

Gestão de Qualidade de Software

Unidade 5 – Princípios de Engenharia de Requisitos



Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP
aparecidovfreitas@gmail.com
CTFL – CTFL AT – CPRE

Prof. Aparecido V. de Freitas

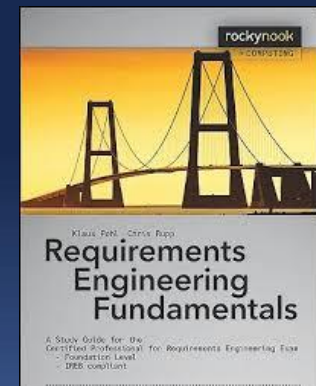
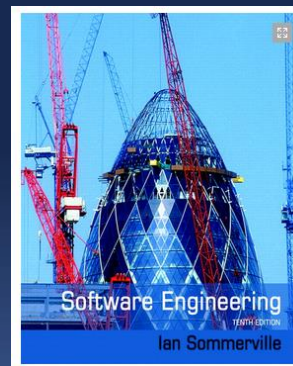
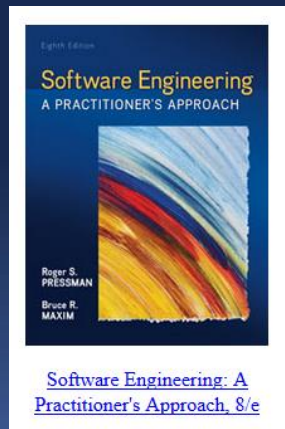


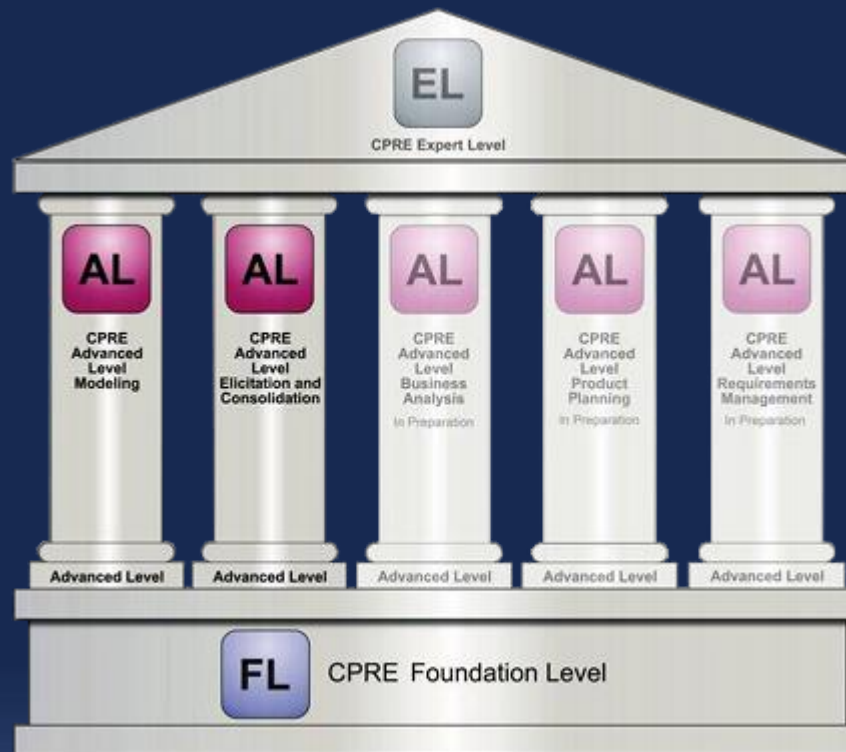
- ◆ **Doutor** em Engenharia da Computação pela **EPUSP** – Escola Politécnica da **USP**;
- ◆ **Mestre** em Engenharia da Computação pela **EPUSP** – Escola Politécnica da **USP**;
- ◆ **Engenharia** Plena pela Escola de Engenharia **Mauá**;
- ◆ Bacharel em **Matemática** pela **Fundação Santo André**;
- ◆ Atuou durante 15 anos como Analista e Supervisor de Suporte Técnico na área de **TI** da **Volkswagen** do Brasil;
- ◆ Especialista na plataforma **IBM i** (desde 1993);
- ◆ Experiência na plataforma **IBM Mainframe** (desde 1980);
- ◆ Autor do livro “**Fundamentos do Sistema IBM i**”, Editora Makron Books;
- ◆ Professor da **USCS** dos cursos da Escola da Computação;
- ◆ Professor do curso de **Engenharia de Computação** da Escola de Engenharia **Mauá**;
- ◆ Ex-Professor da **Fatec** e da **UMESP** – Universidade Metodista de São Paulo;
- ◆ Ex-Gestor dos cursos de Computação da **USCS** há 13 anos;
- ◆ Sócio – Diretor da Qualitsys Consultoria de TI (Treinamento em plataforma Mainframe);
- ◆ **Certificação Internacional IREB - Certified Professional for Requirements Engineering (CPRE)**;
- ◆ Certificação Internacional **ISTQB** - Certified Testing Foundation Level (**CTFL**);
- ◆ Certificação internacional em Testes Ágeis de Software – **ISTQB – CTFL-AT**
- ◆ Consultor **CAIPIMES** – Elaboração de provas de Concursos da área de TI, desde 1990.



Bibliografia

- **Fundamentos de Engenharia de Requisitos – POHL K., RUPP C. – IREB 2012**
- **Software Engineering – A Practitioner's Approach – Roger S. Pressman – Eight Edition – 2014**
- **Software Engineering – Ian Sommerville – 10th edition - 2015**
- Engenharia de Software – Uma abordagem profissional – Roger Pressman - McGraw Hill, Sétima Edição - 2011
- Engenharia de Software – Ian Sommerville – Nona Edição – Addison Wesley, 2007
- Requirements Engineering Fundamentals – IREB Compliant – Klaus Pohl e Chris Rupp – CPRE-FL, 1996.





CPRE-FL (Certified Professional Requirements Engineer – Foundation Level)



Certificação em Engenharia de Requisitos

- **IREB – International Requirements Engineering Board** - entidade criada em 2006 com a visão de criar uma base internacionalmente aceita de profissionalização da disciplina Engenharia de Requisitos;
- **ABRAMTI** – Associação Brasileira de Melhoria em TI;
- **IBTQS** – Instituto Brasileiro de Qualidade em Testes de Software;
- Certificação CPRE – Certified Professional for Requirements Engineering;
 - ✓ Nível Fundamental
 - ✓ Nível Avançado
 - ✓ Nível Especialista





Certificação em Engenharia de Requisitos – CPRE

- Sem **data de expiração**;
- Validade **Internacional**;
- Quatro provas por ano em locais definidos pela ABRAMTI em todo o território nacional;
- Não há **pré-requisitos** para a realização da prova;
- **Não** há necessidade de comprovar experiência na área de Engenharia de Requisitos;





Certificação em Engenharia de Requisitos – CPRE

- A prova CPRE é composta por **45** questões com duração de **75** minutos;
- Sem consulta;
- A pontuação das questões é variável dependendo do grau de importância do assunto;
- Questões assinaladas incorretamente invalidam acertos de outras questões;
- Há questões com mais de uma alternativa correta;
- O enunciado da questão informa quantas respostas certas devem ser assinaladas;
- A aprovação é concedida para os candidatos com pelo menos **60%** de acertos das questões.





Competências do Engenheiro de Requisitos

- Estar ciente das razões que podem levar a uma Engenharia de Requisitos inapropriada;
- Compreender os diversos tipos de Requisitos de Software;
- Compreender a importância dos Requisitos de Qualidade;
- Estar ciente da importância da Comunicação da Engenharia de Requisitos;
- Ter proficiência nas atividades principais da Engenharia de Requisitos;
- Conhecer as responsabilidades e competências do Engenheiro de Requisitos.





Em que situações um projeto de software é bem-sucedido ?



Em que situações um projeto de software é bem-sucedido ?



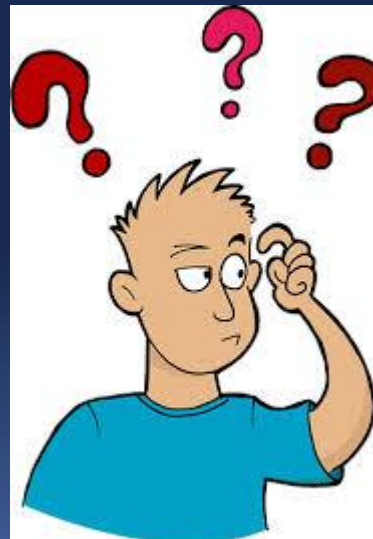
Em que situações um projeto de software é bem-sucedido ?

- ✓ Prazos foram cumpridos !
- ✓ Custos próximos aos planejados !
- ✓ Software agregou valor ao negócio !
- ✓ Especificações de qualidade foram atendidas !
- ✓ Funcionalidades foram atendidas !
- ✓ Processo de definição de requisitos foi adequado !





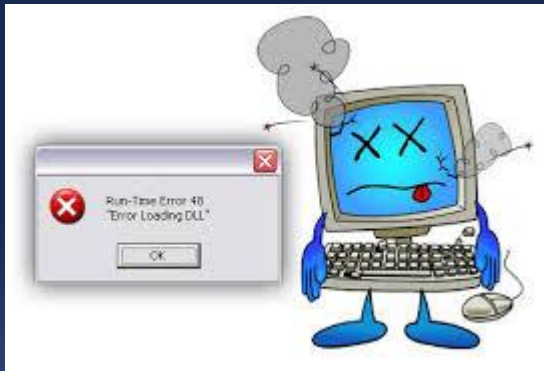
Será que todos os projetos de Software são assim ?



