

ANÁLISE E PROJETO DE SISTEMAS

Aula 6

Felipe Marx Benghi

Requisitos de Usuário e de Sistema

fbenghi@up.edu.com

<https://github.com/fbenghi/AnaliseProjetos2024-1>

Objetivos

- Linguagem Natural Estruturada
 - Requisitos de usuário
 - Requisitos de sistema

Engenharia de Requisitos

Linguagem Natural Estruturada

- Requisitos são escritos de maneira padronizada. Exemplo:
`Alguém` deve `fazer algo` `em certas condições`
- Limita a liberdade do redator de requisitos

Vantagens:

- Mantém grande parte da expressividade e compreensibilidade da linguagem natural
- Garante que alguma uniformidade seja imposta à especificação

Linguagem Natural Estruturada

Iremos especificar os requisitos sob as três perspectivas:

- Negócio: BPMN
- Usuário: histórias de usuário (user stories);
- Sistema: especificação detalhada dos requisitos.

História do usuário

- Afirmação sobre o que o usuário quer fazer com uma característica de software, escrito na perspectiva de um usuário;
- Não deve conter termos técnicos ou objetivos do projeto - linguagem de negócios compreensível a todos;
- Concentra-se em quem, o quê e o porquê de um recurso, não como.

História do usuário

Deve responder a 3 perguntas:

- Quem se beneficia?
 - Parte interessada que se beneficia da história do usuário (ator).
- O que se quer?
 - Visão alto nível da funcionalidade para o usuário (descrição).
- Qual é o benefício?
 - Valor de negócio que a história proporciona (porquê)

História do usuário

Formato sugerido: Como **papel**, eu quero **algo** para **me beneficiar**.

- Como cliente, quero consultar o catálogo, para que eu possa encontrar o produto que desejo comprar.
- Como vice-presidente de vendas, eu quero rever o desempenho histórico de vendas, para identificar regiões geográficas e produtos de melhor desempenho.
- Como CEO, quero que o sistema funcione em diferentes plataformas, para que eu possa atingir uma quantidade maior de clientes.

História do usuário

Alguns cuidados:

- Manifestar objetivos de pequena granularidade que permitam a estimativa do investimento na entrega do software correspondente sem perder uma visão dos resultados associados.
- Não aborda explicitamente como documentar requisitos não funcionais.
- Pode não ser adequada para ambientes mais burocráticos ou quando o nível de documentação do projeto é imposto.

História do usuário - Critérios de qualidade

Modelo 3Cs – Card, Conversation, Confirmation

- Cartão:
 - Pequena o suficiente para caber em um cartão?
- Conversação:
 - Promove a comunicação entre o usuário e o time?
 - Proporciona um entendimento comum da funcionalidade?
- Confirmação:
 - Consta qual o comportamento esperado para confirmar o seu escopo ou o plano de teste?

História do usuário - Critérios de qualidade

Critério INVEST, proposto por Bill Wake em 2003:

	Significado	Descrição
I	Independent (independente)	A história deve ser autossuficiente, sem dependência com outra história
N	<u>Negotiable</u> (negociável)	Podem ser sempre alteradas ou reescritas
V	<u>Valuable</u> (valiosa)	Deve entregar valor para o usuário final
E	<u>Estimable</u> (estimável)	Deve ser sempre possível estimar o tamanho
S	<u>Small</u> (pequena)	Não devem ser grandes a ponto de que seja impossível estimar ou priorizar
T	<u>Testable</u> (testável)	Deve ser possível de ser testada

História do usuário - Prioridade

Análise MoSCoW: define a prioridade para a próxima entrega:

- Must have [deve ter]: necessária
- Should have [deveria ter]: desejável, esperada
- Could have [poderia ter]: desejável, não esperada
- Would have [teria]: fora de escopo

História do usuário - Prioridade

História de Usuário		Prioridade
US001	Como atendente de balcão, quero um módulo para realizar o cadastro do cliente, para que eu possa solicitar a entrega da pizza	Must
US002	Como atendente de balcão, quero um módulo para registrar o pedido da pizza e acompanhamentos, para encaminhar para a cozinha	Must
US003	Como gerente, quero um relatório mensal de pizzas categorizado por sabor, para que eu possa identificar o perfil de consumo dos clientes	<u>Could</u>
US004	Como CEO, quero que o sistema funcione em diferentes plataformas, para que eu possa atingir uma quantidade maior de clientes	<u>Would</u>

