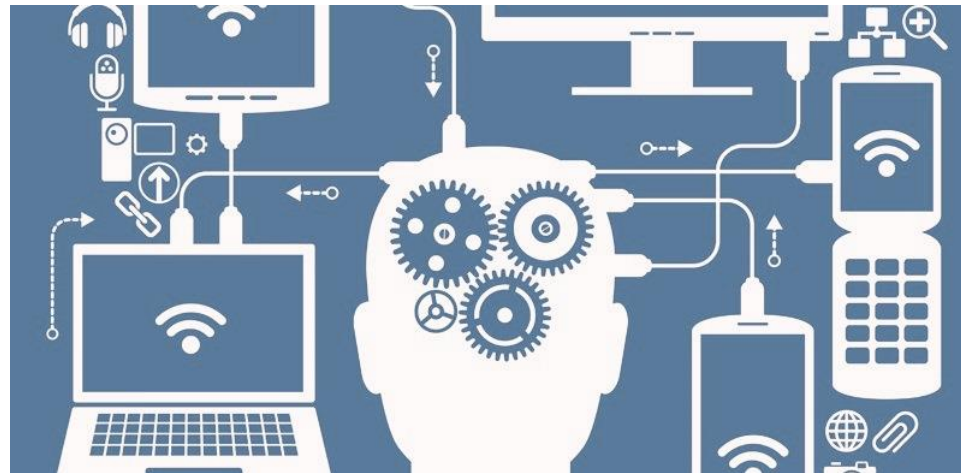


Engenharia de Software 2



Prof. Esp. João Paulo S. Araújo

Requisitos de Software

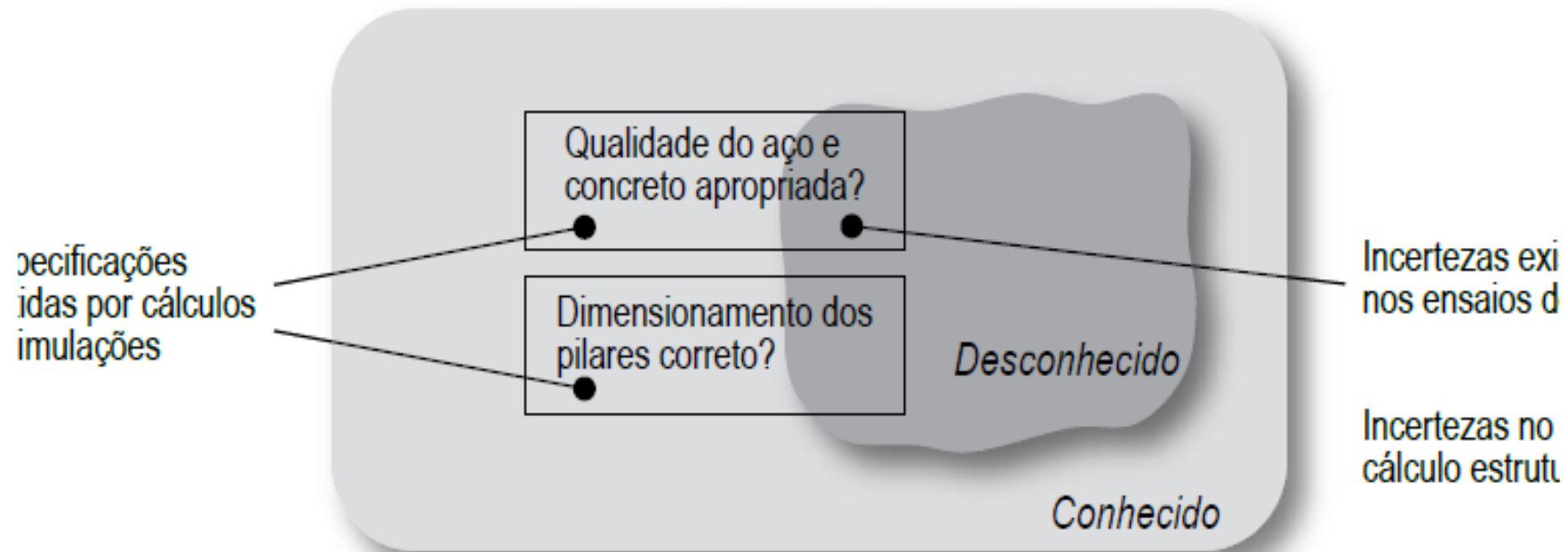
Revisão

Desafios do Desenvolvimento de Software

- *Aspecto não repetitivo torna atividade mais difícil;*
- Imprevisibilidade;
- Dificuldade inicial para delimitar o escopo
