

# Requisitos de *Software*

Prof.<sup>a</sup> Ma. Jessica Oliveira

**Aula 06 – 14/04/2025**

# **Validação e Verificação de Requisitos.**

# Introdução.

- Durante a fase de levantamento e documentação dos requisitos de um sistema de software, é comum que erros aconteçam: requisitos podem ser mal interpretados, mal formulados, contraditórios ou mesmo incompletos.
- Por isso, duas etapas fundamentais da Engenharia de Requisitos são a verificação e a validação.
- Esses processos ajudam a garantir que os requisitos estejam corretos, completos, claros e alinhados às necessidades reais dos usuários e clientes.
- Essas duas atividades, embora muitas vezes mencionadas juntas, têm propósitos diferentes e complementares.
- Compreender essa distinção e saber aplicar técnicas de verificação e validação é essencial para a construção de sistemas de qualidade.

# Diferença entre Verificação e Validação.

# Verificação.

- Serve para responder à pergunta: “Estamos construindo o produto corretamente?”.
- É o processo de **analisar a documentação dos requisitos** com o objetivo de garantir que ela esteja **tecnicamente correta, bem estruturada, completa, coerente e conforme os padrões** definidos.
- Ou seja, trata-se de um processo interno, técnico e sistemático, que busca encontrar falhas antes mesmo da construção do sistema.

# Verificação.

- Objetivos principais da verificação:
  - Identificar requisitos incompletos, inconsistentes ou ambíguos.
  - Verificar a conformidade com padrões de documentação.
  - Garantir que os requisitos estejam escritos de forma clara e precisa.
  - Conferir se as dependências entre os requisitos foram respeitadas.
- **Exemplo prático:** se um requisito afirma que "o sistema deve ser rápido", um verificador identificará que essa definição é vaga. O que é “rápido”? Em termos mensuráveis? Para quem? Isso deve ser especificado para evitar diferentes interpretações.

# Validação.

- Ajuda a responder o questionamento: “Estamos construindo o produto certo?”.
- Tem como foco assegurar que o sistema atenda às necessidades e expectativas dos *stakeholders*, ou seja, que os requisitos representem corretamente aquilo que o cliente ou usuário final espera.
- Diferente da verificação, a validação **envolve a participação dos próprios *stakeholders*** (usuários, clientes, gestores), que irão avaliar se o que está sendo definido como requisito de fato corresponde ao que desejam.
- Essa atividade busca responder: o que está documentado realmente atende às necessidades do usuário?

# Validação.

- Objetivos principais da validação:
  - Garantir que os requisitos representam corretamente as necessidades do negócio.
  - Confirmar que os requisitos são compreensíveis e aceitáveis para o cliente.
  - Evitar o desenvolvimento de funcionalidades indesejadas ou equivocadas.
- **Exemplo prático:** ao apresentar um protótipo de tela para o cliente, ele percebe que o campo de busca que havia solicitado não está presente. Isso é um indício de que o requisito não foi validado corretamente com ele.



# Importância da Verificação e Validação.

- Erros cometidos na fase de levantamento de requisitos podem ser extremamente caros se forem detectados apenas nas fases finais do desenvolvimento.
- Estudos mostram que **quanto mais tarde um erro é detectado, maior o seu custo de correção**. Por isso, a verificação e a validação são consideradas atividades preventivas, e não corretivas.
- Sem essas práticas, os riscos incluem:
  - Requisitos conflitantes ou mal compreendidos.
  - Desenvolvimento de funcionalidades desnecessárias.
  - Retrabalho na codificação.
  - Baixa aceitação do sistema pelos usuários.
  - Insatisfação dos stakeholders.

- Além disso, V&V são fundamentais para alcançar os atributos de qualidade do software, como:
  - **Confiabilidade** (requisitos claros e corretos levam a sistemas mais estáveis);
  - **Usabilidade** (validação garante que a interface e a lógica sejam compreensíveis);
  - **Manutenibilidade** (requisitos bem escritos são mais fáceis de manter e evoluir).

