado desejado.

# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 6. Segurança (*Security*)

Grau em que um produto ou sistema protege informações e dados para que pessoas ou outros sistemas tenham o grau de acesso apropriado para seus tipos e níveis de autorização. Subcaracterísticas:



- **Confidencialidade (*Confidentiality*)**: Grau em que um produto ou sistema garante que os dados sejam acessíveis apenas àqueles autorizados a ter acesso.



- **Integridade (*Integrity*)**: Grau em que um sistema, produto ou componente impede o acesso não autorizado ou a modificação de programas ou dados de computador.

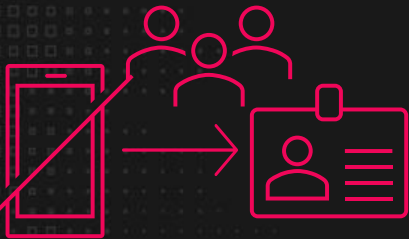
# ISO/IEC 25010

## Modelo de Qualidade de Produto

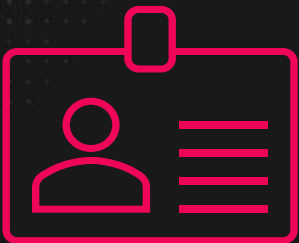
### 6. Segurança (*Security*) subcaracterísticas:



- **Não-repúdio (*Non-repudiation*):** Grau em que ações ou eventos podem ter seu acontecimento comprovado para que não possam ser repudiados posteriormente.



- **Responsabilidade (*Accountability*):** Grau em que as ações de uma entidade podem ser atribuídas exclusivamente à essa entidade.



- **Autenticidade (*Authenticity*):** Grau em que a identidade de um sujeito ou recurso pode ser provada como aquela reivindicada.

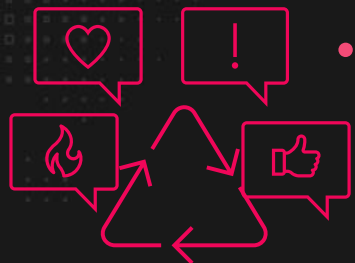
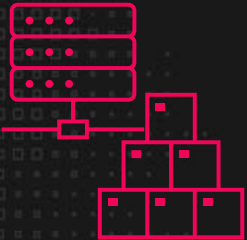
# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 7. Manutenibilidade (*Maintainability*)

Grau de eficácia e eficiência com que um produto ou sistema pode ser modificado para melhorá-lo, corrigi-lo ou adaptá-lo à mudanças de ambiente ou requisitos. Subcaracterísticas:

- **Modularidade (*Modularity*)**: Grau em que um sistema ou programa de computador é composto de componentes discretos, de modo que uma alteração em um componente tenha impacto mínimo em outros componentes.
- **Reutilização (*Reusability*)**: Grau em que um ativo pode ser usado em mais de um sistema ou na construção de outros ativos.

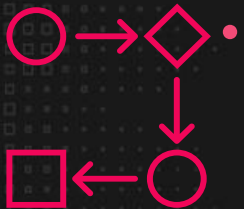


# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 7. Manutenibilidade (*Maintainability*) subcaracterísticas:

- **Analísabilidade (*Analysability*)**: Grau de eficácia e eficiência com o qual é possível avaliar o impacto de uma alteração pretendida em uma ou mais partes de um produto ou sistema, ou diagnosticar um produto quanto a deficiências ou causas de falhas, ou identificar partes serem modificadas.
- **Modificabilidade (*Modifiability*)**: Grau em que um produto ou sistema pode ser eficaz e eficientemente modificado sem introduzir defeitos ou degradar a qualidade do produto existente.
- **Testabilidade (*Testability*)**: Grau de eficácia e eficiência com o qual os critérios de teste podem ser estabelecidos para um sistema, produto ou componente e com o qual os testes podem ser executados para determinar se esses critérios foram atendidos.