Será que todos os projetos de Software são assim ?

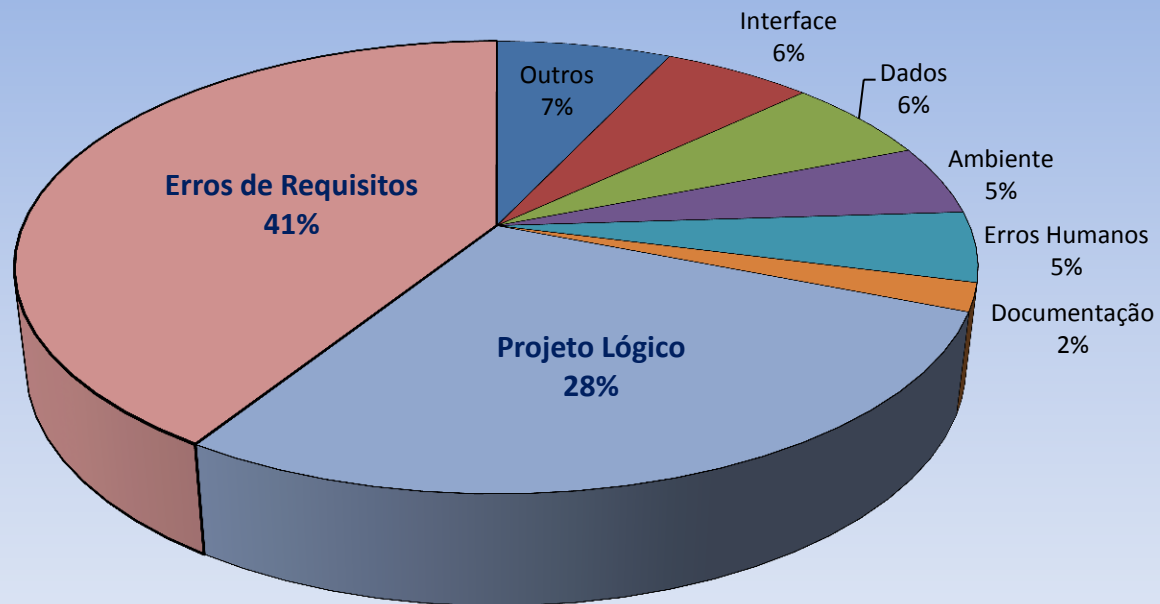


- Estudos mostram que **40% a 60%** dos problemas ocorridos em projetos de software, são decorrentes de **falhas no processo de requisitos**.



Aurum A. ; Wohlin C. – Engineering and Managing Software Requirements

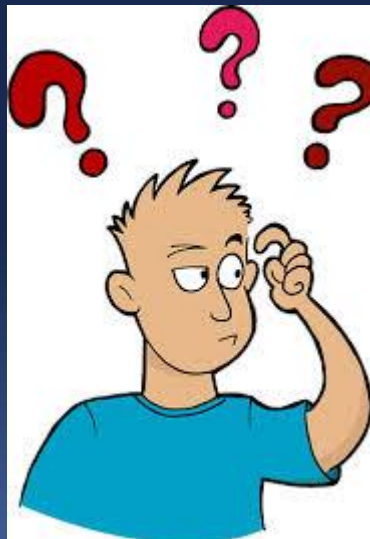
Erros de Projetos



Fonte: **Projeto US Air Force**



Quais as consequências de um processo de requisitos mal elaborado ?



Quais as consequências de um processo de requisitos mal elaborado ?

- Falha da definição de requisitos acarreta **defeitos** de software;
- Requisitos são fundamentais para a obtenção de Software de **qualidade**;
- O Software entregue não irá refletir as **necessidades** reais do stakeholder;
- Alto impacto nos **custos** e **prazos** do projeto;
- Software com **baixa QUALIDADE**.



Como desenvolver um bom processo de Requisitos ?



Como desenvolver um bom processo de Requisitos ?

- Empregar técnicas de elicitação e análise de Requisitos;
- Conhecer o negócio do stakeholder;
- Levantar adequadamente as necessidades do stakeholder;
- Definir requisitos em conjunto com stakeholder (e não **isoladamente**).



Qual o impacto dos requisitos no custo de um Software ?



Impacto dos requisitos no custo de um Software

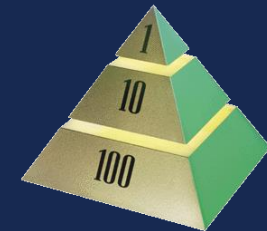
- Um erro de requisito acarreta:
- Custo 2.5 vezes mais no projeto;
- Custo 5 vezes mais na codificação;
- Custo 10 vezes mais no Teste unitário;
- Custo 25 vezes mais no Teste de aceitação;
- Custo 100 vezes mais na Manutenção

Propagação dos custos



O alto custo dos Requisitos Errados

A regra do 1-10-100



“Os resultados mostram como corrigir erros encontrados nos requisitos custa até 200 vezes menos do que em estágios mais avançados do ciclo de vida”

Fonte: Boehm 1988

Ok, mas então o que é um Requisito ?



Requisito

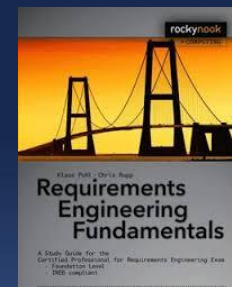
- De forma bem abrangente, requisito é uma definição clara, sem ambiguidade e precisa das funcionalidades que o sistema é capaz de realizar, juntamente com restrições, para que os objetivos do negócio do stakeholder sejam atendidos.
- Uma condição ou capacidade que deve ser alcançada ou estar presente em um sistema ou componente de sistema para satisfazer um contrato, norma, especificação ou outro documento formalmente imposto.

Norma IEEE 610.12-1990 – Página 62



Engenharia de Requisitos

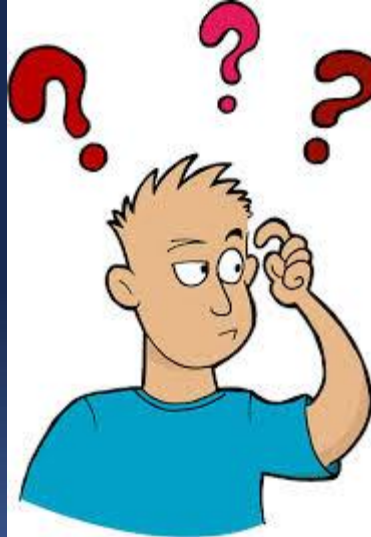
- ✓ Processo de Engenharia que cuida do **levantamento, análise, documentação e validação** de **Requisitos**.
- ✓ A Engenharia de Requisitos é uma abordagem **sistemática** e disciplinada para a especificação e gerenciamento de requisitos.



Fonte: Pohl, 1996



Quais as principais fontes de obtenção dos Requisitos ?



Stakeholders

- São as **mais importantes fontes** para obtenção de requisitos;
- São as **pessoas** ou organizações que têm impacto sobre os requisitos;
- Isso inclui as pessoas que irão interagir com o sistema (por exemplo, usuários e administradores);
- Inclui também pessoas que não irão usar o sistema, mas que têm interesse no sistema (por exemplo, alta gerência ou um cracker do qual o sistema precisa ser protegido...);
- Inclui também **entidades legais**, instituições, **normas**, documentação de sistemas **legados**, etc



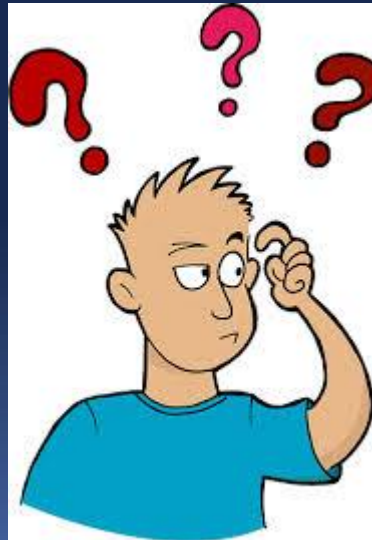
Definição – Stakeholders

- Um **stakeholder** de um sistema é uma pessoa ou uma organização que tem influência (direta ou indireta) nos requisitos de um sistema;
- Qualquer pessoa ou organização que seja parte interessada no projeto.



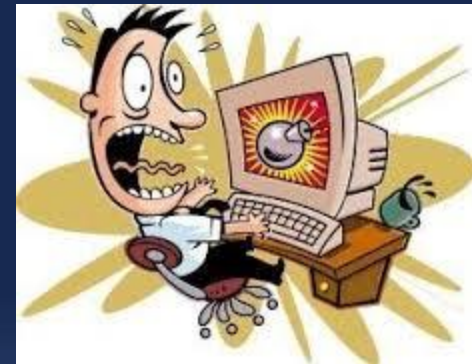
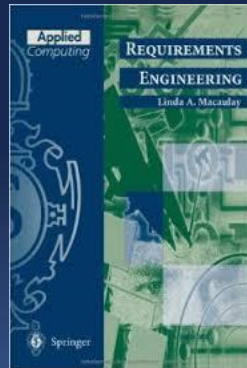


Qual o impacto no desenvolvimento do software quando não se leva em conta todos os Stakeholders ?



Impacto da consideração dos Stakeholders

- Sem considerá-los, pode-se obter uma elicitação fragmentada de requisitos;
- Isso resulta em requisitos **incompletos**.