# Métodos e Técnicas de Verificação e Validação de Requisitos.

# *Walkthrough.*

- É uma revisão informal de um documento de requisitos.
- Um membro da equipe conduz uma apresentação dos requisitos enquanto os demais participantes (geralmente colegas de equipe, analistas ou desenvolvedores) acompanham e apontam possíveis dúvidas ou problemas.
- Pode envolver leitura em conjunto do documento.
- O foco está na compreensão coletiva e na identificação de falhas.
- É útil para disseminar conhecimento e identificar erros iniciais.

# Inspeção.

- É uma técnica mais formal do que o *walkthrough*.
- Envolve uma análise detalhada do documento por um grupo de revisores, que seguem um roteiro previamente definido.
- O objetivo é encontrar erros, inconsistências e violações de padrão.
- É uma técnica sistemática e documentada.
- Geralmente conta com papéis definidos: moderador, leitor, escritor de atas.
- É comum em ambientes que seguem normas ISO ou CMMI.

# Revisões Técnicas.

- São reuniões formais e estruturadas onde documentos técnicos (como os requisitos) são revisados por especialistas da equipe.
- Buscam verificar a qualidade técnica do conteúdo, mas também promover a consistência entre documentos (como requisitos, modelo de dados e protótipos).

# Prototipação.

- A prototipação é uma técnica de validação muito eficaz, pois consiste em criar representações visuais ou funcionais de partes do sistema, como telas ou fluxos de navegação.
- Os protótipos ajudam os usuários a entender melhor o que está sendo proposto.
- Pode ser feita com ferramentas como Figma, Canva, Balsamiq ou até papel.
- Permite validação antecipada da interface e da usabilidade.
- Facilita a comunicação entre o time técnico e o cliente.



# Testes de Requisitos.

- Consiste em criar casos de teste diretamente a partir dos requisitos, com o objetivo de verificar se eles foram compreendidos corretamente e se podem ser testados (ou seja, se são verificáveis).
- Um bom requisito deve ser testável.
- Exemplo: “O sistema deve permitir que usuários cadastrem até 3 dependentes por CPF válido.”
  - Caso de teste: tentar cadastrar 4 dependentes → erro esperado.

# Ambiguidades e Inconsistências em Requisitos.

- Um dos maiores desafios da Engenharia de Requisitos é garantir que os requisitos sejam escritos de forma **clara, objetiva e sem interpretações múltiplas**. A ambiguidade ocorre quando uma frase pode ser entendida de mais de uma maneira. Já a inconsistência acontece quando dois ou mais requisitos se contradizem.
- Exemplos de Ambiguidade:
  - “O sistema deve ser rápido.” → o que é “rápido”? 1s? 5s? Para qual tipo de operação?
  - “O sistema deve gerar relatórios mensais.” → todos os meses? Quais dados?
- Exemplos de Inconsistência:
  - Requisito 1: “Somente administradores podem excluir registros.”
  - Requisito 2: “Todos os usuários podem excluir seus próprios registros.”
- Ambos os casos precisam ser detectados e resolvidos nas fases iniciais, usando técnicas de V&V, para evitar impactos negativos no projeto.

# Participação dos *Stakeholders* no Processo de Validação.

- *Stakeholders* são todas as pessoas ou grupos interessados no produto final (como clientes, usuários, gestores, desenvolvedores e até órgãos reguladores).
- A validação só é efetiva quando essas partes interessadas participam ativamente do processo.
- Formas de envolver stakeholders:
  - Entrevistas ou reuniões de revisão.
  - Apresentação de protótipos navegáveis.
  - Simulações de uso com cenários reais.
  - Aplicação de questionários de validação.
- Além de validar os requisitos, esse contato fortalece o alinhamento de expectativas e contribui para a construção de confiança entre as partes.

# Vamos para um desafio?

# Dúvidas?

[jessica.oliveira@p.ucb.br](mailto:jessica.oliveira@p.ucb.br)