 - *Volatilidade de **requisitos***

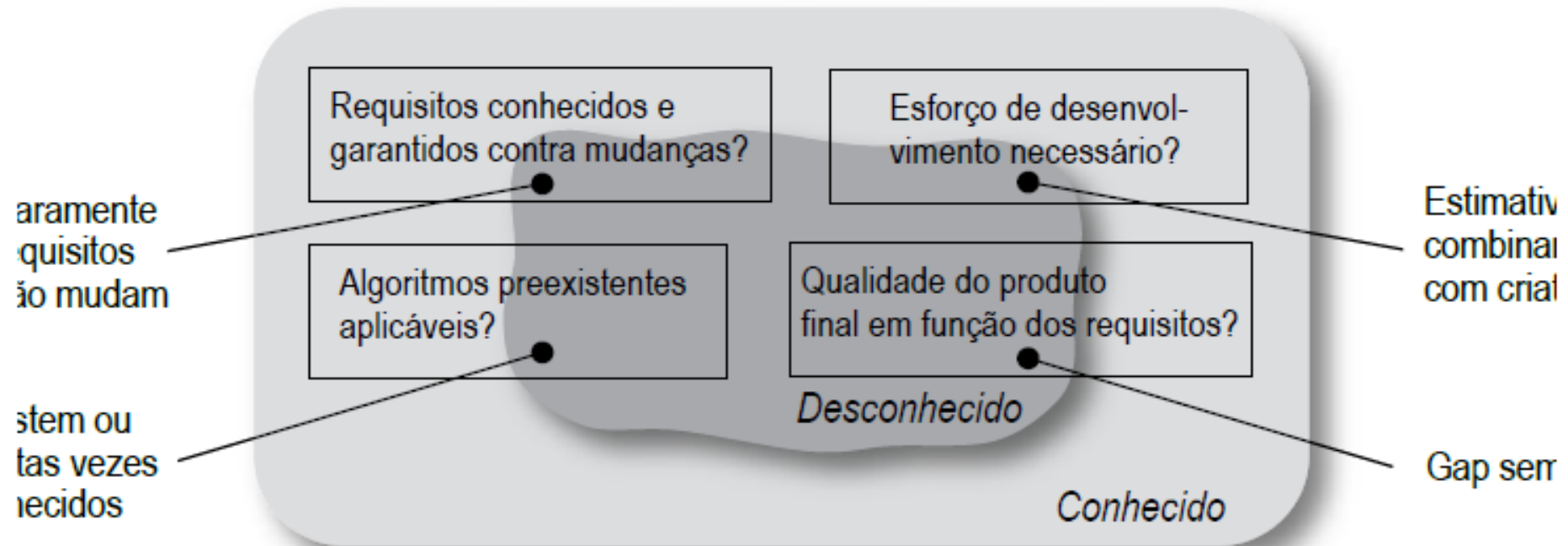
Imprevisibilidades e incertezas em projeto de engenharia civil (*zonas de sombra*)

Projeto e realização de uma ponte



Imprevisibilidades e incertezas em projeto de software (*zonas de sombra*)

Projeto e implementação de software



Mas onde estão os **problemas/defeitos**???

