

# RESUMO AULA 3 – VALIDAÇÃO DE REQUISITOS DE SOFTWARE

# Objetivo da aula:

1. Reconhecer a importância da verificação de requisitos de software.
2. Aplicar checklists de verificação de requisitos funcionais e não funcionais de software.
3. Criar casos de teste para a verificação de requisitos funcionais e não funcionais de software.



# Introdução

Todos sabemos que o mercado tem se tornado cada vez mais exigente com relação à qualidade dos produtos de software. Isso se dá em parte porque existe uma grande quantidade de softwares disponíveis para resolver diferentes tipos de problemas do cotidiano.



# Introdução

Essa intensa exposição à tecnologia na vida pessoal torna os usuários mais exigentes inclusive com os softwares que ele utiliza no trabalho. Nesse ambiente, entretanto, não é tão simples substituir um produto por outro.

E como garantir que os produtos de software tenham a qualidade esperada e proporcionem uma experiência extraordinária ao usuário?



# Introdução

E o que podemos fazer para reduzir os efeitos nocivos desses erros?

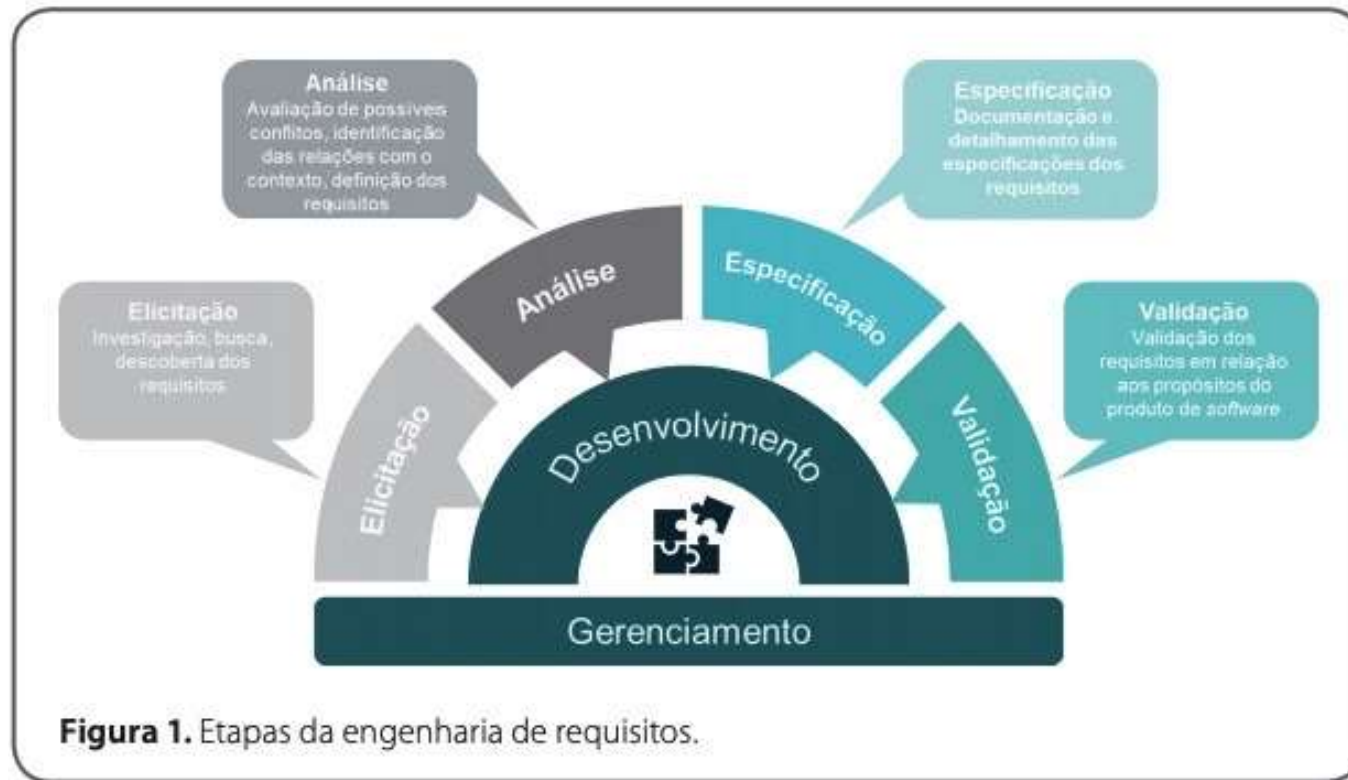
Utilizar técnicas de verificação e validação ao longo das atividades de todo o ciclo de vida.