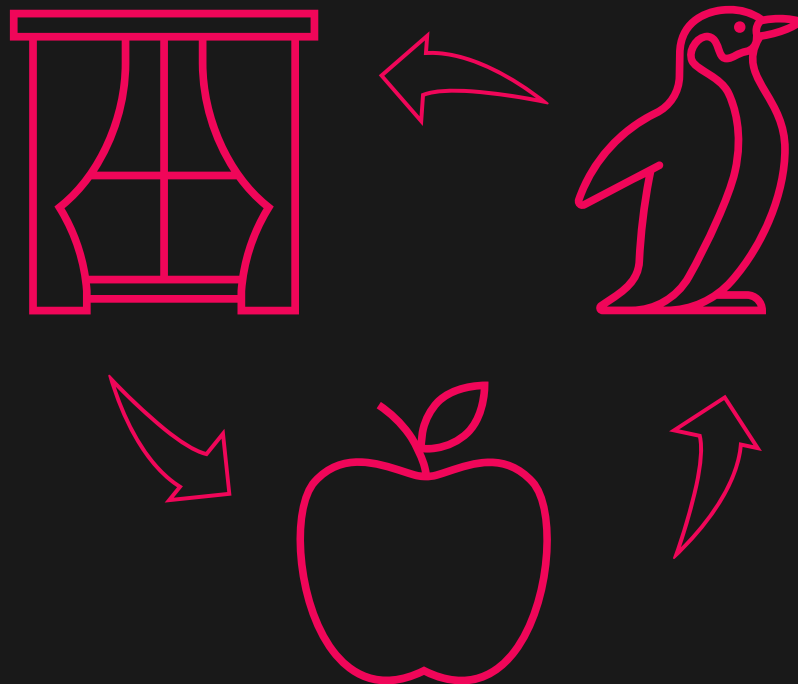


# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 8. Portabilidade (*Portability*)

Grau de eficácia e eficiência com que um sistema, produto ou componente pode ser transferido de um hardware, software ou ambiente operacional para outro.





# ISO/IEC 25010

## Modelo de Qualidade de Produto

### 8. Portabilidade (*Portability*) subcaracterísticas:



- **Adaptabilidade (*Adaptability*)**: Grau em que um produto ou sistema pode ser eficaz e eficientemente adaptado para hardware, software ou outros ambientes operacionais ou de uso diferentes ou em evolução.



- **Instalabilidade (*Installability*)**: Grau de eficácia e eficiência com o qual um produto ou sistema pode ser instalado e/ou desinstalado com sucesso em um ambiente especificado.



- **Substituibilidade (*Replaceability*)**: Grau em que um produto pode substituir outro produto de software especificado para o mesmo propósito no mesmo ambiente.



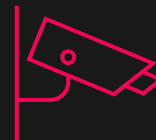
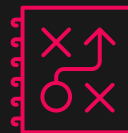
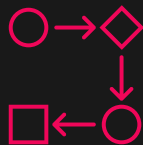
# ISO/IEC 29119

## Teste de Software

A série de normas **ISO/IEC/IEEE 29119** é dedicada aos **processos de teste de software** e fornece diretrizes detalhadas para todas as etapas do ciclo de vida do teste.

A série é composta por várias partes, definindo:

- Vocabulário,
- Processos,
- Documentação,
- Técnicas
- E um modelo de avaliação do processo de teste de software.



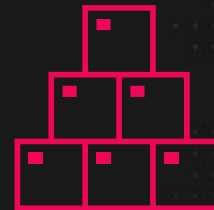
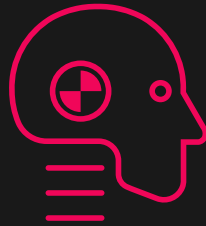
# ISO/IEC 29119

## Teste de Software

Foi projetada para ser flexível e adaptável a diferentes contextos e tipos de projetos de software.

A norma foi baseada em outros padrões já existentes, como

- IEEE 829 (Documentação para Testes de Software),
- IEEE 1008 (Testes Unitários),
- BS 7925-1 (Vocabulário)
- E BS 7925-2 (Componentes de Software).



# ISO/IEC 29119

## Teste de Software

A série é composta por cinco padrões:

- **ISO/IEC 29119-1**: Conceitos & Definições (2013)
- **ISO/IEC 29119-2**: Processos de Teste (2013)
- **ISO/IEC 29119-3**: Documentação de Teste (2013)
- **ISO/IEC 29119-4**: Técnicas de Teste (2015)
- **ISO/IEC 29119-5**: Testes Baseados em Palavras-Chave (2016)

# ISO/IEC 29119

## Teste de Software

### 1. Conceitos & Definições

A **ISO/IEC/IEEE 29119 Parte 1** facilita o uso das outras partes do padrão, introduzindo o vocabulário no qual a série é construída e fornecendo exemplos de sua aplicação na prática.