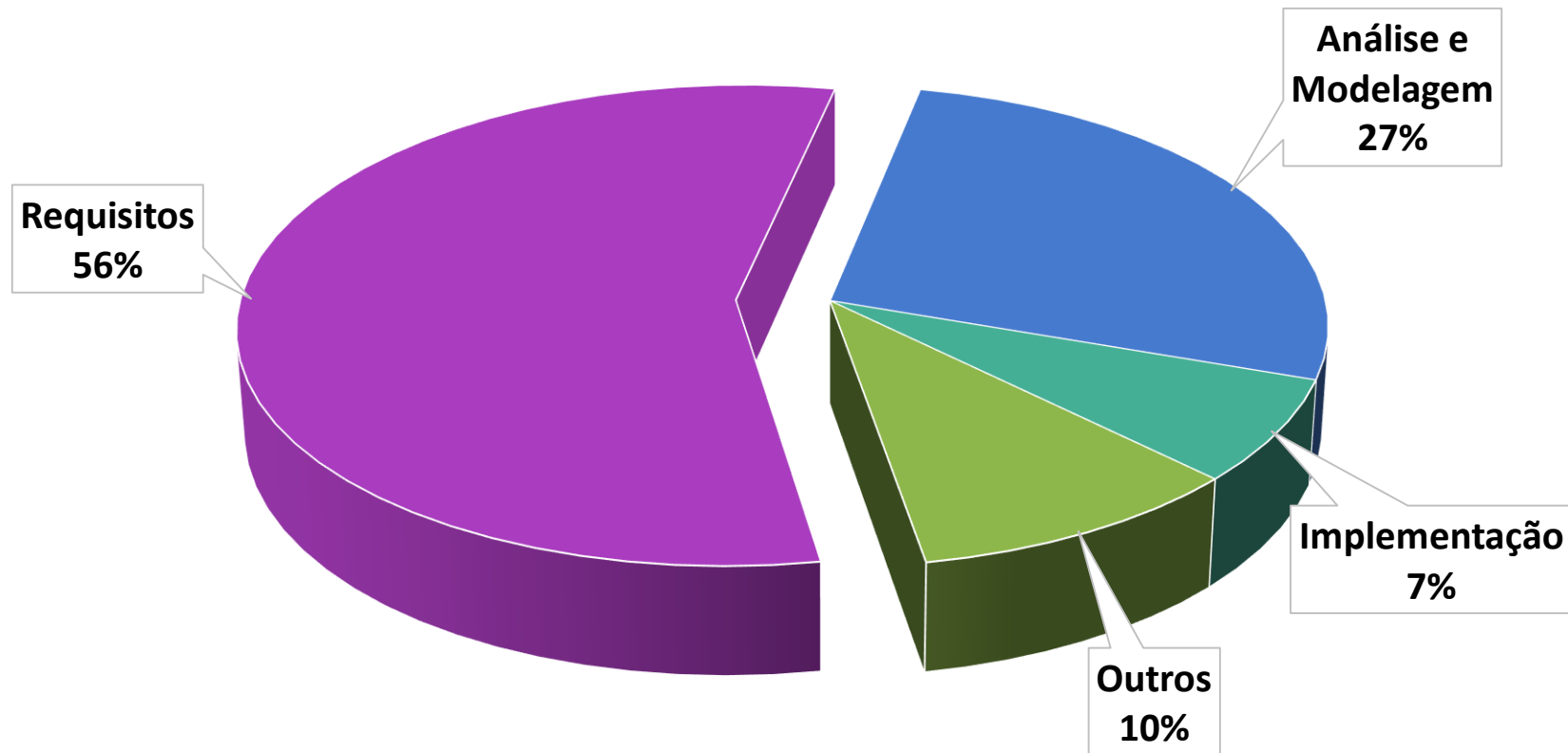
*Há uma visão comum de entender que os **problemas** e **defeitos** de software estão apenas no **código-fonte** ou ainda que somente os profissionais de **desenvolvimento** e **testes** são os responsáveis por um software sem defeitos;*

.

