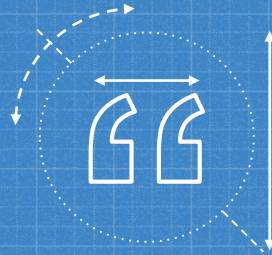




ENGENHARIA DE SOFTWARE I

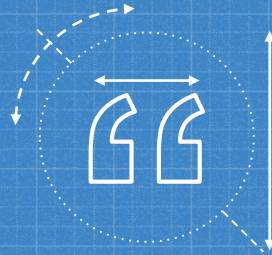
Engenharia de Requisitos

REQUISITOS



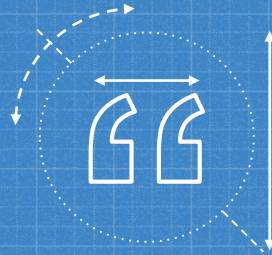
Condição ou capacidade que um sistema (ou componente) deve alcançar ou possuir para satisfazer um contrato, padrão, especificação ou outro documento formalmente imposto.

STAKEHOLDERS



Partes interessadas e/ou envolvidas em um projeto, que devem ser convocadas ou no mínimo consideradas para o levantamento de requisitos.

STAKEHOLDERS



Partes interessadas e/ou envolvidas em um projeto, que devem ser convocadas ou no mínimo consideradas para o levantamento de requisitos.

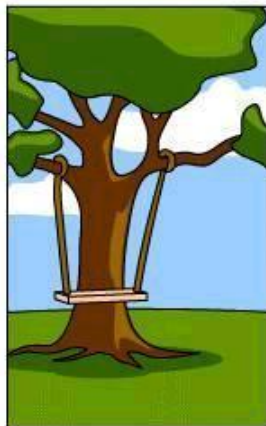
“Interessadas” no sentido mais amplo do conceito.

LEVANTAMENTO DE REQUISITOS

- **Primeira etapa** do ciclo de vida de software.
- **Principal objetivo:** Fazer com que todos os stakeholders tenham a mesma visão do problema a ser resolvido.
- **Objetivos secundários:**
 - Verificar a viabilidade de implementação dessas necessidades.
 - Levantar informações relevantes para que, na etapa de projeto, seja possível estabelecer cronogramas realistas e alocar funções adequadamente aos elementos participantes.



Como o cliente explicou...



Como o líder de projeto entendeu...



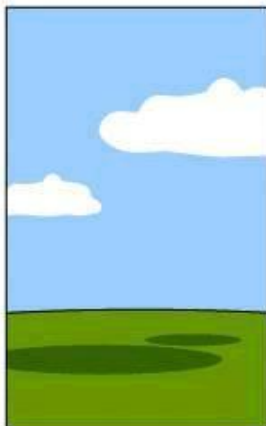
Como o analista projetou...



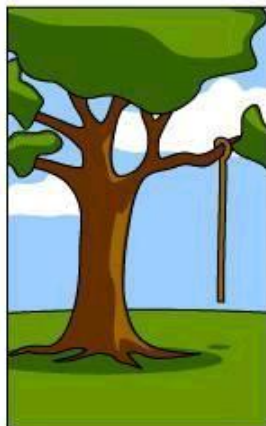
Como o programador construiu...



Como o Consultor de Negócios descreveu...



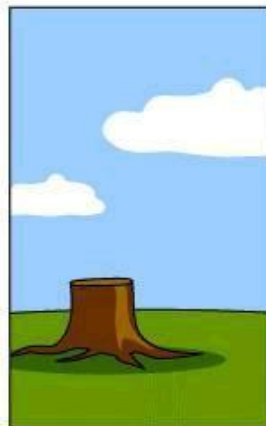
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...

