

Processos de software

Prof. Dr. William Simão de Deus

william.deus@ifpr.edu.br
Instituto Federal do Paraná (IFPR)
Campus Pinhais
Gestão da Tecnologia da Informação – 2025.01

O que é software?

- Definir o que é software, com exatidão, é um desafio grade!
- (SOMMERVILLE, 2011), no entanto, sumariza isso da seguinte forma: *software são **programas de computador e documentação associada***

- **Intangíveis:** o software de um computador, o código de um aplicativo, ou os dados armazenados na nuvem são intangíveis.
 - Você pode tocar o celular ou o disco rígido (o hardware), mas não o software em si

- **Intangíveis:** o software de um computador, o código de um aplicativo, ou os dados armazenados na nuvem são intangíveis.
 - Você pode tocar o celular ou o disco rígido (o hardware), mas não o software em si
- **Dados:** muitos software utilizam dados (ou produzem) algum tipo de dado

- **Intangíveis:** o software de um computador, o código de um aplicativo, ou os dados armazenados na nuvem são intangíveis.
 - Você pode tocar o celular ou o disco rígido (o hardware), mas não o software em si
- **Dados:** muitos software utilizam dados (ou produzem) algum tipo de dado
- **Hardware:** os softwares interagem com o hardware

Segundo (SOMMERVILLE, 2011), existem softwares que representam:

- **Produtos genéricos:** produzidos por uma organização de desenvolvimento e disponíveis no mercado
 - Exemplos

Segundo (SOMMERVILLE, 2011), existem softwares que representam:

- **Produtos genéricos:** produzidos por uma organização de desenvolvimento e disponíveis no mercado
 - Exemplos
- **Produtos sob encomenda:** produzidos para um cliente em particular
 - Exemplos

É difícil classificar ou organizar as categorias de softwares:

- Sistemas operacionais (ex.: Windows, Linux, macOS)
- Aplicativos (ex.: navegadores, editores de texto, planilhas)
- Softwares embarcados (ex.: em carros, eletrodomésticos)
- Jogos
- Programas utilitários (ex.: antivírus, compactadores)
- ...

Apesar das diferenças, quase todos os softwares seguem um processo genérico de criação

Um processo de software geralmente é composto por 4 etapas:

- Especificação
- Desenvolvimento (projeto e implementação)
- Validação
- Evolução

- A especificação de software é o processo de definir e documentar os requisitos que o sistema deve atender.
- Inclui a identificação das funcionalidades desejadas, as restrições do sistema, e os critérios de aceitação do produto final.
- Envolve interação com os *stakeholders* para garantir que o software a ser desenvolvido reflita corretamente as necessidades dos usuários e clientes.
- A especificação pode ser apresentada em diferentes formatos, como documentos escritos ou modelos gráficos.

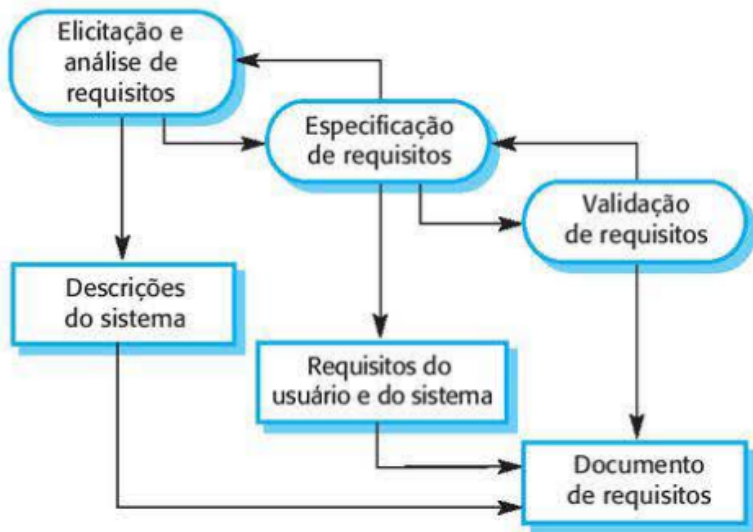


Figura: Fonte: (SOMMERVILLE, 2011)

- Elicitação e análise: observações, conversas, reuniões...
- Especificação: traduzir as informações em um documento
- Validação: realismo, consistência e integridade

- O desenvolvimento de software abrange as atividades de projeto, codificação e construção do sistema.
- A fase de desenvolvimento transforma as especificações em um sistema operacional utilizando linguagens de programação e ferramentas adequadas.
- Envolve a implementação de funcionalidades conforme definido na especificação, além de garantir que o código seja eficiente, escalável e manutenível.
- Pode seguir metodologias variadas.

