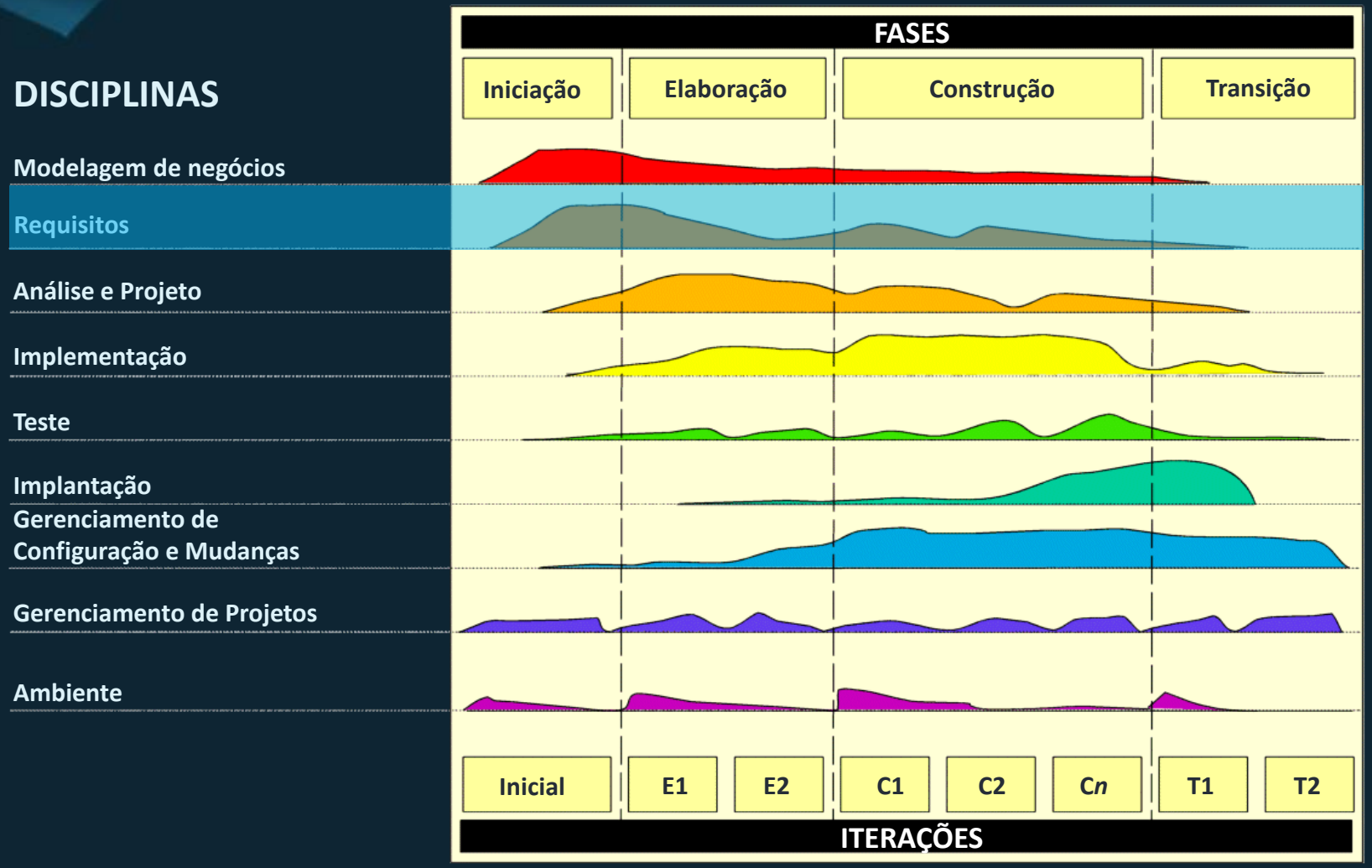
Universidade POSITIVO

ENGENHARIA DE REQUISITOS

ANÁLISE E PROJETO DE SISTEMAS

Prof. Evandro Zatti, M. Eng.

PROCESSO UNIFICADO



fonte:
KRUCHTEN, 2003
(adaptado)

FUNDAMENTOS

"Antes de você iniciar qualquer trabalho técnico, é uma boa ideia criar um conjunto de **requisitos** para quaisquer das atividades de engenharia. Estas tarefas levam a um entendimento do impacto de negócio que o software terá, o que o usuário deseja, e como os usuários finais irão interagir com o software."