A Parte 1 fornece definições, uma descrição dos conceitos de teste de software e formas de aplicar essas definições e conceitos às outras partes do padrão.

# ISO/IEC 29119

## Teste de Software

### 2. Processos de Teste

A **Parte 2** define um modelo de processo genérico para teste de software. Ele compreende descrições que definem os processos de teste de software

no nível organizacional, de gerenciamento de teste (projeto) e de processo de testes dinâmicos  
(não houve consenso sobre a inclusão de testes estáticos).

Os processos definidos neste padrão podem ser usados em conjunto com diferentes modelos de ciclo de vida de desenvolvimento de software.

# ISO/IEC 29119

## Teste de Software

### 3. Documentação de Teste

A **Parte 3** trata da documentação de teste de software e inclui modelos e exemplos que são produzidos durante o todo processo de teste.

Os modelos suportam principalmente os três níveis primários de processo de teste da Parte 2, mas também inclui o mapeamento para outros padrões existentes.

Os documentos definidos na ISO/IEC/IEEE 29119-3 são listados a seguir:

# ISO/IEC 29119

## Teste de Software

### Documentação do Processo de Teste Organizacional

- Política de teste
- Estratégia de teste organizacional

### Documentação do processo de gerenciamento de teste

- Plano de Teste (incluindo uma Estratégia de Teste)
- Status do teste
- Conclusão do teste

### Documentação do processo de teste dinâmico

- Especificação do Projeto de Teste
- 
- Especificação do Caso de Teste
- 
- Especificação do procedimento de teste
- 
- Requisitos de dados de teste
- 
- Relatório de Prontidão de Dados de Teste



### Documentação do processo de teste dinâmico

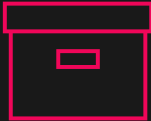
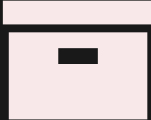
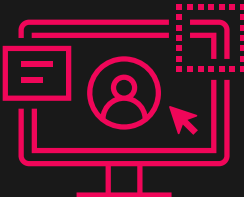
- Requisitos do Ambiente de Teste
- Relatório de Prontidão do Ambiente de Teste
- Resultados reais- Resultado do teste
- Registro de Execução de Teste
- Relatório de Incidente de Teste

# ISO/IEC 29119

## Teste de Software

### 4. Técnicas de Teste

A **Parte 4** fornece definições de técnicas de design de teste de software (ou métodos de teste) e respectivas medidas de cobertura, que podem ser usadas durante os processos de design e implementação de teste definidos na Parte 2. As técnicas de aqui mencionadas são categorizadas em três categorias principais:

- Baseadas em **especificação** (caixa-preta) 
- Baseadas em **estrutura** (caixa-branca) 
- Baseadas em **experiência** 

# ISO/IEC 29119

## Teste de Software

### Técnicas de design de teste baseadas em especificações

Essas técnicas são baseadas na especificação funcional do sistema em teste, também são chamadas de técnicas de **caixa preta**.

- **Partição de equivalência:** modelo que segmenta as entradas e saídas do teste em grupos semelhantes (partições equivalentes), onde cada grupo irá gerar uma condição de teste.
- **Cobertura de Instrução / Árvore de classificação:** modelo que classifica os diferentes tipos de entrada e os representa em um gráfico de árvore, onde cada classe (e sub-classe) de entrada não se sobrepõe a outra. Cada classe gera uma condição de teste.

# ISO/IEC 29119

## Teste de Software

### Técnicas de design de teste baseadas em especificações

- **Análise de valor limite:** modelo que particiona entradas e saídas em conjuntos ordenados com limites identificáveis, sendo cada limite (antes, exato e depois) uma condição de teste.
- **Tabela de decisão:** envolve a criação de tabelas que mostram combinações de entradas e as ações que devem ser tomadas pelo software em resposta a essas combinações
- **Grafos de causa e efeito:** gráfico direcionado que mapeia um conjunto de causas para um conjunto de efeitos. As causas são a entrada do programa e os efeitos são a saída. Pode facilitar a criação de uma tabela de decisão.