Fonte: Macaulay, Requirements Engineering, 1996

Quais os objetivos da Engenharia de Requisitos ?



Objetivos da Engenharia de Requisitos

- Durante o processo de desenvolvimento, a **Engenharia de Requisitos** assegura que os requisitos são corretamente **elicitados** das partes interessadas (**stakeholders**), que são adequadamente documentados, validados e verificados.
- Garante também que esses requisitos são gerenciados durante todo o ciclo de vida do software em desenvolvimento.



Quais as atividades centrais da Engenharia de Requisitos ?





Atividades centrais da Engenharia de Requisitos

- **Concepção** (identificação da necessidade do negócio)
- **Elicitação** (levantamento de requisitos)
- **Análise** (modelagem de requisitos)
- **Negociação** (Tratamento de conflitos e definição de prioridades)
- **Especificação** (Documento textual formal ou modelagem)
- **Validação** (Eliminação de ambiguidade, inconsistências e revisão)
- **Gestão** (Controle de mudanças de requisitos)

Qual o perfil de um Engenheiro de Requisitos ?

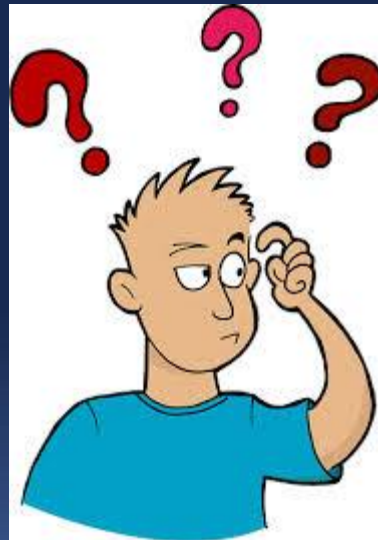


O Engenheiro de Requisitos

- **Raciocínio Analítico.** Capacidade de familiarizar-se com domínios que lhe sejam desconhecidos.
- **Empatia.** Boa intuição e capacidade de agir como se fosse um Usuário.
- **Competência Comunicativa.** Saber ouvir, fazer perguntas certas na hora correta. Boa capacidade de comunicação.
- **Resolução de Conflitos.** Capacidade de identificar conflitos.
- **Moderação.** Deve ser um moderador nato.
- **Auto-confiança:** Capacidade de se defender em caso de fortes objeções às suas opiniões, jamais levando tais comentários como uma crítica pessoal.
- **Persuasão:** Capacidade de sustentar e defender requisitos dos Usuários em reuniões de equipe e apresentações.



Como a Engenharia de Requisitos é incorporada nos modelos de Processo de Software ?



Engenharia de Requisitos no Modelo Cascata



- No modelo Cascata (Waterfall) [Royce,1987], os requisitos são totalmente elicitados antes do desenvolvimento propriamente dito.
- Nesses processos, a ER é compreendida como uma fase inicial do desenvolvimento de sistemas, finita e restrita no tempo.

Engenharia de Requisitos no Modelo Evolucionário



- Nos modelos evolucionários elicit-se, inicialmente, somente os requisitos necessários para a implementação, visto que é difícil “prever” funcionalidades futuras;
- Nesse modelo, os requisitos mudam ao longo do projeto e, portanto, a ER é vista como um processo contínuo e abrangente que engloba e integra todas as fases do desenvolvimento do software;
- No início de cada iteração, novos requisitos são identificados e priorizados.

Importância da Comunicação

- Requisitos precisam ser comunicados;
- Na maioria dos casos, utiliza-se a linguagem natural como forma de comunicação;
- Emissor e receptor usam um código comum intrínseco (por exemplo, a língua portuguesa);
- Quanto mais afinidades existirem entre emissor e receptor, melhor será a troca de informações;
- No entanto, na maioria das vezes, tais condições ideais não existem entre stakeholders;
- Entre o emissor e o receptor, pode haver ruídos na mensagem;
- Isso é amenizado por meio de um glossário, nos quais os termos importantes são explicados;
- Alternativamente, pode-se adotar uma linguagem descritiva formal, por exemplo, a Linguagem de Modelagem Unificada (UML).



Influência do Meio

- Na comunicação verbal, emprega-se redundância (linguagem e gestos, entonação, feedback);
- Na comunicação escrita, pelo contrário, as informações são transmitidas com um mínimo de redundância e feedback.



Ambiguidade

- Muitas vezes, as informações não são adequadamente transmitidas, ou nem mesmo transmitidas;
- O autor pode pressupor que o leitor tenha algum tipo de conhecimento implícito anterior sobre o tema;
- Essas simplificações podem causar problemas em termos de Requisitos, pois os Requisitos podem tornar-se suscetíveis a diversas interpretações (ambiguidade).

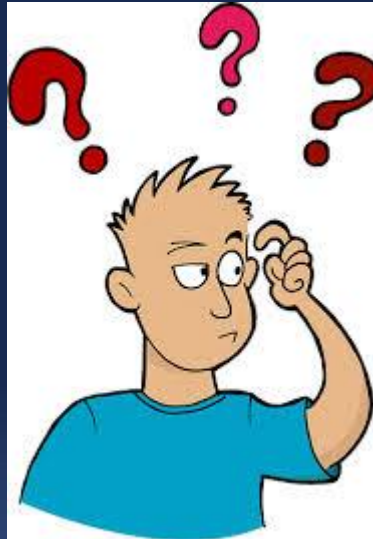


Ambiguidade

- O velho senhor encontrou o rapaz em seu escritório !
- Visitamos o teatro do povoado que foi fundado em 1980.
- O rapaz viu o mendigo sentado na varanda.
- O policial deteve o bandido que roubou o banco na Av. Brasil.
- O vendedor disse ao cliente que seu preço estava incorreto.



Quais os tipos de Requisitos ?



Tipos de Requisitos

- Requisitos funcionais
- Requisitos Não Funcionais ou de Qualidade
- Restrições



Requisitos Funcionais

- Requisitos funcionais definem a funcionalidade oferecida pelo sistema a ser desenvolvido.
- Um **requisito funcional** é um requisito relacionado ao resultado de algum comportamento a ser fornecido por uma função do sistema.



Requisitos Não Funcionais

- Também conhecidos por Requisitos de Qualidade;
- Definem qualidades desejadas do sistema a ser desenvolvido;
- Muitas vezes influenciam a arquitetura do sistema mais que os requisitos funcionais;
- Tipicamente requisitos de qualidade definem o desempenho, a disponibilidade, a confiabilidade, a escalabilidade ou a portabilidade de um sistema.



Requisitos de Qualidade

- Geralmente requisitos de qualidade costumam ser especificados por meio de linguagem natural;
- Devem ser quantificáveis. Por exemplo, o sistema deverá processar 95% de todas as requisições dentro de 1 segundo. O processamento de todas as requisições não deve ultrapassar 3 segundos em qualquer instante;
- Norma **ISO 9126** auxilia na especificação e categorização dos requisitos de qualidade.



Restrições

- Não podem ser influenciadas pelos membros da equipe;
- Exemplo: O sistema deverá estar disponível no mercado no mais tardar no quarto trimestre de 2016;
- Ao contrário dos requisitos funcionais e de qualidade, as restrições não são implementadas, mas devem ser cumpridas, pois meramente limitam o espaço da solução disponível durante o processo de desenvolvimento.

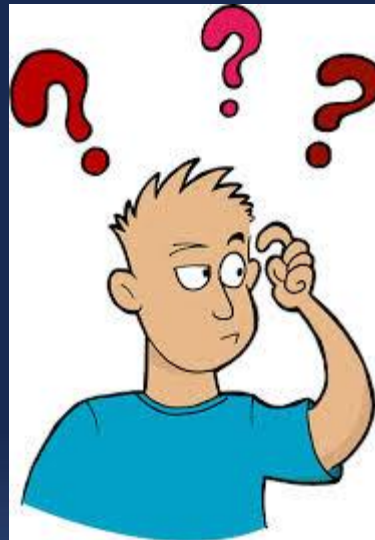


Os limites do Sistema e do Contexto

- Na Engenharia de Requisitos, o propósito de definir os **limites do sistema** e do **contexto** é identificar a parte do ambiente que influencia os requisitos do sistema a ser desenvolvido.



O que é contexto do sistema ?