(PRESSMAN e MAXIM, 2015, p. 131)

FUNDAMENTOS

- **Requisito**: é uma condição para se alcançar determinado fim.
- Os requisitos de um software são descrições dos **serviços** por ele fornecidos e as suas **restrições operacionais**.
- O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado de **Engenharia de Requisitos**.

PROCESSO DE ENGENHARIA DE REQUISITOS

A Engenharia de Requisitos compreende **três etapas**...

1. Levantamento de Requisitos;
2. Especificação de Requisitos;
3. Validação de Requisitos.

... sob **três perspectivas** diferentes:

1. Negócio;
2. Usuário;
3. Sistema.

REQUISITOS DE NEGÓCIO

- É o que o sistema deve conter para atender às necessidades do negócio;
- O **mapeamento de processos de negócio** evidencia em que momento o software irá contribuir.

REQUISITOS DE USUÁRIO

- São declarações em uma **linguagem natural** com **diagramas**, de quais **serviços** são esperados do sistema e as **restrições** sob as quais ele deve operar, sob a visão do usuário.
- Problemas que podem surgir:
  - ✓ Falta de Clareza;
  - ✓ Confusão de Requisitos;
  - ✓ Fusão de Requisitos.

REQUISITOS DE SISTEMA

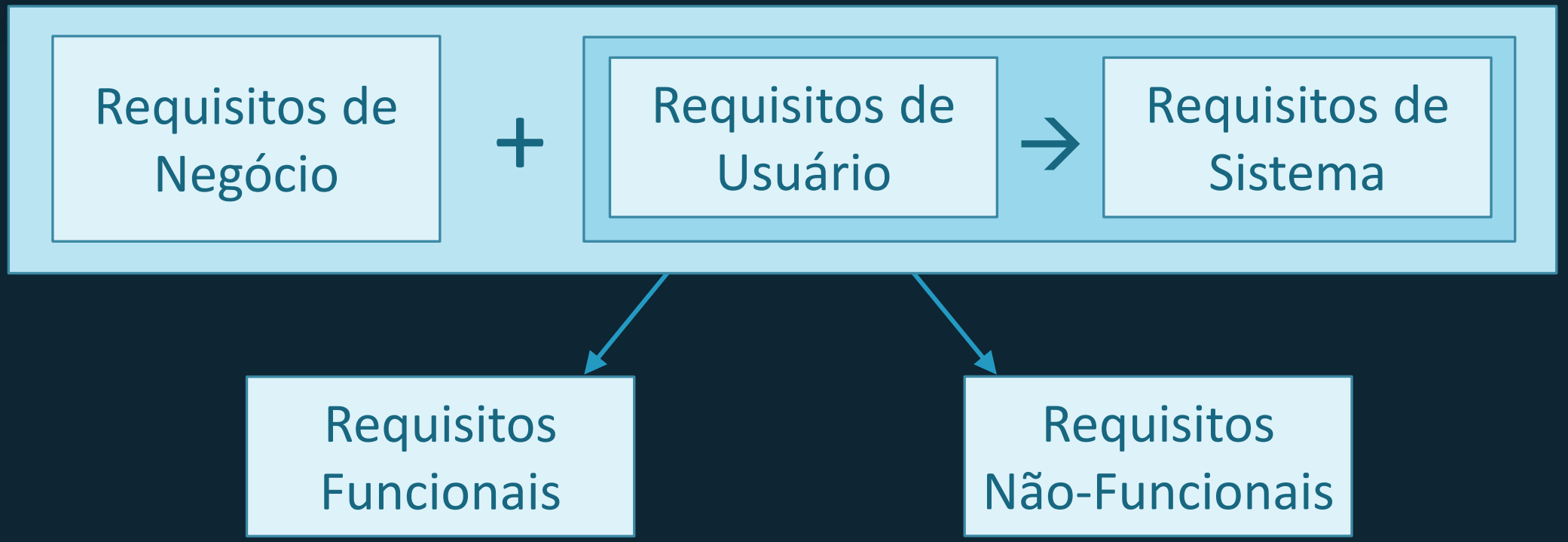
- São **versões expandidas** dos requisitos de usuários usados pelos engenheiros de software como um ponto de partida para o projeto do sistema.
- Definem **detalhadamente** as funções, os serviços e as restrições operacionais do sistema.
- O documento de requisitos de sistema (às vezes chamado de **especificação funcional**) deve ser preciso; ele deve definir exatamente o que será implementado.

