



Prof. Ms. Claudio Souza Nunes

# Requisitos de Software

Engenharia de  
Software I

Bacharelado em  
Sistemas de Informação

# O que é um "requisito"?

- Uma condição ou capacidade necessária para que o usuário resolva um problema ou atinja um objetivo.
- Uma condição ou capacidade que deve ser atendida por uma solução para satisfazer um contrato, especificação, padrão ou quais outros documentos formais impostos.



# Importância dos Requisitos

Erros na definição dos requisitos (escopo) do projeto estão entre as causas mais comuns para o fracasso de projetos, especialmente no caso dos projetos de software.



# Classificação dos Requisitos - Simples

## Funcional

Determinam os comportamentos que o sistema deve exibir. Os requisitos funcionais normalmente ou são atendidos ou não são atendidos. De maneira genérica podemos dizer que os requisitos funcionais expressam “o que” o sistema dever fazer.

## Não funcional

Os requisitos que não se enquadram como funcionais mas também estão relacionados ao sistemas são considerados “não funcionais”. Normalmente estão relacionados com comportamentos mais subjetivos: qualidade, confiabilidade, usabilidade, etc.. Podemos dizer que requisitos não funcionais expressam “como” o sistema deve se comportar.



# Classificação dos Requisitos - Detalhada

Segundo o glossário de terminologia de engenharia de software definido na norma do IEEE 610.12-1990 há seis tipos de requisitos:

- Requisitos de “design”
- Requisitos funcionais
- Requisitos de implementação
- Requisitos de interface
- Requisitos de desempenho
- Requisitos físicos



# Fontes de requisitos

- Entrevistas individuais
- Entrevistas aceleradas em grupos (JAD ou similar)
- Estudo dos documentos
- Análise de estatísticas
- Técnicas de Amostragem
- Observação *in loco* dos processos
- Utilização de questionários
- Exame dos registros existentes
- *Design Thinking*



# Atributos dos Requisitos

Correto

Testável

Completo

Consistente

Aceitável

Não  
ambíguo

Rastreável

Priorizável

Modificável



# Requisitos durante o processo de software

Requisitos representam diferentes coisas para diferentes participantes do processo de desenvolvimento:

- **Usuários finais**  
Definem os requisitos e homologam o correto entendimento. Também iniciam os processos de mudança de requisito.
- **Gerentes de Desenvolvimento**  
Usam os requisitos para estimar e planejar as tarefas de desenvolvimento
- **Desenvolvedores (implementação)**  
Usam os requisitos para entender o que deve ser realizado
- **Equipe de teste**  
Usam os requisitos para criar os testes de validação do sistema
- **Desenvolvedores (manutenção)**  
Usam os requisitos para identificar as relações entre as partes que compõe o sistema





# Como especificar requisitos?

- Linguagem natural
- Linguagens “estruturadas”
- Diagramas (exemplo: casos de uso)
- Especificação formal (matemática)

# Problemas com linguagem natural

- Ambiguidade
- Excesso de flexibilidade
- Falta de estrutura e modularidade
- Impossíveis de implementar
- Confusos ou misturados

*"O caipira vai a uma consulta e o médico pergunta:  
- O que senhor tem?  
O caipira responde:  
- Uma muié, uma vaca e uma galinha...  
- Não é isso... O que o senhor está sentindo?  
- Ah, tá! Vontade de largá a muié, vendê a vaca e comê a galinha com quiabo!"*



# Alguns erros comuns na definição de escopo

- Usar como base documentação não oficial
- Usar como base documentação desatualizada
- Não se certificar de quem é o ponto focal dos requisitos
- Não atentar para questões de natureza jurídica
- Não cruzar requisitos entre as diversas fontes de informação
- Acreditar que alguns requisitos estão *implícitos*
- Não especificar requisitos inversos (*fora do escopo*)
- Não atentar para performance do sistema em produção
- Não considerar todas as plataformas de interface
- Não documentar adequadamente
- Não implementar gestão de mudanças



# Requisitos - Roteiro

- Identificar objetos (entidades, elementos) que fazem parte do ambiente do sistema
- Identificar objetos e resultados que devem ser produzidos pelo sistema
- Identificar objetos que serão usados pelo sistema para realizar suas funções
- Identificar os serviços (processos e/ou funções) que manipulam ou interagem com os objetos
- Identificar restrições (custo, complexidade, regras de negócio)
- Identificar limitações de performance (tempo de processamento, etc.)