IDENTIFICANDO REQUISITOS: DOMÍNIO DE NEGÓCIO

- O primeiro passo é delimitar e entender o **domínio de negócio**. Também chamado de domínio do problema ou domínio da aplicação, é a área específica ou “tema” que o software aborda.
- Exemplos:
 - Sistemas de controle de vendas e estoque
 - Mensageiros instantâneos
 - Aplicativos de entrega
 - Jogos (e seu gênero específico)
- É altamente recomendável incluir na equipe ao menos um **stakeholder especializado no domínio de negócio**.

TIPOS DE REQUISITOS

- Requisitos Funcionais
- Requisitos Não-Funcionais



REQUISITOS FUNCIONAIS

- As funcionalidades e informações que o sistema deve **fazer ou ter**.
- Notação e exemplos:
 - [RF001] O sistema deve cadastrar usuários.
 - [RF002] O sistema deve emitir relatório de atividades para cada usuário.
 - [RF003] O usuário deve poder consultar seus dados no sistema.

REQUISITOS NÃO-FUNCIONAIS

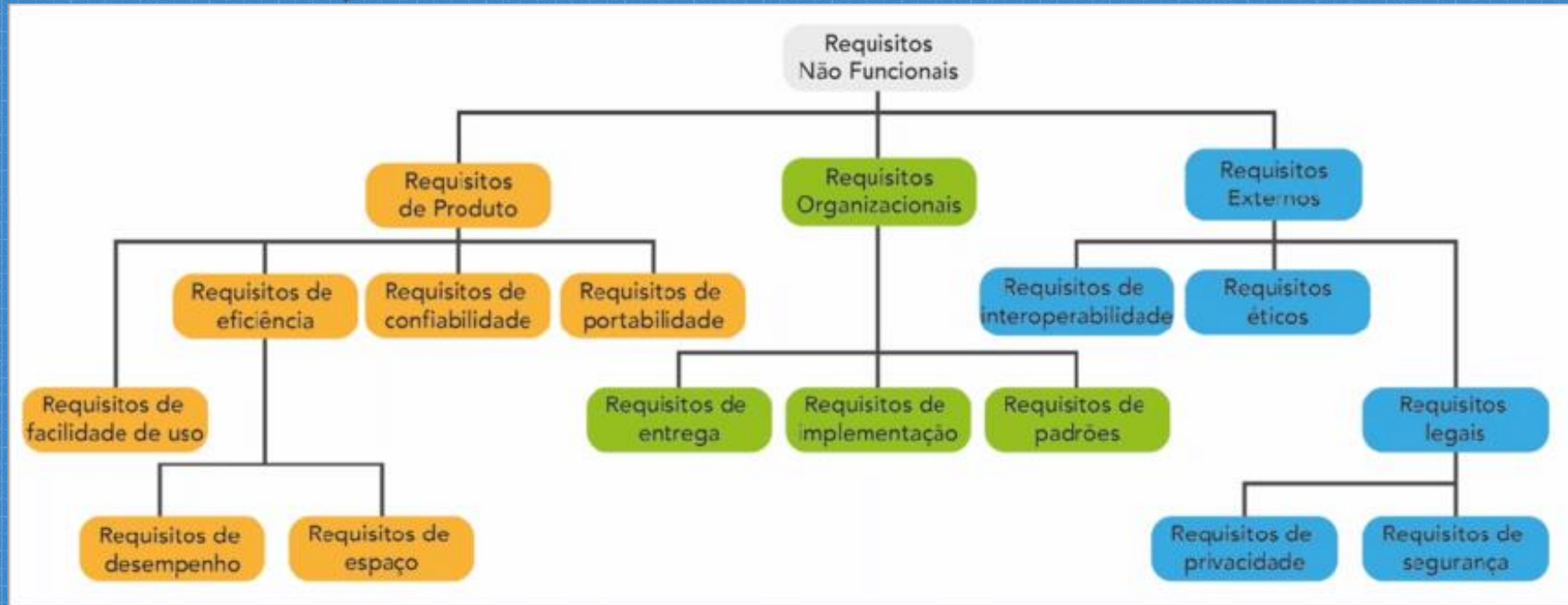
- Não são funcionalidades explícitas do sistema, mas propriedades relevantes para o desenvolvimento.
- **Uma dica importante** é que os requisitos não funcionais são **geralmente** mensuráveis e assim devemos preferencialmente associar uma medida ou referência para cada requisito não funcional.
- Notação e exemplos:
 - [RNF001] O sistema será desenvolvido completamente ou predominantemente em Java.
 - [RNF002] O sistema não deve ultrapassar, quando instalado, 20GB de armazenamento interno.
 - [RNF003] Todas as variáveis numéricas do sistema devem ser de 64-bits.