PROCESSO DE ENGENHARIA DE REQUISITOS

VISÃO ESPIRAL DO PROCESSO DE ENGENHARIA DE REQUISITOS

fonte:
SOMMERVILLE, 2016,
p. 108 (adaptado)

CLASSIFICAÇÃO



REQUISITOS FUNCIONAIS

- São as declarações de **serviços** que o software deve fornecer, como o software deve reagir a entradas específicas e como o software deve se comportar em determinadas situações.
- Em alguns casos, os requisitos funcionais podem também estabelecer explicitamente o que o software **não deve** fazer.
- A especificação de um requisito funcional deve determinar o que se espera que o **software faça** (ou não), **sem** a preocupação de **como** ele faz.

REQUISITOS FUNCIONAIS

- Exemplos:
  - ✓ O software deve possibilitar o cálculo dos gastos diários, semanais, mensais e anuais com pessoal.
  - ✓ O software deve emitir relatórios de compras a cada quinze dias.
  - ✓ Os usuários devem poder obter o número de aprovações, reprovações e trancamentos em todas as disciplinas por um determinado período de tempo.

REQUISITOS NÃO-FUNCIONAIS

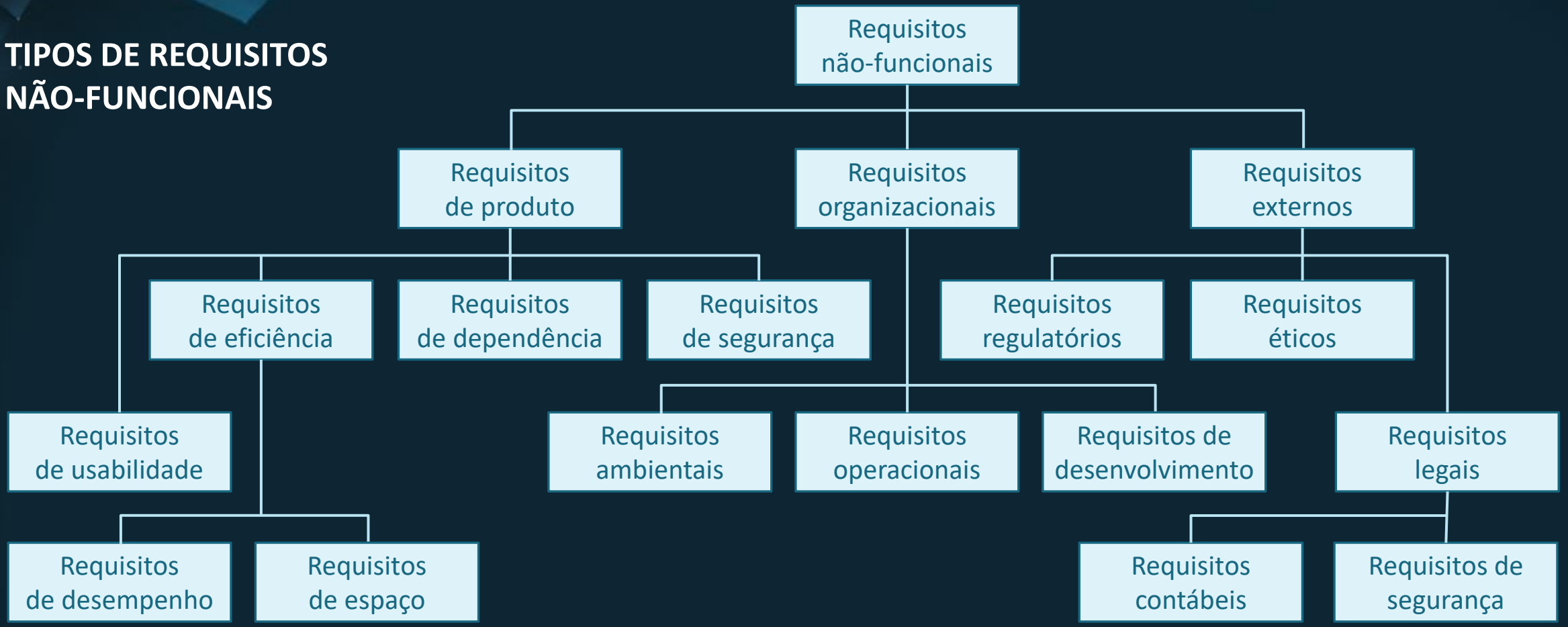
- São **restrições** sobre os serviços ou as funções oferecidas pelo software. Eles incluem restrições de *timing*, restrições sobre o processo de desenvolvimento e padrões.
- Os requisitos não funcionais aplicam-se frequentemente ao software **como um todo**.

