

Requisitos de *Software*

Prof.^a Ma. Jessica Oliveira

Aula 02 – 24/02/2025

Tipos de requisitos e suas características.

O que são requisitos de *software*?

- Representam as **funcionalidades, restrições e expectativas** que o sistema deve atender para satisfazer as necessidades dos usuários e do negócio.
- Eles funcionam como um contrato entre os *stakeholders* (clientes, usuários finais, desenvolvedores e analistas de requisitos), **garantindo que todas as partes compartilhem um entendimento comum sobre o que será desenvolvido.**
- A definição correta dos requisitos minimiza ambiguidades e reduz riscos associados ao desenvolvimento, garantindo que o sistema entregue valor real aos usuários.

Tipos de requisitos de *software*

Requisitos funcionais: definição.

- Especificam o que o sistema **deve fazer**.
- Descrevem as funcionalidades e comportamentos esperados, incluindo interações com usuários e outros sistemas.

Requisitos funcionais: características.

- Definem **ações** e **comportamentos** do sistema;
- Devem ser **específicos** e **mensuráveis**;
- Representam as **interações** entre usuário e sistema.

Requisitos funcionais: exemplos.

- O sistema deve permitir que os usuários realizem *login* utilizando *e-mail* e senha.
- Após a autenticação, o usuário deve conseguir visualizar sua conta bancária e consultar o saldo disponível.
- O sistema deve gerar relatórios financeiros em formato PDF para *download*.

Requisitos não funcionais: definição.

- Especificam como o sistema **deve operar**, estabelecendo restrições sobre desempenho, usabilidade, confiabilidade e segurança.

Requisitos não funcionais: características.

- Definem **propriedades** do sistema, como desempenho e segurança;
- Afetam a **experiência** do usuário e a confiabilidade do *software*;
- Devem ser **quantificáveis** sempre que possível para facilitar a validação.

Requisitos não funcionais: exemplos.

- O sistema deve ser capaz de processar 10.000 requisições simultâneas sem degradação de desempenho.
- A interface do sistema deve estar acessível para pessoas com deficiência visual, seguindo normas de acessibilidade.
- O tempo de resposta para qualquer operação crítica não deve ultrapassar 2 segundos.

Requisitos de negócio: definição.

- Definem os **objetivos estratégicos** da organização que motivam o desenvolvimento do sistema.
- Eles **expressam a razão pela qual um *software* está sendo criado** e os benefícios esperados para o negócio.

Requisitos de negócio: características.

- Relacionam-se aos objetivos estratégicos da organização;
- Nem sempre são implementáveis diretamente no código-fonte do sistema;
- Influenciam os requisitos funcionais e não funcionais.

Requisitos de negócio: exemplos.

- O sistema deve reduzir em 30% o tempo médio de processamento de pedidos da empresa.
- A plataforma deve permitir a automação da gestão de contratos, diminuindo custos operacionais em 15%.

Requisitos de usuário: definição.

- Descrevem as **necessidades e expectativas dos usuários** em relação ao sistema.
- Eles servem como uma ponte entre os requisitos de negócio e os requisitos do sistema.

Requisitos de usuário: características.

- Devem ser descritos em linguagem acessível para os usuários finais;
- Muitas vezes são documentados como histórias de usuário em metodologias ágeis;
- São refinados para se tornarem requisitos funcionais e não funcionais.

Requisitos de usuário: exemplos.

- O sistema deve oferecer uma interface intuitiva e responsiva para facilitar a navegação do usuário.
- O usuário deve conseguir visualizar e editar suas informações pessoais sem necessidade de suporte técnico.

Requisitos de sistema: definição.

- Detalham os **componentes técnicos e arquiteturais** necessários para que o software funcione corretamente.
- Eles representam a **especificação mais técnica** dos requisitos de software.

Requisitos de sistema: características.

- São detalhados e técnicos;
- Estão diretamente relacionados à implementação do *software*;
- São derivados dos requisitos funcionais e não funcionais.

Requisitos de sistema: exemplos.

- O sistema deve ser desenvolvido utilizando a linguagem Java e o banco de dados PostgreSQL.
- O sistema deve suportar a integração com APIs REST para comunicação com sistemas externos.

Critérios de qualidade dos requisitos

Clareza.

- Os requisitos devem ser descritos de maneira objetiva e sem ambiguidades. Frases vagas podem levar a diferentes interpretações.
- **Ruim:** "O sistema deve ser rápido."
- **Bom:** "O tempo de resposta para consultas não deve ultrapassar 2 segundos."

Compleitude.