REQUISITOS NÃO-FUNCIONAIS: TIPOS

- Não existe uma definição formal ou uma lista completa de requisitos não-funcionais.
- A separação entre diferentes tipos de requisitos não-funcionais ajuda os stakeholders a se aprofundarem nas especificações.
- Na dúvida, utilize as separações que a empresa utiliza, ou que seja acordado entre os stakeholders.
- Sempre seja o mais específico possível ao classificar requisitos.

REQUISITOS NÃO-FUNCIONAIS: TIPOS

Para fins avaliativos, seguiremos esse modelo



REQUISITOS NÃO-FUNCIONAIS: EXEMPLOS

❑ Requisitos de produtos:

- **Requisitos de facilidade de uso:** Todo o sistema deve ser instalado e configurado a partir de um único instalador.
- **Requisitos de eficiência:**
 - **Requisitos de desempenho:** O sistema deve gerar notas fiscais em até 30 segundos.
 - **Requisitos de espaço:** Quando instalado, o sistema não deve ocupar mais do que 20GB de armazenamento interno.
- **Requisitos de confiabilidade:** O sistema deve se manter online 24/7.
- **Requisitos de portabilidade:** O sistema deve rodar nos sistemas operacionais Windows e Linux.

REQUISITOS NÃO-FUNCIONAIS: EXEMPLOS

□ Requisitos organizacionais:

- **Requisitos de entrega:** Um relatório sobre o andamento do projeto deve ser entregue toda sexta-feira.
- **Requisitos de implementação:** O sistema deve ser escrito completamente ou predominantemente em linguagem Java.
- **Requisitos de padrões:** O projeto deve seguir o processo de prototipagem.

REQUISITOS NÃO-FUNCIONAIS: EXEMPLOS

□ Requisitos externos:

- **Requisitos de interoperabilidade:** O sistema deve comunicar-se com SQL Server.
- **Requisitos éticos:** O uso de testes A/B deve ser informado aos usuários.
- **Requisitos legais:**
 - **Requisitos de privacidade:** O sistema não utilizará informações privadas sem o consentimento do usuário.
 - **Requisitos de segurança:** O mensageiro instantâneo deve ser protegido com criptografia de ponta a ponta.

REQUISITOS NÃO-FUNCIONAIS: EXEMPLOS

□ Requisitos externos:

- **Requisitos de interoperabilidade:** O sistema deve comunicar-se com SQL Server.
- **Requisitos éticos:** O uso de testes A/B deve ser informado aos usuários.
- **Requisitos legais:** O jogo seguirá classificação indicativa +18.
 - **Requisitos de privacidade:** O sistema não utilizará informações privadas sem o consentimento do usuário.
 - **Requisitos de segurança:** O mensageiro instantâneo deve ser protegido com criptografia de ponta a ponta.

O QUE OBTEMOS AO FINAL DA FASE DE REQUISITOS

- Um documento contendo todos os requisitos do sistema.
- Muitas vezes esse documento é escrito em notação informal (linguagem natural) ou em forma de Caso de Uso, para que pessoas não técnicas possam compreendê-lo facilmente.
- O documento de requisitos estabelece o escopo do sistema (o que faz parte e o que não faz parte do sistema).