Mas onde estão os defeitos e problemas ???

- ✓ **Problemas e defeitos** ocorrem em todas as fases do processo de desenvolvimento de software;
- ✓ Estudos demonstram que a **maior incidência de erros** está nas *fases iniciais* do processo;
 - ✓ *Especificação e/ou entendimento mal compreendidos sobre os objetivos;*

MAS ONDE ESTÃO OS PROBLEMAS ???



Engenharia de Software

"A Engenharia de Software tem por objetivo apoiar o desenvolvimento profissional de software, mais do que a programação individual."

Lembrando que software corresponde a:

- *Aplicativo (programa em si)*
- *Toda a documentação associada*
- *Dados de configurações necessários para execução*

(SOMMERVILLE, 2011)

Desafios da Engenharia de Software

- Lidar com o aumento da diversidade, com as demandas pela diminuição do tempo de entrega e desenvolvimento de software confiável.
- Conciliar disciplina p/ garantir melhor previsibilidade de resultados c/ caráter aleatório da criação de soluções;
- Desenvolvimento de **metodologias** (pedagógicas), **modelos**, **tecnologias**, **ferramentas** (*automatizar, ganhar tempo, ter uniformidade*)

Engenharia de Software – Guia SWEBOK

- Guia SWEBOK (*Software Engineering Body of Knowledge / Corpo de Conhecimento de Engenharia de Software*) pelo IEEE;
 - Dividido em áreas de conhecimento:
 - *Gerência de engenharia, projeto, métodos e ferramentas de engenharia, construção, processo de engenharia, testes, qualidade, manutenção, disciplinas relacionadas e gerência de configuração;*
 - Áreas trabalhando em conjunto;

Se pensarmos que...

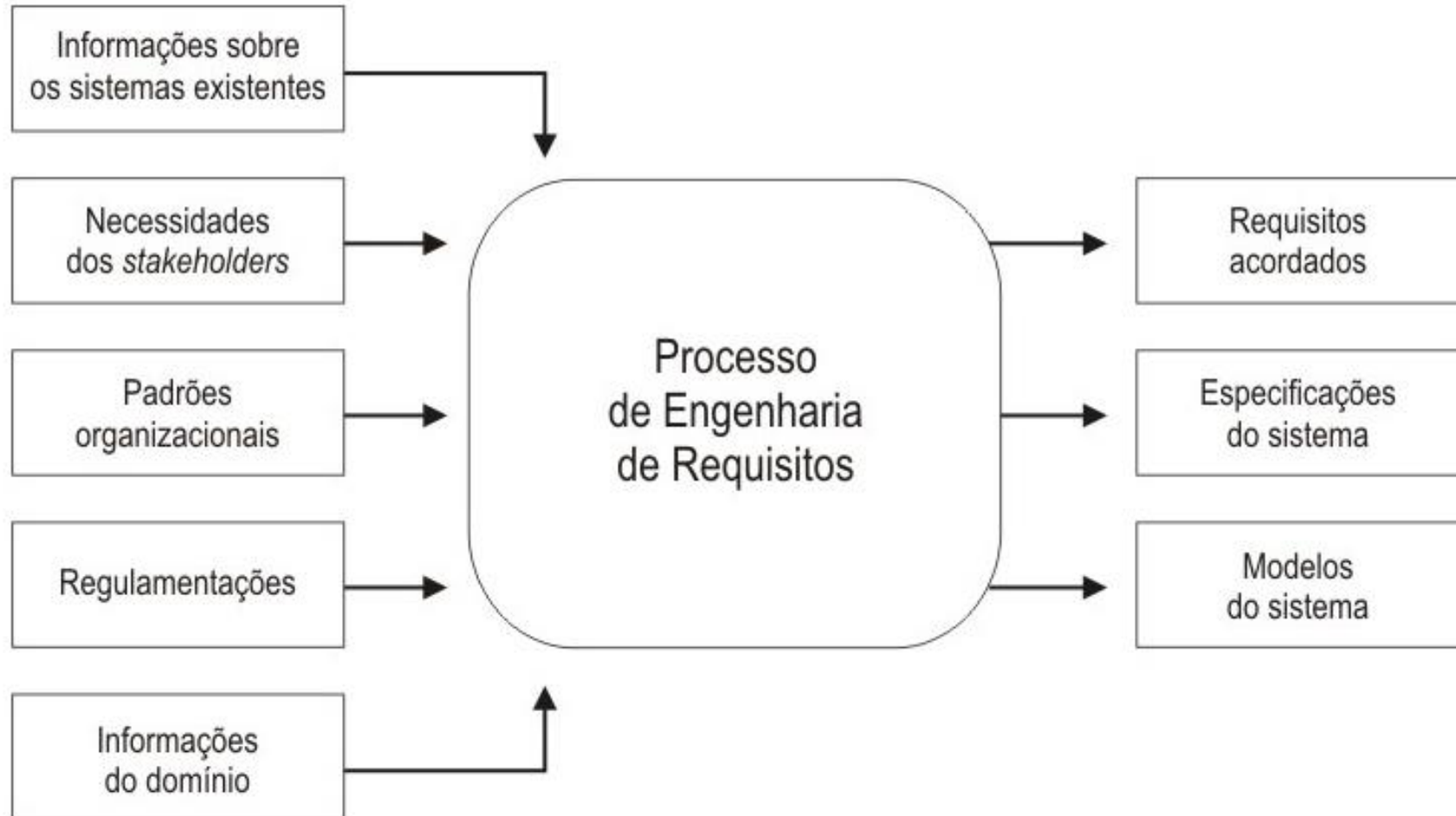
As principais causas de falhas em projetos são relativas aos requisitos...