Requisitos de Sistema

- Descrições mais detalhadas das funções, serviços e restrições operacionais do sistema de software.
- O documento de requisitos do sistema (às vezes chamado de especificação funcional) deve definir exatamente o que será implementado.
- Ele pode fazer parte do contrato entre o comprador do sistema e os desenvolvedores de software
- Requisitos funcionais e não-funcionais

Requisitos Funcional - Detalhado

Segundo SOMMERVILLE (2016, p. 123), um formato padrão usando linguagem estruturada deve conter os seguintes itens:

- Função ou entidade que está sendo especificada;
- Entradas e origens dessas entradas;
- saídas e destinos dessas saídas;
- Ação a ser tomada;
- Informações necessárias;
- Pré-condições e pós-condições para completar a ação;
- Descrição dos efeitos colaterais.

Requisitos Funcional - Detalhado

Software de Controle/Dose de Insulina/SRS/3.3.2	
Função	Calcular dose de insulina: nível seguro de açúcar.
Descrição	Calcula a dose de insulina que deve ser administrada quando o nível atual de açúcar está na zona segura entre 3 e 7 unidades.
Entradas	Leitura do nível atual de açúcar (r2), duas leituras anteriores (r0 e r1).
Origem	Leitura do nível atual de açúcar através de sensor. Outras leituras da memória.
Saídas	<u>CompDose</u> – a dose de insulina a ser administrada.
Destino	Laço de controle principal.
Ação	<u>CompDose</u> é zero se o nível de açúcar está estável ou caindo ou se o nível está subindo mas a taxa de aumento está diminuindo. Se o nível está aumentando e a taxa de aumento está aumentando, então <u>CompDose</u> é calculado dividindo-se a diferença entre o nível de açúcar e o nível anterior por 4 e arredondando o resultado.
Requer	Duas leituras prévias para que a taxa de mudança do nível de açúcar seja calculado.
Pré-condição	O reservatório de insulina contenha ao menos o máximo permitido para uma dose única de insulina.
Pós-condição	r0 é substituído por r1 e então r1 é substituído por r2.
Efeitos colaterais	Nenhum.

Requisitos Funcional - Detalhado

Software de Pizzaria/Pedido de Pizza	
Função	Registrar o pedido de pizza.
Descrição	Permite o registro de um pedido de pizza, com detalhes da pizza e de acompanhamentos.
Entradas	Identificação do cliente; tamanhos e combinações de sabores da pizza; complementos (borda, modificações); acompanhamentos (bebidas).
Origem	Leitura dos dados do cliente; leitura do detalhamento do pedido; outras leituras da memória.
Saídas	Número do pedido, identificação do cliente, dados da pizza, dados dos acompanhamentos.
Destino	Tabela “pedido” no banco de dados; painel de acompanhamento na cozinha; recibo para o cliente.
Ação	Tendo como entrada os dados de identificação do cliente, são solicitados os dados do pedido: tamanho e combinação de sabores da pizza, escolha de complemento ou retirada de itens, escolha de acompanhamentos; os dados do pedido são registrados em banco de dados; os dados são exibidos na cozinha para produção da pizza, e é emitido um recibo para o cliente.
Requer	Cadastramento prévio do cliente para identificação (e telefone + endereço se for para entrega).
Pré-condição	Ter entrado no menu “Registrar → Pedido” e estar com a tela de registro de pedidos aberta.
Pós-condição	O pedido é registrado na tabela específica do banco de dados, os dados são exibidos no painel e o recibo é gerado.
Efeitos colaterais	Durante a geração do número de pedido, o sistema fica indisponível por alguns segundos.

