



# **Verificação e Validação (V&V) de Software**

## **Engenharia de Software II**

**Profa. Andréa Sabedra Bordin**

# Introdução

- O desenvolvimento de software está sujeito a diversos tipos de **problemas**, os quais acabam resultando na obtenção de um **produto diferente daquele que se esperava**.
- Muitos fatores podem ser identificados como causas de tais problemas, mas a maioria deles tem uma única origem: **erro humano** (DELAMARO, MALDONADO e JINO, 2007).

# Processos de Verificação e Validação

- Objetivo:
  - Assegurar que o software:
    - a) **cumpra as suas especificações.**
    - b) **atenda às necessidades do usuário/cliente.**
- Quando acontece:
  - **Ao longo do processo de desenvolvimento.**
- Como acontece:
  - Revisões dos **requisitos**
  - Revisões de **projeto**
  - Inspeções de **código**
  - **Testes** do produto

Garantir a **qualidade** do produto de software

# Verificação e Validação

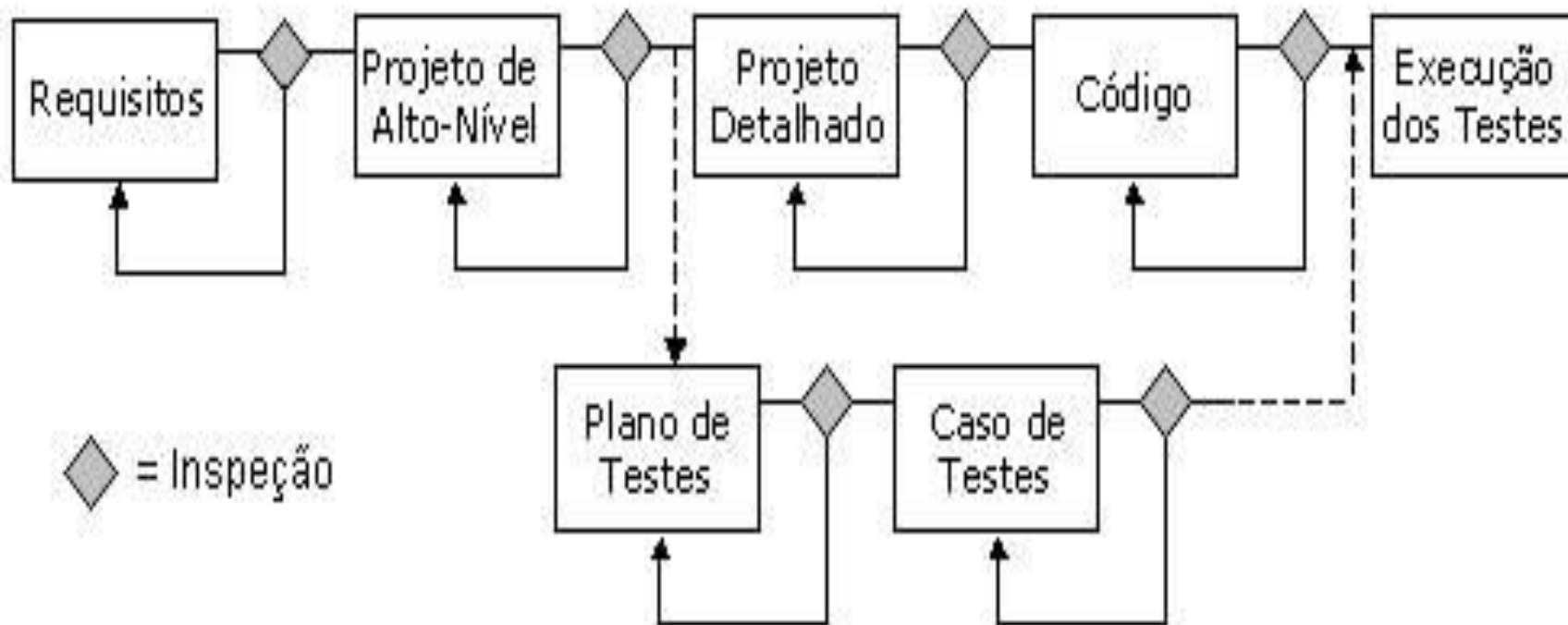
- **Verificação**

- Assegurar consistência, completude e corretude em cada fase do ciclo de desenvolvimento e entre as fases.
- Verifica se o software está de acordo com suas especificações. *Supõe falhas na especificação.*
- “Estamos construindo o produto corretamente?”

- **Validação**

- Assegurar que o sistema deve atender às expectativas do cliente/usuário. Se realiza o que o cliente espera que ele faça.
- “Estamos construindo o produto correto?”

# Verificação e Validação



# Diferenças chave

- O processo de **verificação** inclui a verificação de documentos, *design*, código e programa, enquanto o processo de **validação** inclui testes e validação do produto real.
- A **verificação** não envolve a execução do código, enquanto a **validação** envolve a execução do código.
- A **verificação** usa métodos como revisões, *walkthroughs*, inspeções, enquanto a **validação** usa métodos como teste de caixa preta, teste de caixa branca e teste não funcional.
- A **verificação** verifica se o software confirma uma especificação, enquanto a **validação** verifica se o software atende aos requisitos e expectativas.
- A **verificação** encontra os bugs no início do ciclo de desenvolvimento, enquanto a **validação** encontra os bugs que a verificação não consegue detectar.
- A **verificação** é feita pela equipe de qualidade, enquanto a **validação** é feita pela equipe de teste com a equipe de qualidade.