Falhas se devem as dificuldades em entender o que o usuário quer, descrições incompletas e mudanças não controladas nos requisitos...

Podemos considerar que:

Realizar corretamente o levantamento e a administração de requisitos é *vital* para a *qualidade do software*.

Engenharia de Requisitos



...mas o que são
requisitos de
software mesmo?



Requisitos de Software

- São as descrições sobre o seu comportamento, funções e especificações das operações que deve realizar e especificações sobre suas propriedades ou atributos;
- Compreendem as funcionalidades presentes no software quando este estiver pronto para ser executado;
- Área de estudo de requisitos de software está relacionada com o levantamento, análise, especificação, rastreamento e validação de requisitos;

Requisitos de Software

- Em outras palavras, os **requisitos** definem em princípio **o que o software deve fazer** e também **como as operações serão realizadas**;
- **Dados e operações** do software estão interligados e encadeados entre si;

Tipos de requisitos

Requisitos Funcionais

- Descrevem as funcionalidades e serviços que o sistema deve oferecer;
- Especificação de como deve ser o comportamento do sistema em determinadas situações;
- Deve determinar o que se espera que o software faça, sem se preocupar como ele faz;

Requisitos Funcionais

Exemplos

- O software deve possibilitar o cálculo dos gastos diários e semanais com pessoal;
- O software deve emitir relatórios de pedidos na tela de pesquisa;

Tipos de requisitos

Requisitos Não-Funcionais

- Descrevem restrições aos serviços ou funções oferecidas pelo sistema de forma geral;
 - *Restrições de tempo, restrições do processo de desenvolvimento, padrões, etc;*
 - *Aconselhável usar métricas p/ cada requisito;*
- Qualidades globais de um software:
 - *Usabilidade, desempenho, custos, etc;*

Requisitos NÃO-Funcionais (RNFs)

Desempenho	Interface	Interoperabilidade
Verificação	Eficiência	Robustez
Portabilidade	Qualidade	Confiabilidade
Recursos	Segurança	Manutenabilidade

(Kotonya e SommerVille, 2000)

Requisitos NÃO-Funcionais (RNFs)

Exemplos

- O tempo de resposta do sistema deve ser inferior a 30 segundos;
- O tempo de desenvolvimento não deve ultrapassar 6 meses;

Ex.: Se ao se pedir um saldo num terminal bancário este demorar 10 minutos, o sistema será um fracasso...

