

A classificação dos requisitos não se resume apenas na separação entre funcionais e não funcionais. Existem também os:

Requisitos Voláteis:

Definição: Requisitos voláteis são aqueles que têm maior probabilidade de mudar ao longo do ciclo de vida do projeto. Essas mudanças podem ser causadas por atualizações nas necessidades dos usuários, novos insights ou evoluções nos requisitos de mercado. É importante monitorar e gerenciar esses requisitos de forma ágil.

Exemplo: Um requisito volátil pode ser a interface do usuário, que pode precisar de ajustes à medida que os usuários fornecem feedback durante o desenvolvimento.

Requisitos Estáveis:

Definição: Requisitos estáveis são aqueles que têm pouca probabilidade de sofrer mudanças significativas durante o projeto. Eles foram bem definidos e acordados entre as partes interessadas desde o início do projeto e não são suscetíveis a revisões frequentes.

Exemplo: Um requisito estável pode ser a funcionalidade central de um sistema que foi especificada detalhadamente no início do projeto e não precisa de alterações significativas.

Requisitos de Usuário:

Definição: Requisitos de usuário são aqueles que refletem as necessidades e expectativas dos usuários finais do sistema. Eles descrevem as funcionalidades e características que os usuários desejam ver no software para atender às suas necessidades.

Exemplo: Um requisito de usuário pode ser "Os usuários devem poder salvar documentos em diferentes formatos, como PDF e DOCX."

Requisitos de Sistema:

Definição: Requisitos de sistema são especificações técnicas que detalham os recursos, comportamentos e interações do sistema como um todo. Eles não se concentram nas necessidades dos usuários, mas sim nas especificações técnicas do sistema.

Exemplo: Um requisito de sistema pode ser "O sistema deve ser executado em servidores Linux e usar uma arquitetura de microservices"

Requisitos Inversos:

Definição: Requisitos inversos, às vezes chamados de "requisitos de restrição" ou "requisitos de exclusão", especificam o que o sistema não deve fazer ou as

restrições que devem ser aplicadas. Eles são úteis para estabelecer limites claros para o sistema.

Exemplo: Um requisito inverso pode ser "O sistema não deve permitir o acesso a dados confidenciais sem autenticação adequada."