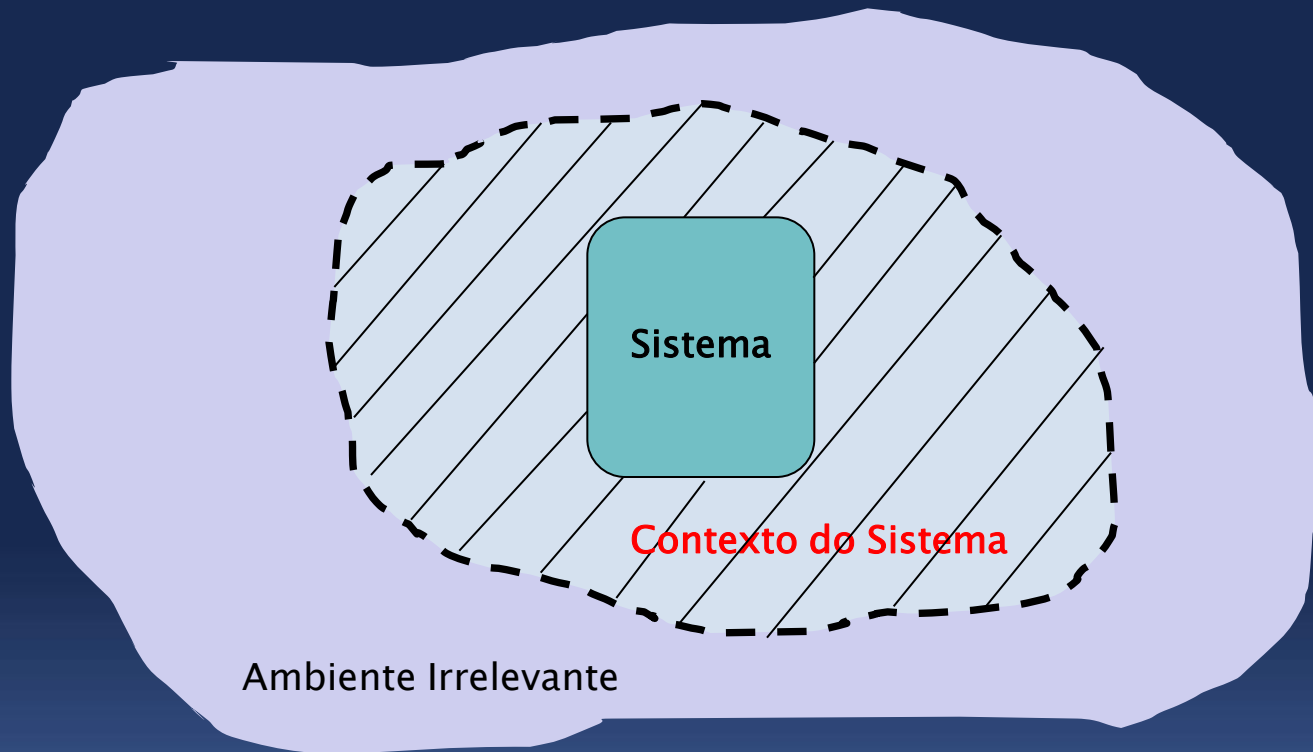
Contexto do Sistema

- É a **parte do ambiente** do sistema que é **relevante** para a definição e a compreensão dos requisitos de um sistema a ser desenvolvido;
- Os seguintes aspectos da realidade influenciam o contexto de um sistema:
 - ⊗ Pessoas (usuários ou grupos de usuários)
 - ⊗ Sistemas em operação
 - ⊗ Processos
 - ⊗ Eventos



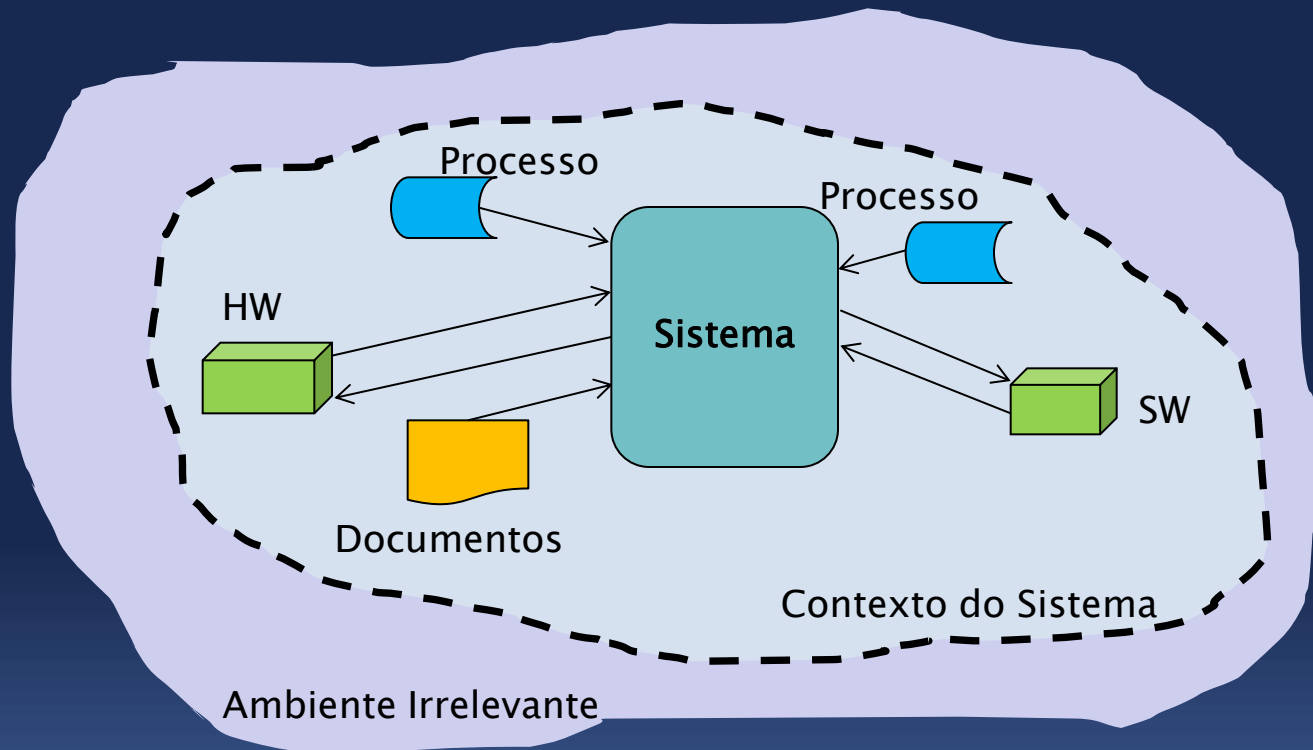
Contexto do Sistema

- É responsabilidade do Engenheiro de Requisitos definir adequadamente o contexto do sistema.

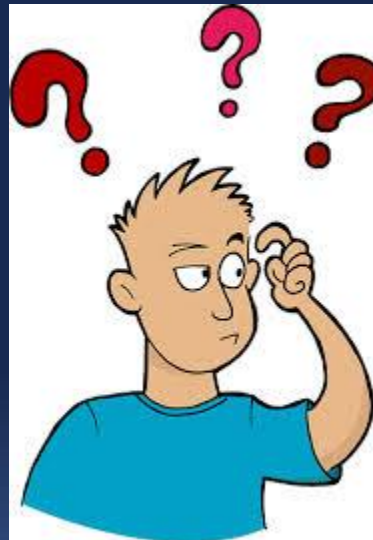


Contexto do Sistema

- Pode ser constituído por outros **sistemas**, por grupos de **usuários** que de uma forma ou outra utilizam a interface do sistema a ser desenvolvido, além de fontes adicionais de requisitos e suas interrelações.

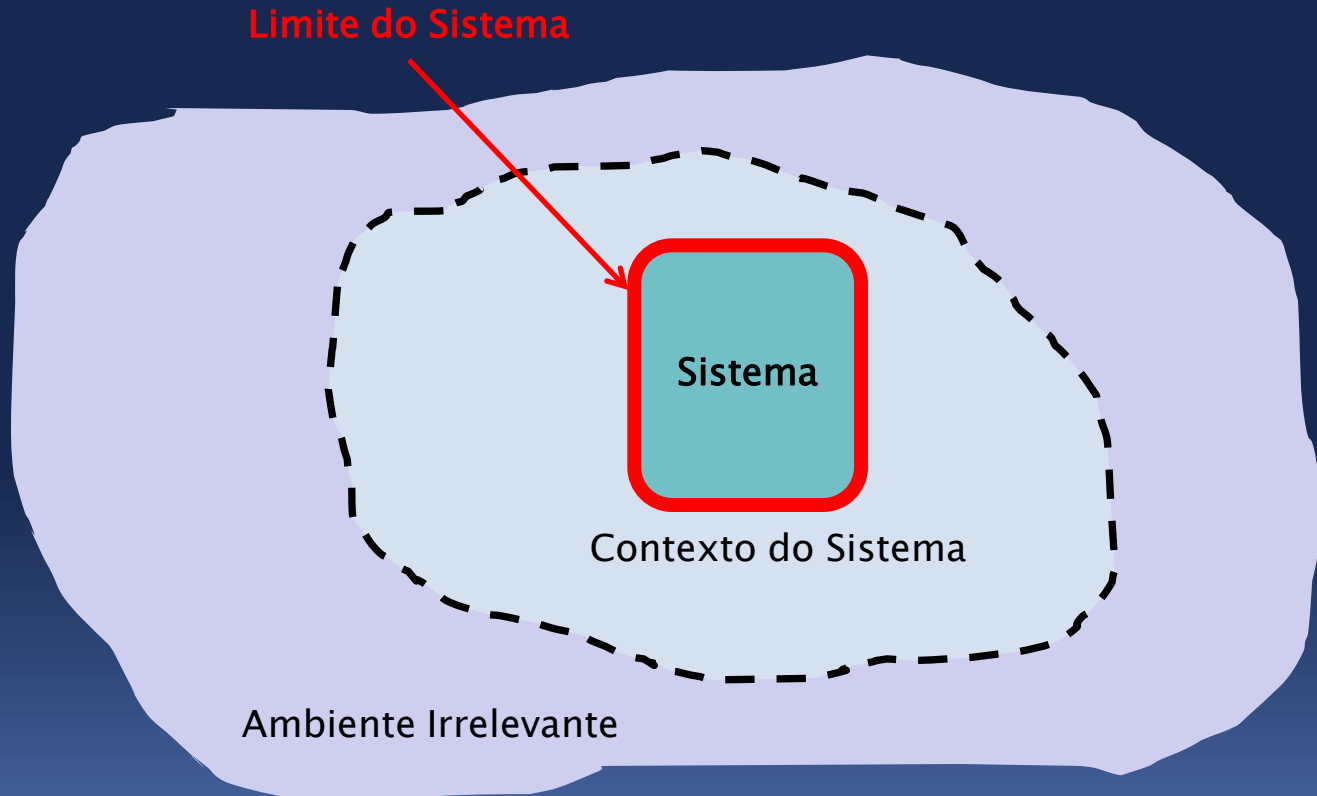


O que é limite do sistema ?