REQUISITOS DE SOFTWARE - *Pontos de vista*

- **Cliente:** Descreve como ele realiza o trabalho e como o software deve lhe auxiliar;
- **Analista/Desenvolvedor:** Questiona como o software deve funcionar internamente;
- **Gerentes de Projeto:** Define preocupações com os recursos, limitações de cronograma e orçamento, etc.;
- **Stakeholders:** Pessoas ou organizações que detêm algum controle sobre a especificação do software;

Documento de Requisitos – *Principais problemas*

- Dificuldades de compreensão do leitor;
- Dificuldades de escrita por parte do redator;
- Falta de organização
- Inconsistência;
- Ambiguidade;

Tipos de Especificação de Requisitos

- Requisitos de usuário:
 - Escritos para o cliente em linguagem natural;
 - Documento com linguagem de alto nível;
 - Pode ser usados diagramas simples, tabelas, listas, etc.
 - Pode ser usado como um contrato com o cliente.
- Requisitos de sistema:
 - Documento com descrição detalhada dos serviços do sistema;
 - Descrição detalhada do software que serve como base para o projeto, implementação e arquitetura do sistema;
 - Em alguns casos também pode ser usado como um contrato entre cliente e contratado;

Documento de Requisitos

- Não existe padrão único para o documento;
- É recomendável que cada organização elabore um formato que seja seguido por todas as equipes e pessoas ligadas ao desenvolvimento de software;
- Uso de linguagem natural
 - *Termos técnicos somente para técnicos*

Documento de Requisitos – *Recomendações*

Normalmente um documento contém:

- Serviços e funcionalidades que o software deve ter (*Requisitos Funcionais*);
- As restrições de operação (*Requisitos não funcionais*);
- Regras específicas ao negócio (*Pode incluir especificações matemáticas*)
- Notações gráficas (*Diagramas UML por exemplo*)
- Propriedades gerais do software;
- Requisitos de hardware;
- Definição de quais outros softwares devem estar integrados

Documento de Requisitos — *Recomendações de escrita*

➤ Dedicção/Alocação de tempo

➤ Consistência

- *Interpretação única ;*
- *Evitar sinônimos desnecessários, expressões de duplo sentido;*

➤ Concisão

- ✓ Evitar uso de adjetivos;
- ✓ Usar frases ou parágrafos curtos;
- ✓ Destacar partes fundamentais
- ✓ Usar expressões mais específicas e quantificáveis;