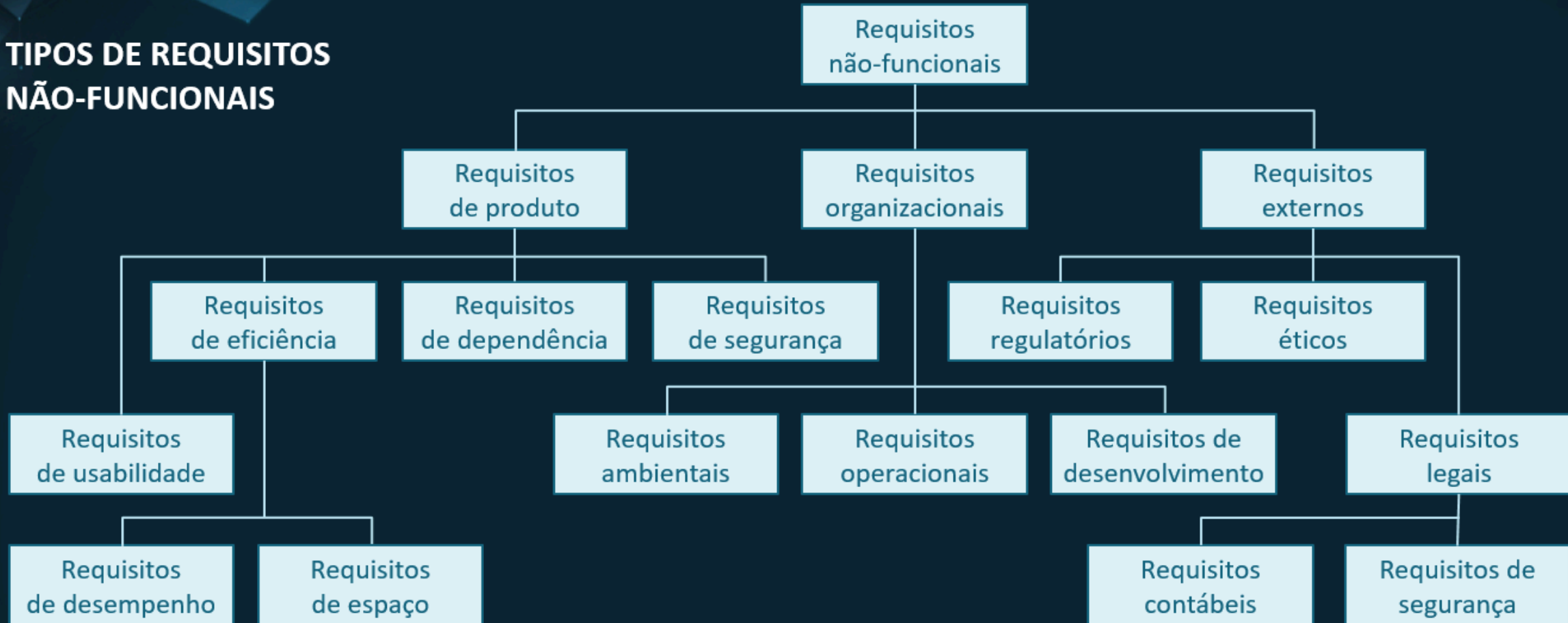
Requisitos Não-Funcional

Formato adotado na disciplina:

Alguém deve fazer algo em certas condições

Requisitos Não-Funcional Tipos

TIPOS DE REQUISITOS NÃO-FUNCIONAIS



Requisitos Não-Funcional - Métricas

- Velocidade
 - Transações processadas por segundo
 - Tempo de resposta do usuário/evento
 - Tempo de atualização da tela
- Tamanho
 - Megabytes (Mbytes)
 - Número de chips ROM

Requisitos Não-Funcional - Métricas

- Facilidade de uso
 - Tempo de treinamento
 - Número de quadros de ajuda
- Confiabilidade
 - Tempo médio até a falha
 - Probabilidade de indisponibilidade
 - Taxa de ocorrência de falhas
 - Disponibilidade

Requisitos Não-Funcional - Métricas

- Robustez
 - Tempo para reiniciar após falha
 - Porcentagem de eventos de falha
 - Probabilidade de corrupção de dados
- Portabilidade
 - Sistemas de destino

Requisitos Não-Funcional - Exemplos

Requisito	Descrição	Característica
RNF001	O <u>back end</u> deve operar em ambiente Windows	Portabilidade
RNF002	O <u>front end</u> deve operar em dispositivos móveis iOS e Android	Portabilidade
RNF003	A base de dados não pode ultrapassar 2GB mensais	Desempenho
RNF006	A autenticação deve acontecer por biometria	Segurança, Eficiência

FIM