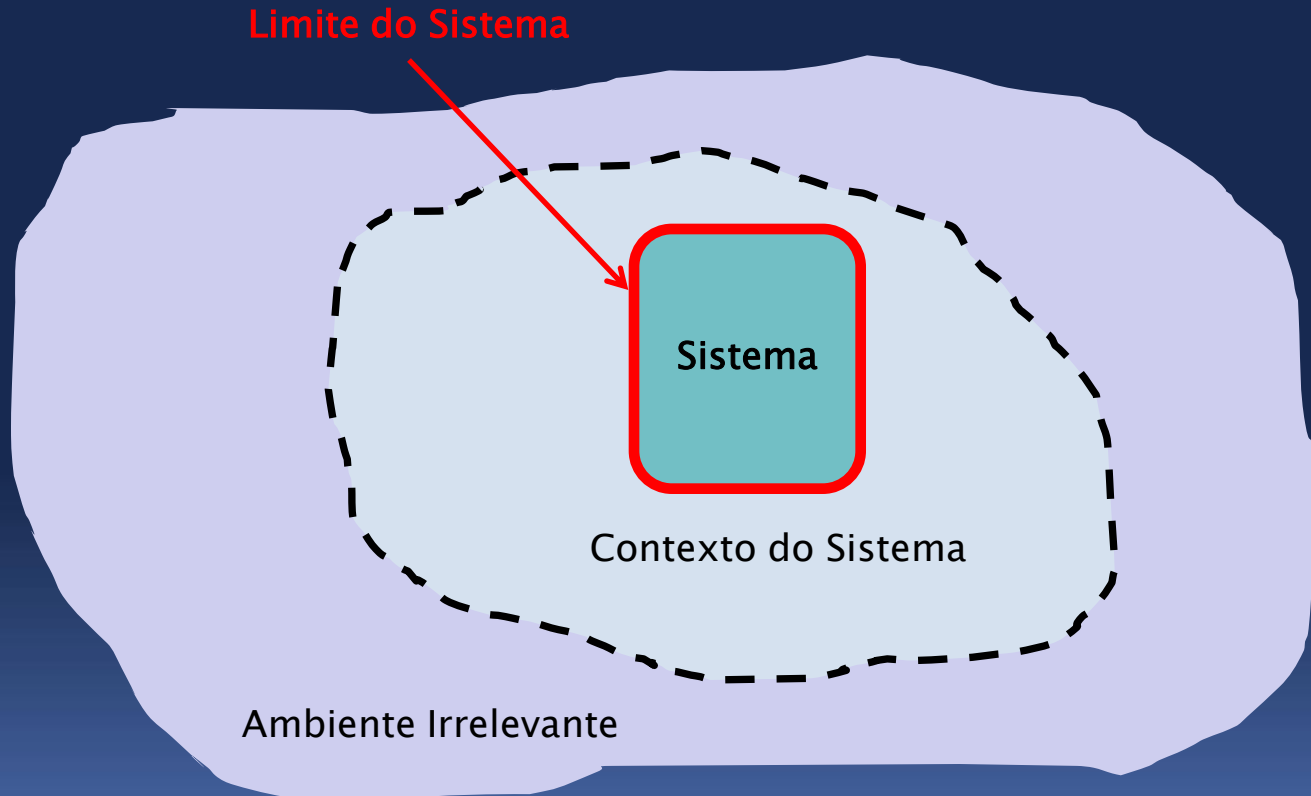
Limite do Sistema

- O **limite do sistema** separa o sistema a ser desenvolvido do seu ambiente (isto é, separa a parte da realidade que pode ser modificada ou alterada pelo processo de desenvolvimento) daqueles aspectos do ambiente que não podem ser mudados ou modificados pelo processo de desenvolvimento.



Limite do Sistema

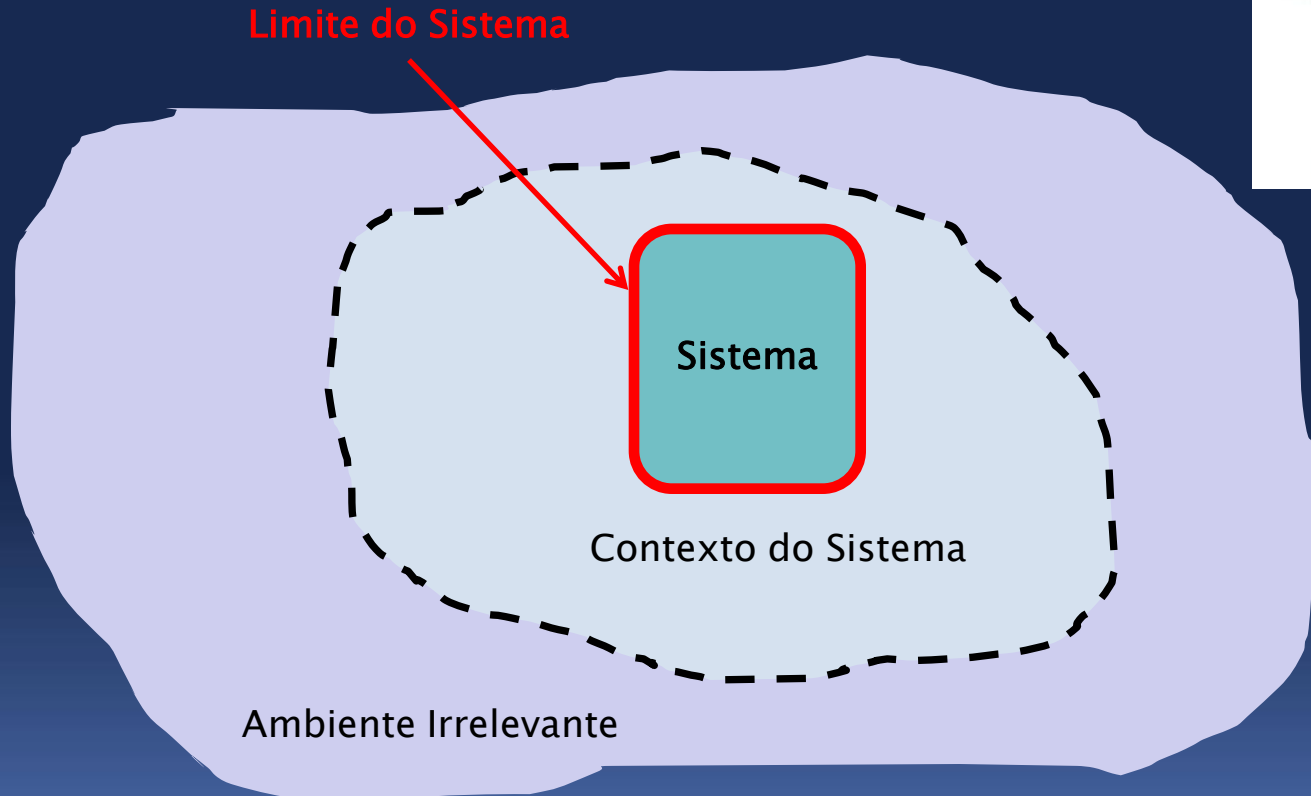
- Todos os aspectos que se encontram dentro do **limite do sistema** podem ser alterados durante o desenvolvimento do sistema;
- Com a definição do **Limite do Sistema**, define-se também o **Escopo do Sistema** (**abrangência do Sistema, fecha-se o escopo do sistema**).



Limite do Sistema

- Para se definir o limite do sistema deve-se responder a seguinte questão:

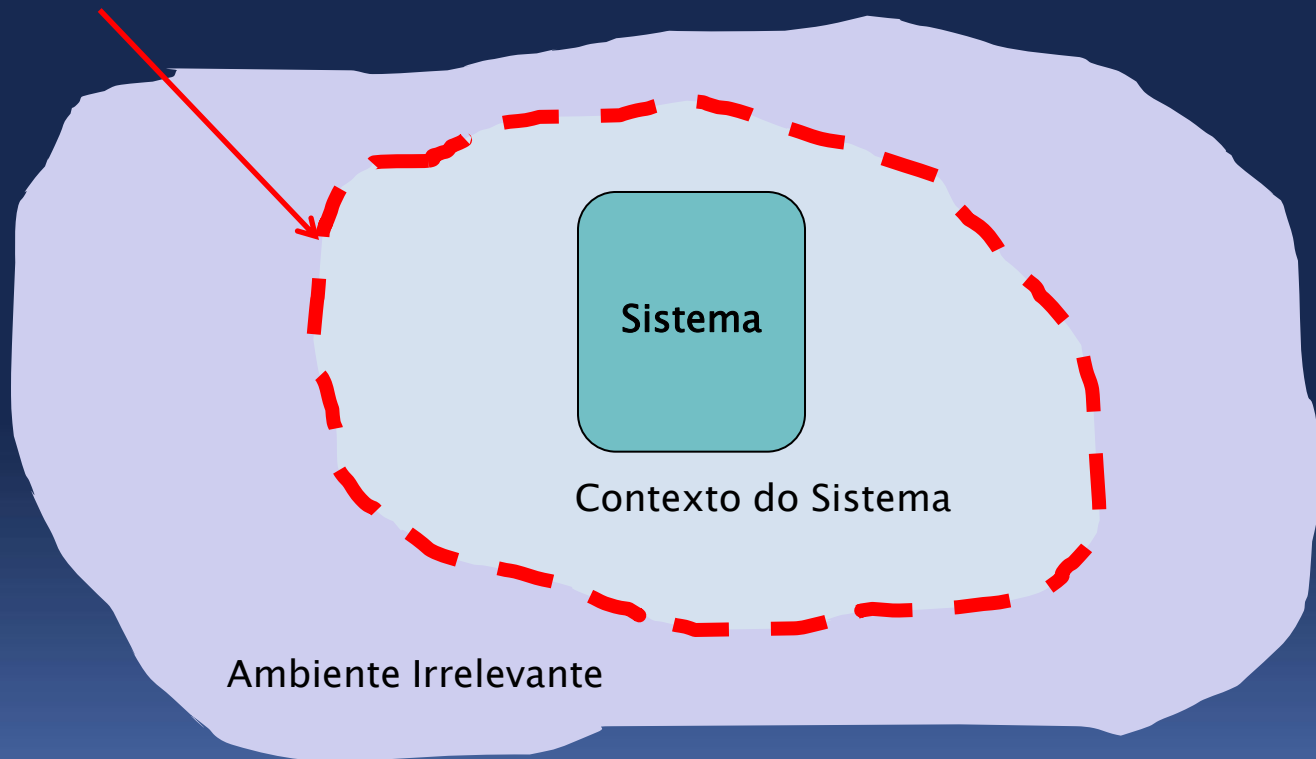
Quais aspectos pertencem ao sistema que será desenvolvido ?