E isto começa pela etapa de requisitos.



# Importância da verificação de requisitos de software

Vamos relembrar essas etapas referentes à engenharia de requisitos por meio da Figura 1.



# Importância da verificação de requisitos de software

**Verificação e validação** são termos diferentes! De acordo com a norma internacional ISO/IEC/IEEE 12207 (ISO/IEC/IEEE, 2017, p. 10–11):

- **Validação é:** “a confirmação, por meio do fornecimento de evidência objetiva, que o requisito foi atendido para um uso ou aplicação pretendidos específicos”.  
“demonstra que a solução vai atender seu uso pretendido no ambiente alvo, isto é, ‘você está construindo a coisa certa’”.
- **Verificação é:** a “confirmação, por meio do fornecimento de evidência objetiva, que os requisitos especificados foram atendidos”.  
“endereço que o produto de trabalho ou a solução refletem adequadamente os requisitos especificados, isto é, ‘você está construindo certo’”.



# Importância da verificação de requisitos de software

É comum dizermos que a verificação analisa se o produto foi construído certo, ou seja, de forma correta; e que a validação analisa se foi construído o produto certo, ou seja, aquele que os stakeholders desejavam (CMMI INSTITUTE, 2018).





# Importância da verificação de requisitos de software

À medida que os softwares foram se tornando mais complexos, mais integrados e mais críticos para todas as atividades cotidianas, mais importante ficou a garantia do seu correto funcionamento.

A grande questão é que há muitos pontos no ciclo de desenvolvimento nos quais erros que podem levar a falhas gigantescas de software.

Isso ocorre porque as atividades envolvidas são de ordem cognitiva. A cada troca de mãos do requisito, ou seja, a cada troca de responsabilidade, um novo defeito pode estar sendo introduzido, como naquela brincadeira infantil do telefone sem fio.



# Importância da verificação de requisitos de software

- **Primeira transição** = Stakeholders ↔ Analista de Requisitos
- **Segunda transição** = Analista de Requisitos ↔ Equipe Técnica
- **Terceira transição** = Equipe Técnica ↔ Testadores
- **Ultima transição** = Testadores ↔ Implantação em produção



# Importância da verificação de requisitos de software

Todas essas diversas trocas de responsabilidade podem levar a erros que se refletirão no ambiente de operação e podem trazer consequências diversas, como o retrabalho e seus custos associados ou falhas na operação, que podem gerar riscos de ferimentos ou de prejuízos financeiros.



# Importância da verificação de requisitos de software

Muitas dessas falhas poderiam ter sido evitadas se procedimentos adequados de qualidade tivessem sido adotados ao longo do ciclo de vida, conforme ilustra a Figura 2, por meio das atividades de Verificação 1 a 5, representadas pelos triângulos. Note que está representado apenas um ciclo de vida genérico e bastante simplificado, que pode ser uma sprint, por exemplo.



# Importância da verificação de requisitos de software

Diversas técnicas podem ser aplicadas para evitar que os erros se propaguem ao longo das fases, funcionando como uma espécie de filtro, como vimos na Figura 2.

Quanto maior a quantidade de filtros e quanto mais finos eles forem, menos erros de propagação. O problema é que esses filtros têm um custo, o chamado custo da qualidade, que deve ser cuidadosamente analisado para determinar o ponto de equilíbrio ideal.



# Como realizar a verificação de requisitos

Os requisitos podem ser verificados de diversas formas:

1. Leitura crítica dos artefatos feito pelo analista de requisitos (Cheklist);
2. Revisão por pares (outros analistas de requisitos ou pessoas técnicas da equipe):

São consideradas revisões por pares:

- inspeções;
- walkthroughs (passo a passo) estruturados;
- refatoração deliberada;
- Programação em pares (pair programming).

Da mesma forma que a autoavaliação, a avaliação por pares pode ou não ser apoiada por um checklist.



# Artefatos de requisitos

**Artefato:** é o resultado da realização de uma atividade, podendo ser um documento, um diagrama ou o próprio código.

Revisões por pares podem ser aplicadas a qualquer artefato do projeto.



# Artefatos de requisitos

Vamos considerar neste capítulo duas formas de especificar requisitos: a abordagem orientada a casos de uso e a abordagem orientada a histórias de usuário.