# Verificação e Validação

A distinção entre os dois termos está amplamente relacionada ao papel das especificações.

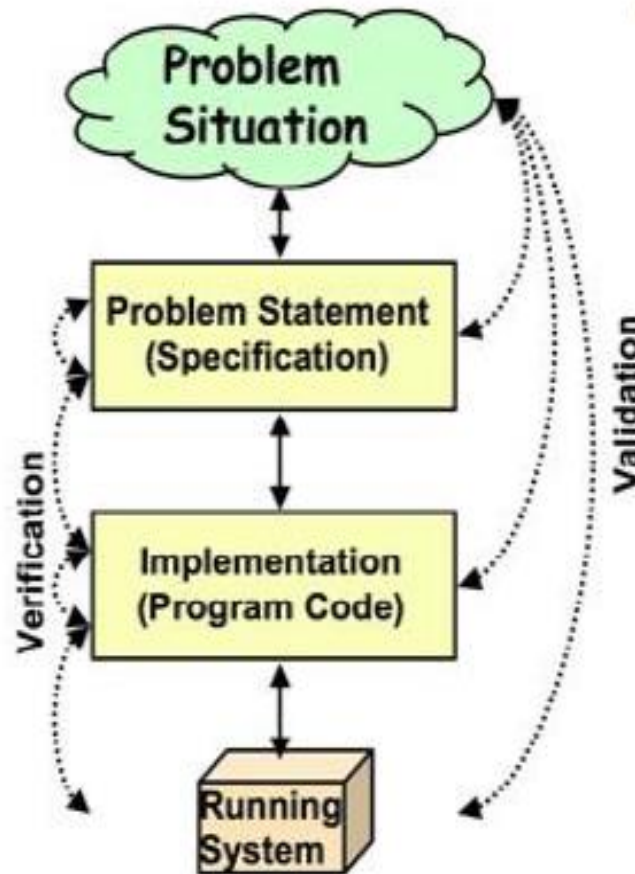
- A **verificação** é o processo de averiguar se o software atende à especificação.
  - Verifica se documentos, projeto, código foram construídos de acordo com os requisitos ou não.
  - As revisões podem ser formais ou informais.
- A **validação** é o processo de averiguar se a especificação captura as necessidades do cliente.
  - Envolve atividades como teste de unidade, teste de integração, teste de sistema e teste de aceitação do usuário.

**Verificação é o processo de verificar se o software atende à especificação.**

- Verifica se o software é bem construído e sem erros.
- Verificação ajuda a determinar se o software é de **alta qualidade** mas não assegura que o sistema é útil.
- É um processo relativamente **objetivo**, na medida em que, se os vários produtos e documentos são expressos com precisão suficiente, não devem ser necessários julgamentos subjetivos, a fim de verificar o software .

**Validação é o processo de verificar se a especificação capta as necessidades do cliente.**

- Validação é um processo **subjetivo**. Trata-se de fazer avaliações subjetivas de quão bem o sistema proposto aborda uma necessidade no mundo real.





# Técnicas de V&V

## **Estática: Inspeção de software**

- Verificação de requisitos, diagramas, código-fonte (correspondência com a especificação).
- Qualquer representação legível de software pode ser inspecionada.
- Não se verifica propriedades como desempenho e confiabilidade.

## **•Dinâmica: Teste de software**

- Execução da implementação do software com dados de teste.
- Comportamento operacional/desempenho e confiabilidade são observados .

# Referências

- Sommerville, Ian. Engenharia de Software - 8ª Edição 2007.
- Pressman, R.S., Engenharia de Software. 6a edição, McGrawHill, 2006.
- Delamaro, M.E., Maldonado, J.C., Jino, M., Introdução ao Teste de Software, Série Campus – SBC, Editora Campus, 2007.
- Kosciński, A., Soares, M.S., Qualidade de Software, Editora Novatec, 2006.
- Myers, G.J., The Art of Software Testing, 2nd edition, John Wiley & Sons, 2004.
- McGregor, J.D., Sykes, D.A., A Practical Guide to Testing Object-Oriented Software, Addison-Wesley, 2001.
- <http://istqbexamcertification.com/what-is-validation-in-software-testing-or-what-is-software-validation/>
- Brazilian Software Test Qualification Board. <http://www.bstqb.org.br/>
- **Introdução à Inspeção de Software. <http://www-di.inf.puc-rio.br/~kalinowski/publications/KalinowskiS07.pdf>**

# Referências

IEEE STANDARDS ASSOCIATION



**IEEE Standard for System, Software,  
and Hardware Verification and  
Validation**

IEEE Computer Society

Sponsored by the  
Software and Systems Engineering Standards Committee

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8055462>