Limite do Contexto

- O limite do contexto separa a parte relevante do ambiente de um sistema a ser desenvolvido da parte irrelevante, isto é, a parte que não influencia o sistema a ser desenvolvido e, sendo assim, não precisa ser considerado durante a Engenharia de Requisitos.

Limite do Contexto



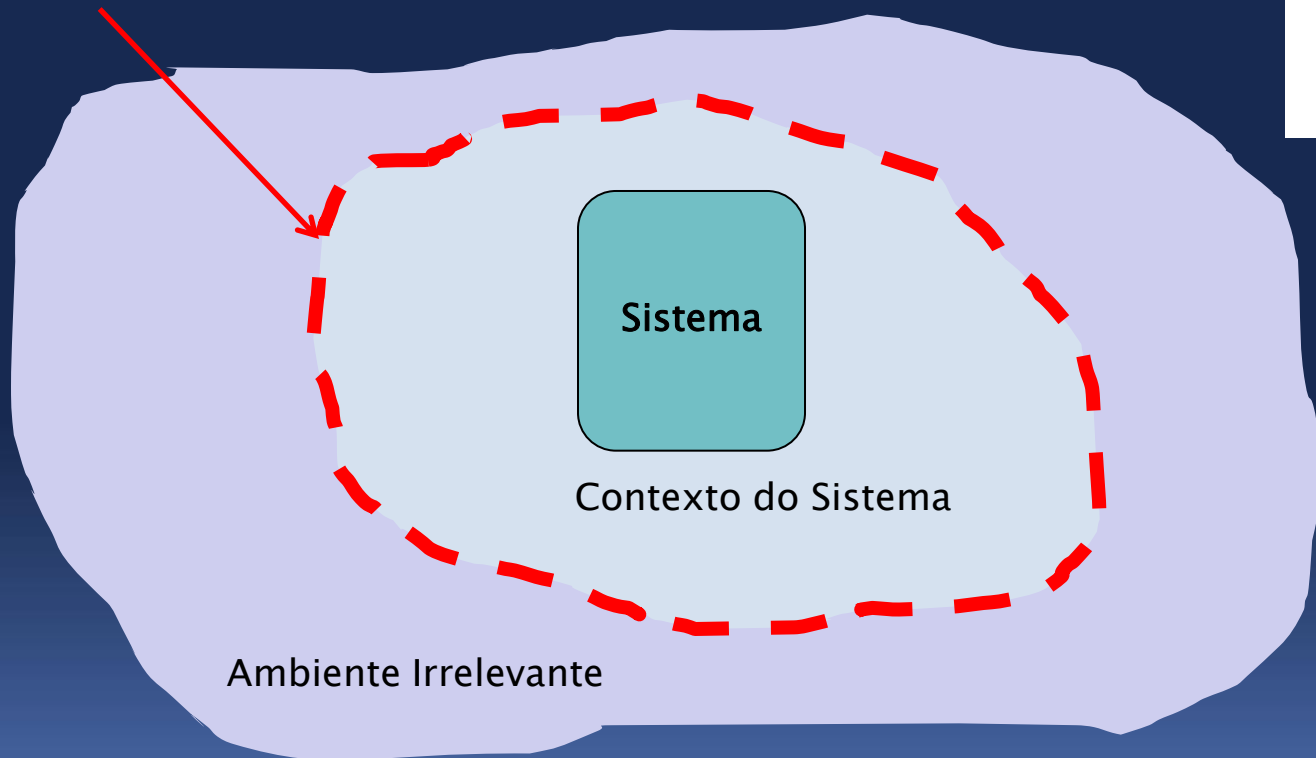
Limite do Contexto

- Para se definir o Limite do Contexto, deve-se responder a seguinte questão:

Quais aspectos pertencem ao contexto do sistema e quais são irrelevantes?