Além disso, vamos considerar também a lista de requisitos, que pode ser utilizada nas duas abordagens. Dessa forma, os seguintes artefatos serão considerados neste capítulo:

- Lista de requisitos funcionais e não funcionais;
- Casos de uso: ■ Diagrama de casos de uso. ■ Especificação de casos de uso.
- Histórias de usuário: ■ Cartão da história de usuário. ■ Critérios de aceitação das histórias de usuário.





# Checklist para revisão de artefatos de requisitos

Um checklist pode ser considerado como uma lista contendo pontos de atenção para apoiar os revisores.

Geralmente, ela é constituída dos itens que anteriormente já foram considerados problemáticos e que a equipe deseja evitar.

O ideal é que cada artefato tenha seu checklist com itens próprios.



# Checklist para revisão de artefatos de requisitos

Um checklist pode ser considerado como uma lista contendo pontos de atenção para apoiar os revisores.

Geralmente, ela é constituída dos itens que anteriormente já foram considerados problemáticos e que a equipe deseja evitar.

O ideal é que cada artefato tenha seu checklist com itens próprios.

**Quadro 1.** Checklist para a lista de requisitos

Atributo	Definição	Sim	Não	Não se aplica
Completo	O requisito está especificado de forma completa e que possibilita que o desenvolvedor o implemente?			
Correto	O requisito reflete o que o usuário, cliente ou seus representantes desejam?			
Único	O requisito descreve uma única capacidade, característica, restrição ou atributo de qualidade?			



# Checklist para revisão de artefatos de requisitos

**Quadro 1.** Checklist para a lista de requisitos

Atributo	Definição	Sim	Não	Não se aplica
Viável	O requisito é viável técnica e financeiramente de ser implementado, de acordo com as restrições do projeto?			
Necessário	O requisito tem um motivo de existir, que é representado pelo seu relacionamento a uma fonte de informação e a um objetivo de negócio?			
Priorizado	O requisito tem uma prioridade atribuída para que possa ser alocado a uma versão do <i>software</i> ?			
Não ambíguo	O requisito não contém ambiguidades que levem os <i>stakeholders</i> a interpretá-lo de forma diferente?			
Verificável	O requisito é possível de ser verificado posteriormente quanto à sua implementação?			
Conforme	O requisito está em conformidade com os padrões de especificação estabelecidos, se houver?			

**Fonte:** Adaptado de Wiegers e Beatty (2013) e ISO/IEC/IEEE (2018).



# Checklist para revisão de artefatos de requisitos

O **diagrama de casos de uso** é um diagrama que ajuda o analista de requisitos a se comunicar com os usuários.

Embora seja composto por elementos simples, o diagrama tem um grande poder de comunicação.

O Quadro 2 no próximo slide, apresenta uma sugestão de checklist para verificação do diagrama de casos de uso.



# Checklist para revisão de artefatos de requisitos

**Quadro 2.** Checklist para o diagrama de casos de uso

Atributo	Definição	Sim	Não	Não se aplica
Atores (escopo)	Todos os atores estão representados no diagrama?			
Atores (formato)	Todos os atores estão representados por um boneco palito e um texto com a identificação do ator?			
Relacionamentos de generalização entre atores (representação)	O relacionamento de generalização entre os atores é representado por uma seta vazada que aponta do ator filho para o ator pai?			
Relacionamentos de generalização entre atores (significado)	O relacionamento de generalização expressa corretamente a intenção de herança entre os atores envolvidos?			
Casos de uso (representação)	Todos os casos de uso são representados por elipses e identificados por meio de um verbo no infinitivo seguido de um complemento?			
Casos de uso (escopo)	Todos os casos de uso que representam as funcionalidades que os <i>stakeholders</i> desejam estão representados?			
Relacionamentos dos casos de uso com atores	Todos os casos de uso estão ligados a pelo menos um ator?			

(Continua)

# Checklist para revisão de artefatos de requisitos

(Continuação)

**Quadro 2.** Checklist para o diagrama de casos de uso

Atributo	Definição	Sim	Não	Não se aplica
Relacionamentos de <include> entre casos de uso	Os relacionamentos de <include> estão representados por setas pontilhadas que partem do caso de uso principal para o caso de uso incluído?			
Relacionamentos de <extend> entre casos de uso	Os relacionamentos de <extend> estão representados por setas pontilhadas que partem do caso de uso que estende para o caso de uso principal?			
Relacionamentos de generalização entre casos de uso (representação)	O relacionamento de generalização entre os casos de uso é representado por uma seta vazada que aponta do caso de uso filho para o caso de uso pai?			
Relacionamentos de generalização entre casos de uso (significado)	O relacionamento de generalização expressa corretamente a intenção de herança entre os casos de uso envolvidos?			