- Os requisitos devem cobrir todas as funcionalidades e restrições necessárias para o sistema.
- **Ruim:** "O usuário pode fazer *login*." (Não especifica como ou quais métodos de autenticação são aceitos.).
- **Bom:** "O usuário deve fazer *login* utilizando e-mail e senha ou autenticação via Google."

Consistência.

- Os requisitos não devem ser contraditórios entre si.
- Se um requisito disser que "O sistema deve exibir relatórios em PDF" e outro disser que "O sistema não deve suportar arquivos PDF", há uma inconsistência.

Rastreabilidade.

- Os requisitos devem ser rastreáveis, permitindo que cada um seja associado às suas origens (documentos de negócio, requisitos de usuário, código implementado, etc.).

Importância da definição precisa dos requisitos

Problemas causados...

- No dia a dia das empresas de *software*, a consequência mais comum dos problemas em requisitos é o aumento do retrabalho, ou seja, a necessidade de refazer coisas que já estavam prontas, levando a custos não previstos inicialmente.
- Quanto mais atividade já tiver sido realizada sobre o requisito (especificação, implementação, testes, entrega), mais difícil e trabalhoso é consertar e maiores são os impactos negativos para o projeto ou para a *sprint*.

Os problemas estão relacionados ao fato de que um requisito pode:

- Estar incompleto;
- Estar em um nível de detalhe insuficiente para as etapas seguintes do ciclo de desenvolvimento;
- Conter ambiguidades ou imprecisões que levem os membros da equipe a interpretá-lo de forma diferente do que o esperado, levando a erros nas fases seguintes do ciclo de desenvolvimento;
- Ser incompatível com outro requisito;
- Ser tecnicamente inviável;
- Ser difícil de testar e validar;

Os problemas estão relacionados ao fato de que um requisito pode:

- Ter priorização conflitante sob a ótica dos diversos *stakeholders*;
- Os objetivos de negócio, a visão e o escopo do projeto nunca foram definidos claramente;
- Os clientes estavam muito ocupados para dedicar mais tempo trabalhando com os analistas ou os desenvolvedores nos requisitos;
- O time não pode interagir diretamente com usuários representativos para entender suas necessidades;
- Os clientes argumentaram que todos os requisitos eram críticos, por isso eles não estabeleceram prioridades;

Os problemas estão relacionados ao fato de que um requisito pode:

- Os desenvolvedores encontraram ambiguidades e informações faltantes quando foram codificar, então, eles precisaram adivinhar certas informações;
- As comunicações entre o cliente e os desenvolvedores focaram na aparência da interface do usuário e não no que os usuários necessitavam alcançar com o software;
- Os clientes nunca aprovaram os requisitos;
- Os usuários aprovaram os requisitos para um release ou iteração e então mudaram eles continuamente;

Os problemas estão relacionados ao fato de que um requisito pode:

- O escopo do projeto aumentou à medida que as mudanças de requisitos foram aceitas, mas o cronograma se desviou porque nenhum recurso adicional foi fornecido e nenhuma funcionalidade foi removida;
- As mudanças solicitadas nos requisitos foram perdidas; ninguém sabia o status de solicitações de mudanças específicas;
- Os clientes requisitaram certa funcionalidade e os desenvolvedores a desenvolveram, mas ninguém nunca utilizou;
- Ao final do projeto, a especificação foi satisfeita, mas o usuário e os objetivos de negócio não.

E por que esses problemas acontecem com os requisitos?

- Tipo de projeto;
- Contexto de negócio com as características da equipe e das tecnologias envolvidas;
- Problemas de comunicação.

Lembrem-se:

“A parte mais difícil ao se construir um sistema de *software* é decidir precisamente o que construir. Nenhuma outra parte do trabalho conceitual é tão difícil quanto estabelecer os requisitos técnicos detalhados, incluindo a interface com as pessoas, máquinas e outros sistemas de software. Nenhuma outra parte afeta tanto o sistema resultante se for feita de forma errada. Nenhuma outra parte é tão difícil de consertar depois.”

(BROOKS, 1987, p. 13)

Para o pós-aula...

Desenvolver um mapa mental que ilustre:

- Os diferentes tipos de requisitos (funcionais, não funcionais, de negócio, de usuário e de sistema);
- Suas principais características e diferenças;
- Exemplos práticos de cada tipo.

No mesmo arquivo, explicar o que foi possível compreender sobre a importância da definição precisa dos requisitos, em um parágrafo de, no mínimo, 300 palavras.

Dúvidas?

jessica.oliveira@p.ucb.br