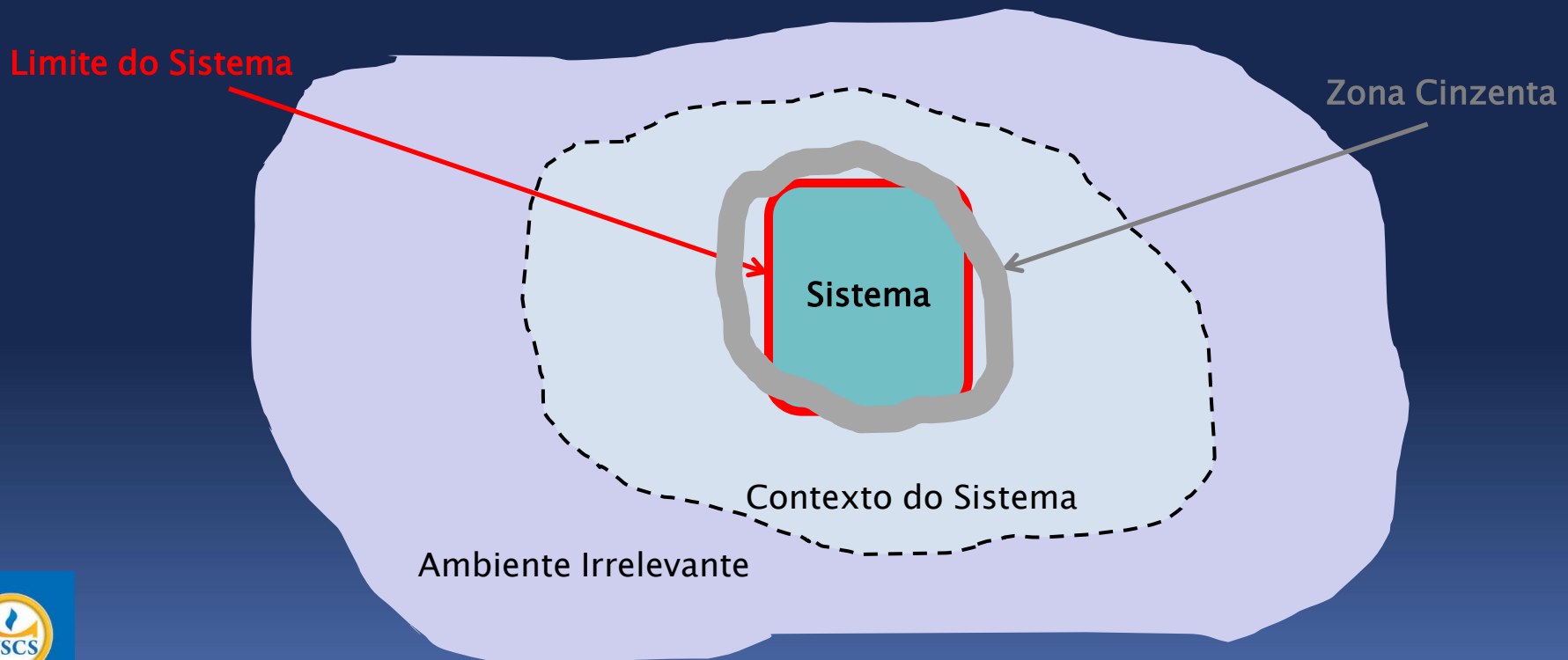


Limite do Contexto



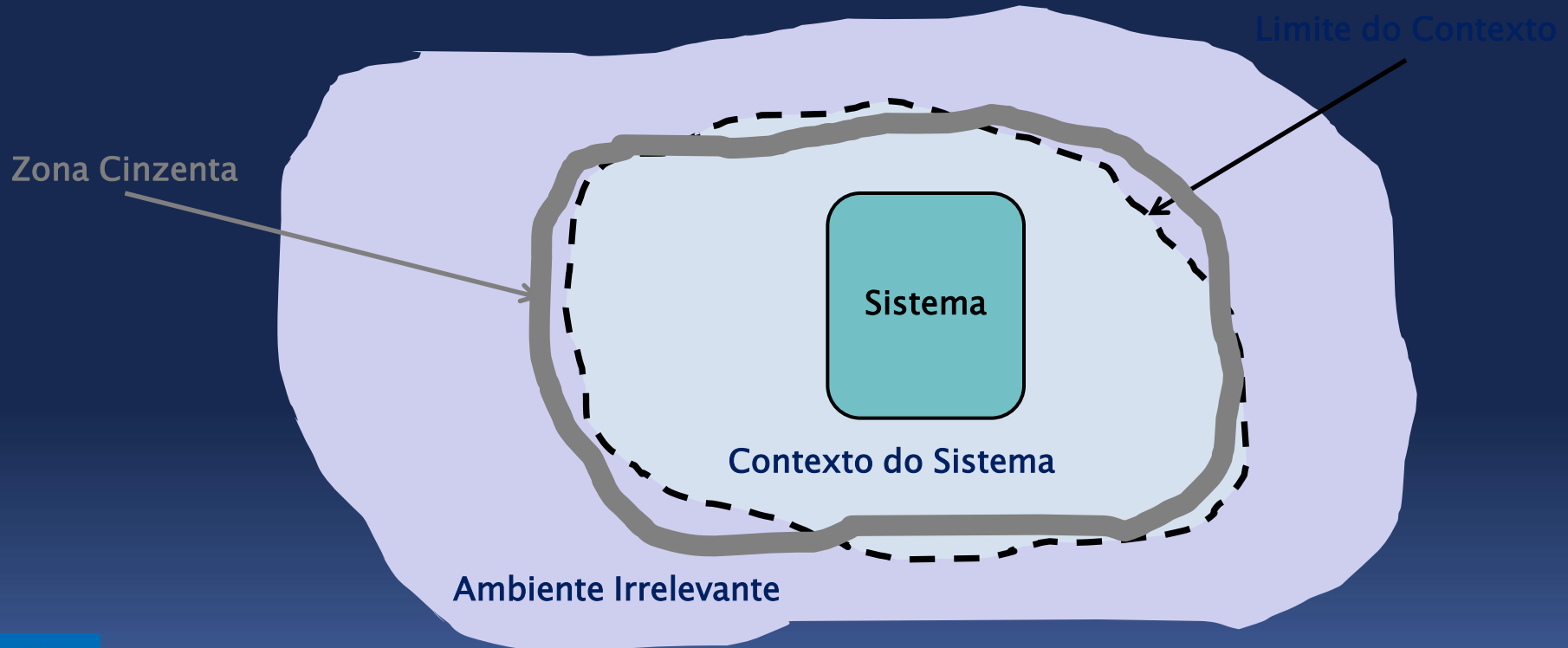
Zona Cinzenta entre o Sistema e o Contexto do Sistema

- Corresponde à separação inicialmente vaga (indefinida) do sistema e seu contexto;
- O limite do Sistema geralmente não é bem definido até o final do processo de Engenharia de Requisitos;
- Um bom processo de Engenharia de Requisitos deve resolver (ou eliminar) a zona Cinzenta (eliminar dúvidas e fechar o escopo do software).



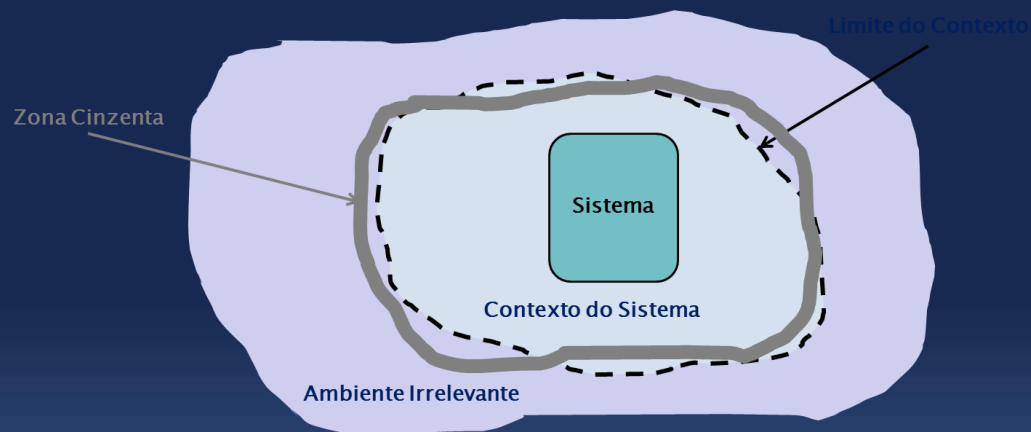
Zona Cinzenta entre o Contexto e o ambiente irrelevante do Sistema

- Uma delimitação completa e precisa do Limite do Contexto é praticamente **impossível**;
- **Nem sempre é possível** visualizar aspectos do ambiente que influenciam ou não o sistema.

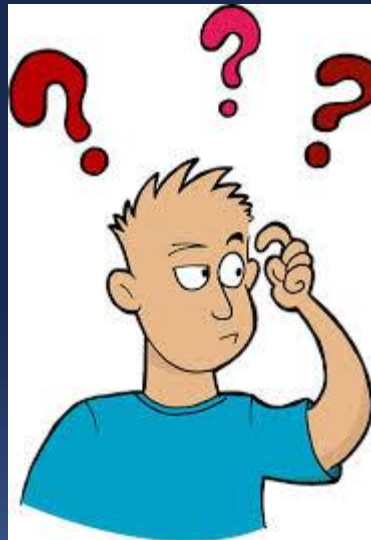


Zona Cinzenta entre o Contexto e o ambiente irrelevante do Sistema

- Diferentemente da zona cinzenta entre o Sistema e o Contexto, a zona cinzenta entre o Contexto e o Ambiente Irrelevante **não** precisa ser necessariamente resolvida.



Como documentar o Contexto de um sistema ?



Documentação

- Diagramas de Casos de Uso (UML)
- Diagramas de Contexto (Análise Estruturada)

Limite do Contexto

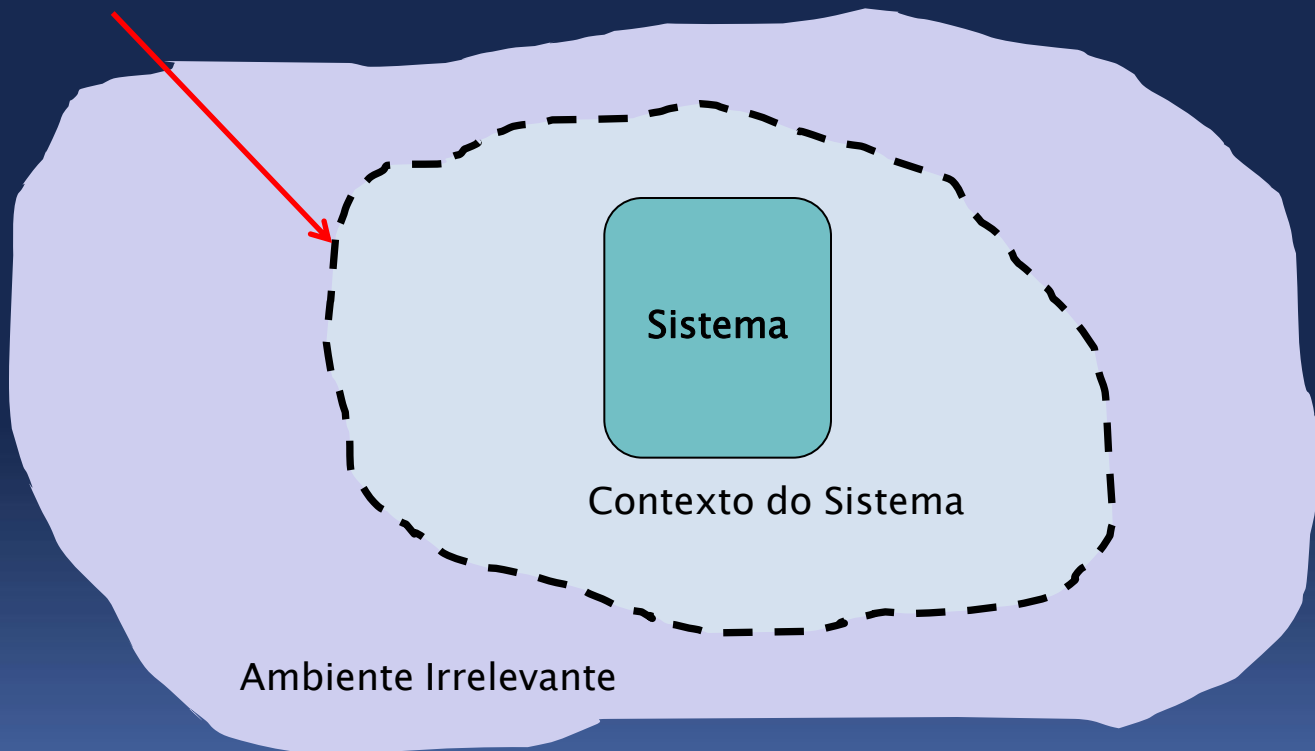
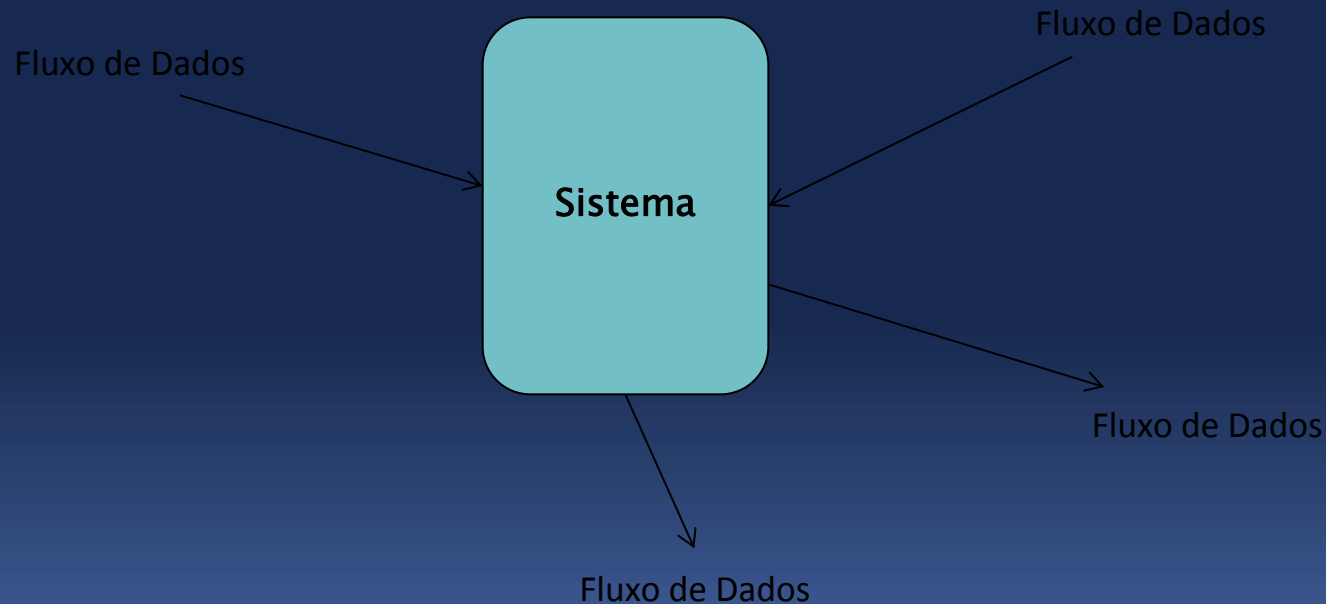
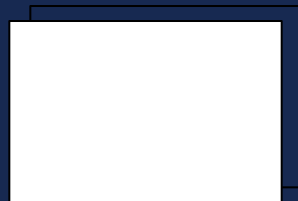