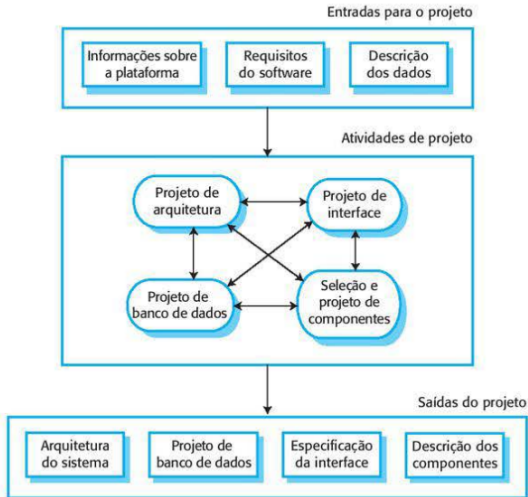


Figura: Fonte: (SOMMERVILLE, 2011)

- Projeto de arquitetura: identificação da estrutura do sistema e seus componentes principais
- Projeto de banco de dados: estruturas de dados e suas representações
- Projeto de interface: criação das interfaces
- Seleção e projeto de componentes: criar ou reusar componentes

- O retrabalho é inevitável

- O retrabalho é inevitável
- Não é apenas sair programando, mas organizar todas as entradas e saídas do projeto

- O retrabalho é inevitável
- Não é apenas sair programando, mas organizar todas as entradas e saídas do projeto
- Devem ser feitas buscas por componentes reusáveis

- O retrabalho é inevitável
- Não é apenas sair programando, mas organizar todas as entradas e saídas do projeto
- Devem ser feitas buscas por componentes reusáveis
- Caso não haja componentes adequados, são projetados novos componentes de software

- A validação de software é o processo que assegura que o produto final atende aos requisitos e expectativas dos *stakeholders*.
- Envolve a realização de testes, revisões e inspeções para garantir que o software funcione corretamente e sem defeitos.
- A validação responde à pergunta "Estamos construindo o produto certo?", verificando se o software resolve o problema esperado pelo cliente.
- Inclui diferentes níveis de teste, como testes de unidade, integração, sistema e aceitação.

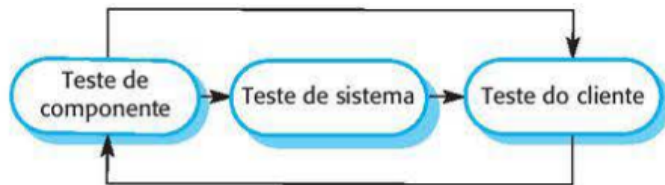


Figura: Fonte: (SOMMERVILLE, 2011)

Teste de componentes:

- Componentes podem ser entidades simples (funções, classes) ou agrupamentos coerentes dessas entidades

Teste de componentes:

- Componentes podem ser entidades simples (funções, classes) ou agrupamentos coerentes dessas entidades

Teste de sistema:

- Verifica se o sistema satisfaz requisitos funcionais e não funcionais
- Testa as propriedades emergentes do sistema após a integração dos seus componentes

Teste do cliente:

- O sistema é testado pelo cliente (ou cliente potencial) em vez de usar dados de simulação
- Pode revelar erros e omissões na definição dos requisitos do sistema, pois os dados reais exercitam o sistema de maneiras diferentes das que ocorrem com os dados de teste
- Também pode revelar problemas de requisitos nos quais os recursos do sistema não satisfazem realmente as necessidades dos usuários ou o desempenho do sistema é inaceitável

- A evolução de software refere-se às mudanças e melhorias feitas no sistema após sua entrega inicial.
- Essas mudanças podem ser motivadas por novos requisitos, correção de defeitos ou adaptações tecnológicas.
- A evolução é uma parte contínua do ciclo de vida do software, garantindo que ele continue útil e relevante ao longo do tempo.
- Inclui atividades como manutenção corretiva, adaptativa e preventiva.

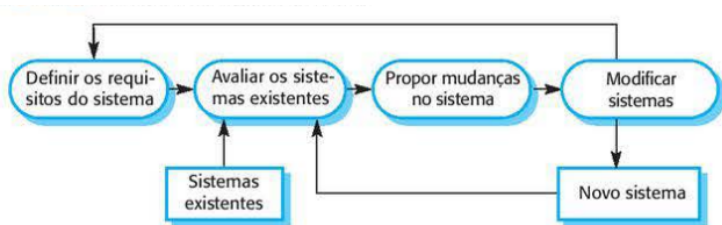


Figura: Fonte: (SOMMERVILLE, 2011)

Muitas organizações (e projetos) organizam o desenvolvimento e a evolução do software como sendo coisas distintas:

- Desenvolvimento: criação, criatividade, construção
- Evolução: sinônimo de manutenção, correção

Desassociar essas duas atividades pode representar um desafio enorme: o software é produzido, mas não é evoluído!

 SOMMERVILLE, I. *Engenharia de Software - 9 ed.* [S.l.]: Pearson, 2011.