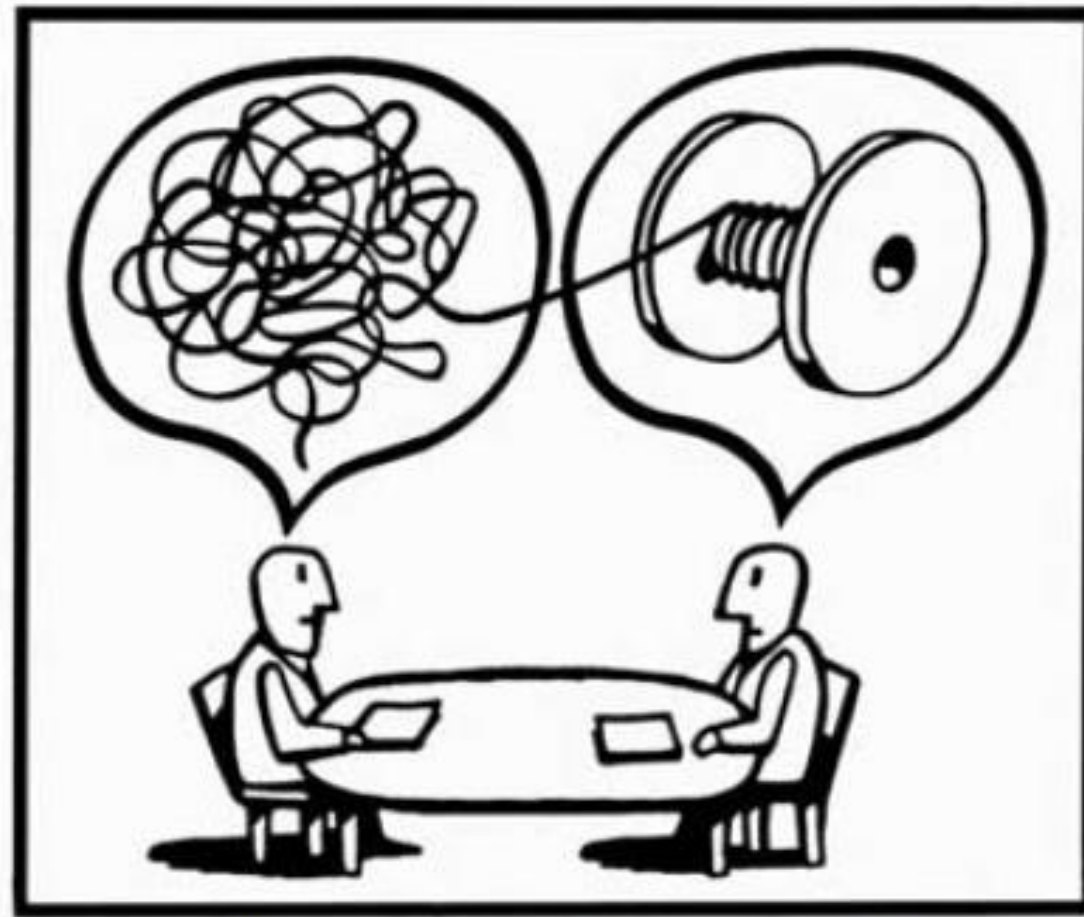
Ex.: Trocar "O sistema deve ser rápido"

por "O sistema deve processar um n° X de requisições por hora";

Requisitos de Software na Abordagem Ágil

- A abordagem ágil consiste em simultaneamente descrever os requisitos enquanto o sistema é desenvolvido;
 - *Histórias de Usuário / Backlog itens;*
- Normalmente permitem maior acompanhamento dos requisitos, porém muitas vezes pode ser difícil de ser aplicado em alguns sistemas de grande porte.
- Requer dedicação e engajamento da equipe e das partes interessadas (*stakeholders*)

Diálogo entre cliente e Analista/Desenvolvedor



Cliente

Engenheiro de
Software

Diálogo entre cliente e Analista / desenvolvedor

- *"O cliente não sabe o que quer."*
- *"Não pedi porque é óbvio."*
- *"Basta incluir dois campos a mais no formulário."*

Diálogo entre cliente e Eng. de Soft. / desenvolvedor

"É só montar uma telinha aí..."

"Coloca um botãozinho..."

"É só fazer um ifzinho aí..."

Algumas questões p/ refletir...

- Quanto custam estas mudanças!?
- Qual o impacto!?
- Como é o controle e gerenciamento destas “pequenas” alterações!?
- Será que o número de alterações não daria um outro projeto ou módulo?

Planejando o Gerenciamento de requisitos

- Identificação dos requisitos
- Processo de gerenciamento de mudanças
 - *Avaliação de impacto e custo de mudanças*
- *Políticas de rastreabilidade*
 - *Definição do relacionamento entre os requisitos e o projeto*
 - *Definição de como manter os registros*
- *Ferramentas de Apoio*
 - *Envolvem desde textos e planilhas até sistemas especializados*
 - *Armazenamento em repositório seguro, gerenciamento de mudanças e rastreabilidade*

Gerenciamento de Mudanças de Requisitos