REQUISITOS NÃO-FUNCIONAIS

- Exemplos:
  - ✓ A base de dados deve ser protegida para acesso apenas de usuários autorizados.
  - ✓ O tempo de resposta do sistema não deve ultrapassar 30 segundos.
  - ✓ O software deve ser operacionalizado no sistema Linux.
  - ✓ O tempo de desenvolvimento não deve ultrapassar seis meses.

REQUISITOS NÃO-FUNCIONAIS

TIPOS DE REQUISITOS NÃO-FUNCIONAIS



fonte: SOMMERVILLE, 2016, p. 108 (adaptado)

QUALIDADE DE SOFTWARE X REQUISITOS

- A **qualidade** de um produto ou serviço está diretamente relacionada a atender às **expectativas** do cliente;

EXPECTATIVA = REQUISITO

- Para atender aos requisitos (tanto funcionais quanto não funcionais), deve-se atentar para as **normas** de qualidade em software.

NORMAS E MODELOS DE QUALIDADE

- As principais instituições utilizadas como referência para normas relacionadas a software são:
  - ✓ **ISO**: *International Organization for Standardization* (Organização Internacional para Padronização);
  - ✓ **IEC**: *International Electrotechnical Commission* (Comissão Eletrotécnica Internacional);
  - ✓ **IEEE***: *Institute of Electrical and Electronics Engineers* (Instituto de Engenheiros Eletricistas e Eletrônicos).] ** pronuncia-se I-3-E*
  - ✓ **ABNT**: Associação Brasileira de Normas Técnicas.

NORMA ISO/IEC 25010 (2011)

Substituta da ISO/IEC 9126-1 (2001)

- Características de qualidade de software (NBR 13596):
  - ✓ **Funcionalidade**: finalidade do produto;
  - ✓ **Usabilidade**: esforço para utilizar o produto;
  - ✓ **Confiabilidade**: frequência de falhas e recuperabilidade;
  - ✓ **Eficiência**: desempenho;
  - ✓ **Manutenibilidade**: esforço necessário para modificar o produto;
  - ✓ **Portabilidade**: capacidade de transferir o produto para outros ambientes;
  - ✓ **Segurança**: confidencialidade e integridade;
  - ✓ **Compatibilidade**: coexistência e interoperabilidade.

NORMA ISO/IEC 25010 X REQUISITOS NÃO-FUNCIONAIS

- A primeira característica da norma (funcionalidade) diz respeito aos requisitos funcionais;
- As demais (7) características dizem respeito aos requisitos não funcionais.

LEVANTAMENTO DE REQUISITOS

- Sob a perspectiva do Design Thinking:
  - ✓ Triangulação das pesquisas

Entrevistas
O que as pessoas dizem que fazem

Olhar etnográfico
Olhar o que as pessoas fazem

Design participativo
Fazer o que as pessoas fazem

TÉCNICAS DE LEVANTAMENTO DE REQUISITOS

- JAD;
- Entrevista;
- *Brainstorming*;
- Questionário;
- Observação Direta;
- Análise de Documentos.

JAD

- JAD (*Joint Application Design/Dev.*) é uma técnica para promover **cooperação**, entendimento e trabalho em **grupo** entre os usuários;
- Facilita a criação de uma **visão compartilhada** do que o produto de software deve ser;
- Os desenvolvedores ajudam os usuários a formular problemas e explorar soluções; dessa forma, os usuários ganham um sentimento de envolvimento, posse e responsabilidade com o sucesso do produto;

JAD

- Princípios:
  1. Dinâmica de **grupo**: são realizadas reuniões com um líder experiente, analista, usuários e gerentes, para despertar a força e criatividade dos participantes
  2. Uso de **técnicas visuais**: para aumentar a comunicação e o entendimento;
  3. Manutenção do **processo organizado** e **racional**: o JAD emprega a análise *top-down* e atividades bem definidas;
  4. Utilização de **documentação padrão**: preenchida e assinada por todos os participantes.

ENTREVISTA

- É uma das técnicas tradicionais mais simples de utilizar e que produz bons resultados na fase inicial de obtenção de dados;
- É indicada para **pequeno número** de **envolvidos**;
- Convém que o entrevistador dê margem ao entrevistado para expor as suas ideias;
- É necessário ter um plano de entrevista para que não haja dispersão do assunto principal e a entrevista fique longa;
- Deve ser conduzida em ambiente **informal** e de **confiança mútua**.