# ISO/IEC 29119

## Teste de Software

### Técnicas de design de teste baseadas em especificações

- **Teste combinatório:** abordagem usada quando há diversas combinações de parâmetros e valores de entrada a serem considerados para teste, criando subgrupos representativos ao invés de testar todas as combinações possíveis.
- **Transição de estado:** técnica que considera cada estado que uma variável ou componente pode apresentar, as transições entre estados, eventos que causam as transições e as ações resultantes delas.

# ISO/IEC 29119

## Teste de Software

### Técnicas de design de teste baseadas em especificações

- **Transição de estado:** técnica que considera cada estado que uma variável ou componente pode apresentar, as transições entre estados, eventos que causam as transições e as ações resultantes delas.
- **Teste de cenário:** modelo que estipula sequências de ações e interações representando um fluxo de uso do item de teste (cenário). Deve ser definido um cenário principal e então considerados cenários alternativos desses fluxos de uso.
- **Teste aleatório:** técnica que considera o uso de valores aleatórios de entrada, que devem ser escolhidos considerando todo o domínio possível de valores do objeto de teste. Modelos de distribuição de valores como normal, uniforme e perfil de operação podem ser usados.

# ISO/IEC 29119

## Teste de Software

### Técnicas de design de teste baseadas em estrutura

Essas técnicas são baseadas na estrutura interna do sistema, ou seja, o seu código-fonte. Também são chamadas de técnicas de **caixa-branca**.

- **Cobertura de instruções:** técnica que testa as instruções executáveis do código. A cobertura é uma porcentagem do número de instruções executadas pelos testes em relação número total de instruções executáveis existentes.
- **Cobertura de decisão:** técnica que testa as decisões existentes no código e o código executado com base nos resultados da decisão. Decisões são pontos no código onde o escolhe um de dois ou mais resultados possíveis (IF, ou SWITCH/CASE).
- **Cobertura de fluxo de dados:** abordagem onde casos de teste são definidos de acordo com o caminho de um par “definição-uso” de uma determinada variável. “Definição” é a atribuição de valor a uma variável, enquanto seu “uso” pode ser predicado (loops, decisões) ou computacional (cálculos, resultados).



# ISO/IEC 29119

## Teste de Software

### Técnicas de design de teste baseadas em experiência

Essas técnicas são baseadas na habilidade e intuição do testador e de sua experiência com aplicativos e tecnologias semelhantes.

- **Suposição de erro:** técnica usada para prever a ocorrência de erros, defeitos e falhas, com base no conhecimento do testador, considerando funcionamento passado, tendência de erros cometidos, falhas ocorridas em outros sistemas.
- **Teste exploratório:** técnica em que testes informais (não pré-definidos) são modelados, executados, registrados e avaliados dinamicamente durante a execução da tarefa. O resultado, além de encontrar novas falhas, pode ser usado para aprender mais sobre o sistema ou escrever outros testes formais.



### 5. Testes Baseados em Palavras-Chave

Este padrão apresenta uma abordagem para especificar testes de software normalmente automatizados.

A ideia é fornecer um conjunto de blocos de construção (as palavras-chave) que podem ser usados para criar casos de teste sem a necessidade de conhecimento profundo de programação ou da própria ferramenta de teste.

# ISO/IEC 29119

## Teste de Software

### 5. Testes Baseados em Palavras-Chave

test cases	StartAPP CreateFile InputContents <b>saveFile</b> Exit	InitializeCamera CreatePreview <b>takePicture</b> VerifyPicture Exit
domain layer keywords	<b>saveFile:</b> getContents <b>SelectMenu</b> selectSave	<b>takePicture:</b> initialize <b>InvokeAPI</b> CameraSnapshot finalize
test interface layer keywords (script code)	<b>SelectMenu()</b> { hWnd= GetWindowHandler() postMsg(hWnd, MenuMsg); }	<b>InvokeAPI(API)</b> { setupParameter() Res= Call(API) check(Res) }
test interface	GUI	API

## Referências:

ISTQB Syllabus v3.1.1: [https://bcr.bstqb.org.br/docs/syllabus\\_ctfl\\_3.1.1br.pdf](https://bcr.bstqb.org.br/docs/syllabus_ctfl_3.1.1br.pdf)

<https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

<https://wildart.github.io/MISG5020/>

<https://malenezi.github.io/malenezi/SE401/Books/114-the-art-of-software-testing-3-edition.pdf>