# Checklist para revisão de artefatos de requisitos

(Continuação)

**Quadro 2.** Checklist para o diagrama de casos de uso

Atributo	Definição	Sim	Não	Não se aplica
Relacionamentos de <include> entre casos de uso	Os relacionamentos de <include> estão representados por setas pontilhadas que partem do caso de uso principal para o caso de uso incluído?			
Relacionamentos de <extend> entre casos de uso	Os relacionamentos de <extend> estão representados por setas pontilhadas que partem do caso de uso que estende para o caso de uso principal?			
Relacionamentos de generalização entre casos de uso (representação)	O relacionamento de generalização entre os casos de uso é representado por uma seta vazada que aponta do caso de uso filho para o caso de uso pai?			
Relacionamentos de generalização entre casos de uso (significado)	O relacionamento de generalização expressa corretamente a intenção de herança entre os casos de uso envolvidos?			

# Checklist para revisão de artefatos de requisitos

Quando a equipe de desenvolvimento utiliza métodos ágeis, é comum que os requisitos funcionais sejam expressos como histórias de usuário.

As histórias de usuário são compostas por cartões, com uma declaração da necessidade do usuário e por critérios de aceitação que normalmente detalham a história ou especificam os requisitos não funcionais.

O responsável pelos itens que compõem o product backlog, e que irão gerar as histórias, é o product owner (PO). Cabe a ele representar os interesses do cliente ou do usuário junto à equipe de desenvolvimento.

As histórias de usuário geralmente são escritas de forma conjunta, entre o PO e a equipe de desenvolvimento.





# Checklist para revisão de artefatos de requisitos

O Quadro 4 apresenta uma sugestão para o checklist de histórias de usuário. Note que esses critérios são muito similares aos critérios da lista de requisitos.

É sugerido que o revisor complemente as informações dos itens assinalados como “não”, de modo a facilitar as correções a serem realizadas pelo autor do artefato.



# Checklist para revisão de artefatos de requisitos

**Quadro 4.** Checklist para histórias de usuário

Atributo	Definição	Sim	Não	Não se aplica
Formato	A história de usuário está escrita no padrão: "Como (nome do papel), eu quero (...) de modo que (...)?"			
Completa	A história de usuário está especificada de forma completa e que possibilita que o desenvolvedor o implemente?			
Correta	A história de usuário reflete o que o usuário, cliente ou seus representantes desejam?			
Única	A história de usuário descreve uma única capacidade, característica, restrição ou atributo de qualidade?			

(Continua)

# Melhoria da qualidade das revisões

As revisões por pares são momentos preciosos para promover a melhoria na qualidade das especificações de requisitos.

Para que as revisões sejam proveitosas e se ganhe em produtividade, Wiegers (2006) apresenta as dicas a seguir.

- **Eduque os revisores;**
- **Não sobrecarregue os revisores;**
- **Construa parcerias colaborativas com os representantes dos usuários e com outros membros da equipe do projeto;**
- **Convide os revisores certos;**
- **Faça os revisores analisarem os entregáveis apropriados;**
- **Projete para facilitar a revisão;**
- **Inspecione todos os entregáveis de requisitos;**
- **Enfatize encontrar os maiores erros.**



# Casos de teste de requisitos de software

A técnica mais utilizada para realizar a verificação de artefatos de desenvolvimento de software é o teste.

Geralmente, os casos de teste são escritos como apoio para que a equipe de testes possa realizar o seu trabalho.



# Casos de teste de requisitos de software

Para a especificação dos casos de teste, o template apresentado no Quadro 5 pode ajudar:

Se tudo der certo e o sistema prover a resposta esperada, ele indica isto marcando um X na coluna ✓. Caso a execução não tenha ocorrido com sucesso, então ele marcará um X na coluna que aparece como X na planilha.



# Casos de teste de requisitos de software

Quadro 5. Template para casos de teste

ID do caso de uso	ID do caso de teste	Passos	Resultado esperado	✓	×	N/A
<inserir o id do caso de uso>	<inserir o id do caso de teste>	<inserir os passos detalhados para a realização do caso de teste>	<inserir o resultado esperado ao final da execução do passo>			
EXEMPLO						
UC-CAD-001	CT-CAD-001	1 – Selecionar função Cadastrar Cliente	Tela de Cadastro de Cliente aberta	×		
		2 – Preencher CPF incorreto	Msg de erro exibida: "CPF inválido"		×	
		(...)				

# FIM