BRAINSTORMING

- É uma técnica para **geração de ideias**;
- Ela consiste em uma ou várias reuniões que permitem que as pessoas sugiram e explorem ideias.
- Regras:

1. **Uma** conversa **por vez**
2. **Não critique** nem **julgue**
3. **Encoraje** as ideias **doidas**
4. **Construa** sobre a **ideia dos outros**
5. Seja **visual**
6. Mantenha o **foco**: fique no assunto proposto
7. **Quantidade** importa: crie o máximo de ideias possíveis

QUESTIONÁRIO

- O uso de questionário é indicado, por exemplo, quando há **diversos grupos** de usuários que podem estar em **diversos locais** diferentes do país;
- Neste caso, elaboram-se pesquisas específicas de acompanhamento com usuários selecionados, que a contribuição em potencial pareça mais importante, pois não seria prático entrevistar todas as pessoas em todos os locais;
- Pode ser com perguntas **abertas** ou **fechadas**.

OBSERVAÇÃO DIRETA

- Consiste em **observar** a **rotina** do departamento/empresa/setor/pessoa;
- É necessário autorização prévia;
- Importante obter informações preliminares para não atrapalhar a rotina.

ANÁLISE DE DOCUMENTOS

- Consiste na análise de **documentos existentes** como:
  - ✓ Registros;
  - ✓ Cadastros;
  - ✓ Relatórios;
  - ✓ Outros documentos relevantes.

ESPECIFICAÇÃO DE REQUISITOS

- É o processo de registrar os requisitos de **usuário** e de **sistema** em um documento, em uma linguagem:
  - ✓ Clara e não-ambígua;
  - ✓ Correta;
  - ✓ Consistente;
  - ✓ Confiável
  - ✓ Completa;
  - ✓ Compreensível;
  - ✓ Concisa;
- O registro também é suplementado por **diagramas**, **formulários** padronizados ou **modelos** matemáticos.

ESPECIFICAÇÃO DE REQUISITOS

Notação	Descrição
Sentenças em linguagem natural	Os requisitos são descritos usando sentenças numeradas, em linguagem natural; cada sentença deve expressar um requisito.
Linguagem natural estruturada	Os requisitos são descritos em linguagem natural ou em formulário ou padrão; cada campo provê informações sobre um determinado aspecto do requisito.
Notações gráficas	Modelos gráficos, suplementados com anotações textuais, são usados para definir os requisitos funcionais do sistema. Normalmente são utilizados diagramas de sequência e caso de uso UML (linguagem de modelagem unificada).
Especificações matemáticas	Estas notações são baseadas em conceitos matemáticos, como máquinas de estado finitas ou conjuntos. Apesar destas especificações reduzirem a ambiguidade das especificações, muitos clientes não entendem a especificação formal e não conseguem identificar se representa o que eles querem.

fonte: SOMMERVILLE, 2016, p. 121 (adaptado)

DOCUMENTO DE REQUISITOS DO SISTEMA

- O **documento de requisitos de software** é a declaração oficial do que os desenvolvedores de sistema devem implementar;
- Deve incluir os **requisitos de usuário** de um sistema e uma **especificação detalhada** dos requisitos do sistema;
- Em alguns casos, os requisitos de usuários e de sistema podem estar integrados em uma única descrição.

DOCUMENTO DE REQUISITOS DO SISTEMA

- SOMMERVILLE (2016, p. 127) lista os tipos de usuários de um documento de requisitos:
  - ✓ **Clientes**: especificam e leem os requisitos para verificar se eles atendem às suas necessidades; também especificam mudanças.
  - ✓ **Gerentes**: usam os requisitos para planejar um pedido de proposta para o sistema e planejar o processo de desenvolvimento do sistema.
  - ✓ **Engenheiros de Sistema**: usam os requisitos para compreender qual sistema será desenvolvido.
  - ✓ **Engenheiros de Testes de Sistema**: usam os requisitos para desenvolver testes de validação para o sistema.
  - ✓ **Engenheiros de Manutenção de Sistema**: usam os requisitos para compreender o sistema e os relacionamentos entre suas partes.

DOCUMENTO DE REQUISITOS DO SISTEMA

- A estrutura de um documento de requisitos deve conter:

Capítulo	Descrição
Prefácio	Define os leitores do documento. Contém um histórico de revisões e um sumário das mudanças em cada revisão.
Introdução	Descreve as necessidades do sistema. Deve descrever brevemente as funções do sistema e a interação com outros sistemas. Deve também descrever como o sistema se encaixa nos objetivos e estratégias do negócio.
Glossário	Define os termos técnicos utilizados no documento. Não presuma a experiência ou expertise do leitor.
Definição dos requisitos de usuário	Descreve os serviços fornecidos ao usuário. Os requisitos não-funcionais também devem estar descritos nesta seção. Esta seção deve usar linguagem natural, diagramas, ou outras notações que são compreensíveis para os clientes. Padrões de produto e processo que devem ser seguidos devem ser especificados.

fonte: SOMMERVILLE, 2016, p. 128 (adaptado)

DOCUMENTO DE REQUISITOS DO SISTEMA

Capítulo	Descrição
Arquitetura do sistema	Este capítulo apresenta uma visão geral da arquitetura prevista para o sistema, mostrando a distribuição das funções entre os módulos. Componentes de arquitetura que são reutilizados devem ser destacados.
Especificação dos requisitos de sistema	Descreve os requisitos funcionais e não-funcionais com mais detalhes. Se necessário, pode haver mais detalhamento dos requisitos não-funcionais. Interfaces com outros sistemas devem ser definidas.
Modelos do sistema	Este capítulo inclui modelos gráficos do sistema, incluindo o relacionamento entre os componentes do sistema e o ambiente.
Evolução do sistema	Descreve as premissas no qual o sistema é baseado e quaisquer mudanças que possam ocorrer decorrentes de evolução de hardware, mudança de necessidades do usuário, etc.
Apêndices	Contém informações específicas e detalhadas relacionadas à aplicação que está sendo desenvolvida, como por exemplo descrições de hardware e banco de dados.
Índice	Pode conter vários índices: índice alfabético, lista de figuras, índice de funções, etc.

fonte: SOMMERVILLE, 2016, p. 128 (adaptado)

DEFINIÇÃO DOS REQUISITOS DE USUÁRIO

- Uma boa prática para descrever requisitos de usuário é no formato de **estória de usuário;** uma estória é uma simples 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, e sim ser escrita em linguagem de negócios que é compreensível a todos;
  - ✓ Uma estória deve se concentrar em **quem**, **o quê** e **por quê** de um recurso, **não como**.

ESTÓRIAS DE USUÁRIO

- O formato para especificação é livre, desde que responda a 3 perguntas:
  - ✓ **Quem** se beneficia? (ator): é o usuário/perfil que fará uso.
  - ✓ **O que** se quer? (ação): é a ação em si.
  - ✓ **Para que** serve? (funcionalidade, benefício, motivo): é o resultado obtido ao executar a ação.

ESTÓRIAS DE USUÁRIO

- Estrutura de exemplo para estórias:
  - ✓ Como [**usuário/papel**], quero [**meta**], para que eu possa [**motivo**].
  - ✓ Exemplos:
    - Como atendente de balcão, quero um módulo para realizar o cadastro do cliente, para que eu possa solicitar a entrega da pizza;
    - Como gerente, quero um relatório mensal de pizzas categorizado por sabor, para que eu possa identificar o perfil de consumo dos clientes;
    - Como CEO, quero que o sistema funcione em diferentes plataformas, para que eu possa atingir uma quantidade maior de clientes.

ESTÓRIAS DE USUÁRIO

- Modelo 3Cs – Card, Conversation, Confirmation

  Proposto por Ron Jeffries, em 2001:

  ✓ Cartão: card físico, nota de *post-it*;

  ✓ Conversação: conversa verbal;

  ✓ Confirmação: comportamento, plano de testes.

ESTÓRIAS DE USUÁRIO

- Critério INVEST, proposto por Bill Wake em 2003:

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

ESTÓRIAS DE USUÁRIO

- Análise MoSCoW: define a prioridade da estória 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

ESTÓRIAS DE USUÁRIO

- Especificação como Requisito Funcional:

Requisito	Estória	Prioridade
RF001	Como atendente de balcão, quero um módulo para realizar o cadastro do cliente, para que eu possa solicitar a entrega da pizza	Must
RF002	Como gerente, quero um relatório mensal de pizzas categorizado por sabor, para que eu possa identificar o perfil de consumo dos clientes	Should
RF003	Como CEO, quero que o sistema funcione em diferentes plataformas, para que eu possa atingir uma quantidade maior de clientes	Would

ESPECIFICAÇÃO DETALHADA DOS REQUISITOS FUNCIONAIS

- Segundo SOMMERVILLE (2016, p. 123), um formato padrão usando linguagem estruturada deve conter os seguintes itens:
  - ✓ Descrição da função ou entidade que está sendo especificada;
  - ✓ Descrição das entradas e origens dessas entradas;
  - ✓ Descrição das saídas e destinos dessas saídas;
  - ✓ Descrição da ação a ser tomada;
  - ✓ Informações necessárias para a realização da tarefa ou outras entidades;
  - ✓ Conjunto de pré-condições e pós-condições para completar a ação;
  - ✓ Descrição dos efeitos colaterais da operação.

ESPECIFICAÇÃO DETALHADA DOS REQUISITOS FUNCIONAIS

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	CompDose – a dose de insulina a ser administrada.
Destino	Laço de controle principal.
Ação	CompDose 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 CompDose 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.

fonte: SOMMERVILLE, 2016, p. 124 (adaptado)

ESPECIFICAÇÃO DETALHADA DOS REQUISITOS FUNCIONAIS

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.

ESPECIFICAÇÃO DOS REQUISITOS NÃO-FUNCIONAIS

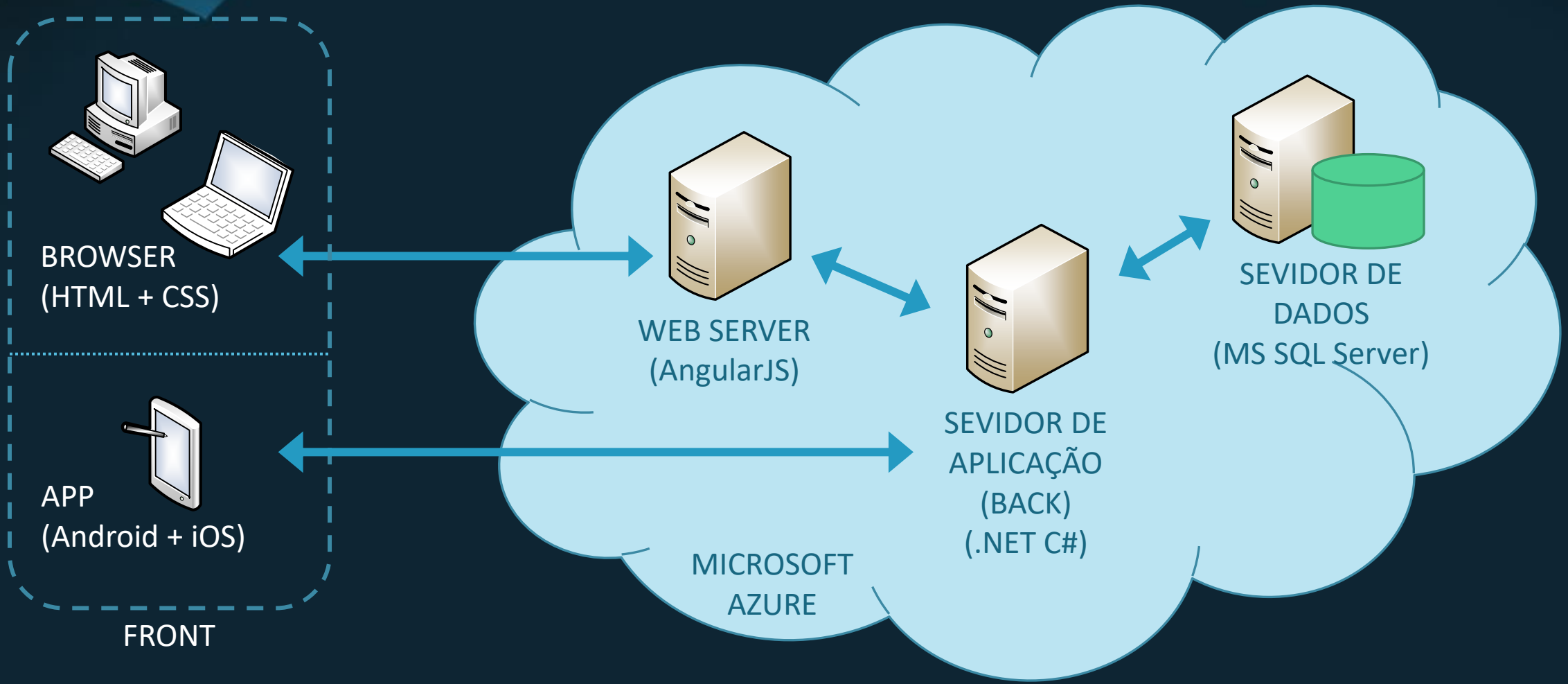
- Uma boa prática ao especificar os requisitos não-funcionais é vinculá-los à(s) característica(s) da norma ISO/IEC 25010 a(s) qual(is) ele atende. Ex.:

Requisito	Descrição	Característica
RnF001	O back end deve ser operado em ambiente Windows	Portabilidade
RnF002	O front end deve ser operado em dispositivos móveis iOS e Android	Portabilidade
RnF003	A base de dados não pode ultrapassar 2GB mensais	Desempenho
RnF004	O software deve ser desenvolvido em C#/.NET	Manutenibilidade
RnF005	A interface (front) não pode fazer uso de cores	Usabilidade
RnF006	A autenticação deve acontecer por biometria	Segurança, Eficiência

PROJETO ARQUITETURAL DO SOFTWARE

- Decisões de projeto: desempenho, segurança, proteção, disponibilidade, manutenibilidade.
- Visões de arquitetura: lógica, física, de desenvolvimento, de processos;
- Padrões de arquitetura: em camadas (MVC, MVVM, ...), repositório, cliente-servidor, canalização, ...;
- Arquiteturas de aplicação: transacionais, de linguagem.

PROJETO ARQUITETURAL DO SOFTWARE



VALIDAÇÃO DE REQUISITOS

- É o processo de verificar se os requisitos definem o sistema que o cliente realmente quer;
- Acontece junto com o levantamento e análise, pois o objetivo é identificar problemas com os requisitos;
- É um processo extremamente importante, pois erros nos requisitos podem levar a retrabalho e aumento de custo.

VALIDAÇÃO DE REQUISITOS

- SOMMERVILLE (2016, p. 129) lista os tipos de verificação que devem ser feitas sobre os requisitos:
  - ✓ **Verificação de validade**: verificam se os requisitos refletem a necessidade real dos usuários, pois os requisitos podem ter mudado depois do levantamento inicial;
  - ✓ **Verificação de consistência**: os requisitos no documento não podem estar em conflito, não devem haver restrições contraditórias;

VALIDAÇÃO DE REQUISITOS

- ✓ **Verificação de completude**: o documento deve conter os requisitos que definem todas as funções e restrições apontadas pelo usuário;
- ✓ **Verificação de realismo**: sabendo das tecnologias existentes, deve-se garantir que os requisitos podem ser implementados com o orçamento disponível e destinado ao projeto;
- ✓ **Verificação de verificabilidade**: para evitar problemas entre cliente e fornecedor, os requisitos devem ser descritos de forma que possam ser verificados se estão atendendo às necessidades.

VALIDAÇÃO DE REQUISITOS

- SOMMERVILLE (2016, p. 130) ainda aponta algumas técnicas para validação de requisitos:
  - ✓ **Revisões de requisitos**: os requisitos são sistematicamente analisados por uma equipe para detectar erros e inconsistências;
  - ✓ **Prototipação**: envolve a construção de um protótipo executável do sistema para avaliar se atende às expectativas do usuário;
  - ✓ **Geração de casos de teste**: os requisitos devem ser testáveis; se há dificuldade na criação de casos de teste, normalmente indica que os requisitos são difíceis de implementar e devem ser reconsiderados.

ATIVIDADE PRÁTICA

- Visando o projeto avaliativo da disciplina:
  - ✓ Descrever entre 10 e 15 requisitos de usuário (funcionais) no formato de estória de usuário [exemplo pág. 41];
  - ✓ Especificar de forma detalhada (requisitos de sistema) pelo menos três requisitos funcionais (uma funcionalidade *core* + um cadastro simples + um relatório) [exemplo pág. 44];
  - ✓ Listar entre 5 e 10 requisitos não-funcionais, classificando-os sob a norma ISO/IEC 25010 [exemplo pág. 45];
  - ✓ Produzir diagrama simples da arquitetura do sistema [exemplo pág. 47].

REFERÊNCIAS

- PRESSMAN, R. W, MAXIM B. R. ***Software Engineering - A Practitioner's Approach***. 8th Ed. New York: McGraw-Hill, 2015.
- SOMMERVILLE, I. ***Software Engineering***. 10th Global ed. Harlow: Pearson, 2016.