Diagrama de Contexto

- O **Diagrama de Contexto** é um caso especial do Diagrama de Fluxo de Dados (Análise Estruturada) no qual uma única bolha representa o sistema inteiro;
- Define a interconexão do sistema com o ambiente no qual está inserido;
- As pessoas, organizações ou sistemas com as quais o sistema a ser desenvolvido se comunica, são denominadas Entidades Externas.



Elementos do Diagrama de Contexto



Entidade Externa: Fonte de informações para o sistema ou destino de informações geradas pelo sistema



Fluxo de Dados



**Sistema
(processo)**

Diagrama de Contexto

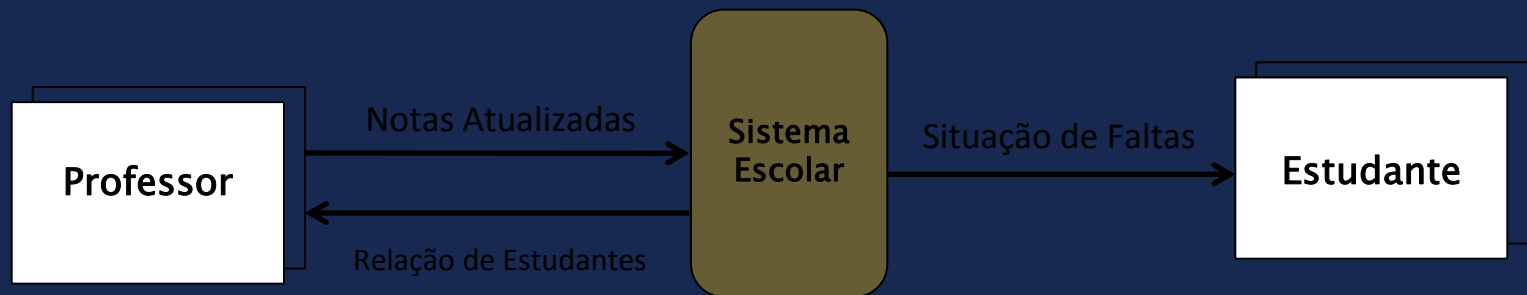
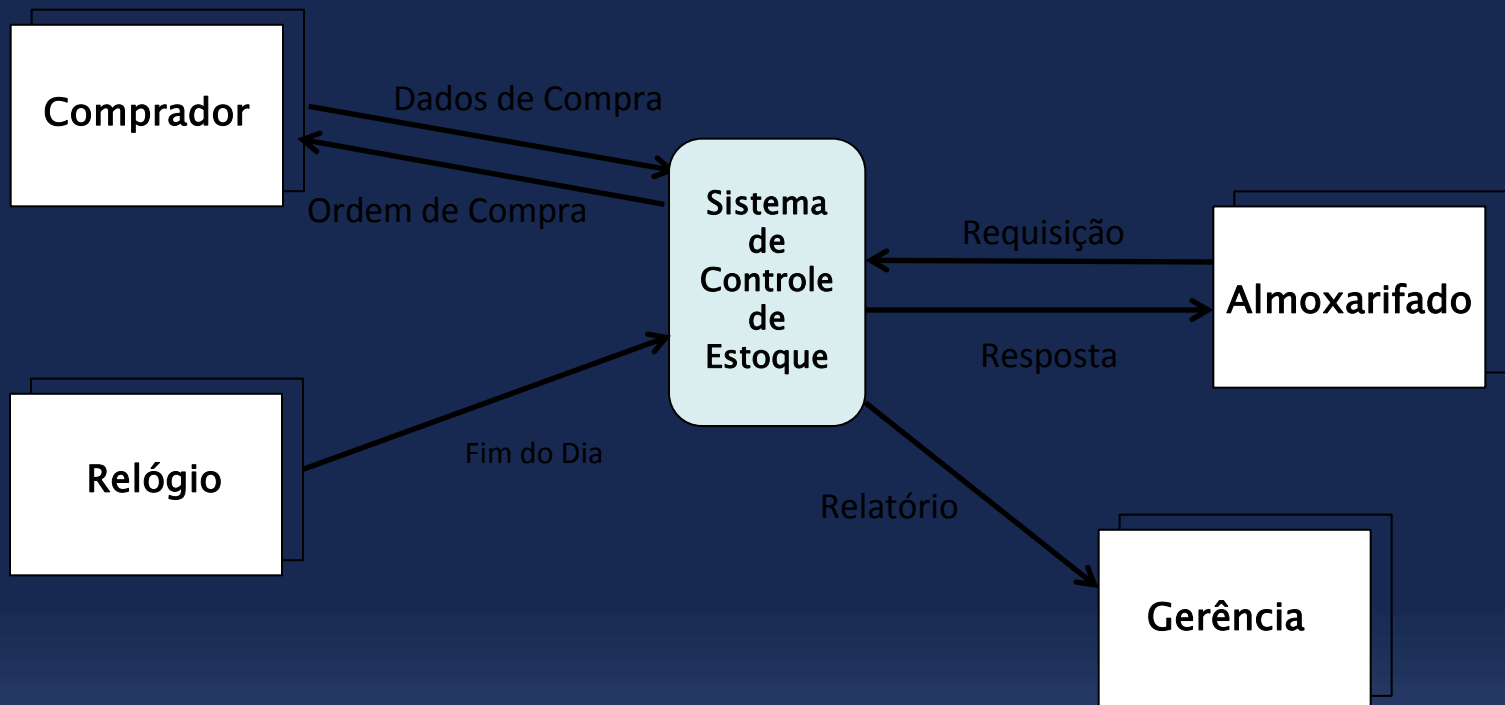


Diagrama de Contexto





O que significa Elicitar Requisitos ?



Elicitação de Requisitos

- Tarefa de identificar, descobrir, tornar explícito, obter o máximo de informações para o conhecimento do software a ser desenvolvido;
- Trás à tona os desejos e necessidades das partes interessadas;
- Auxilia a identificar o contexto do sistema.



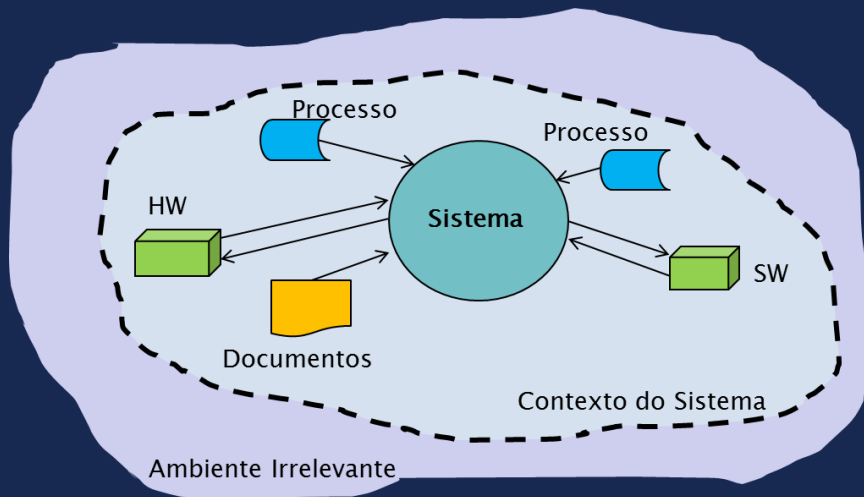


A elicitação de requisitos é feita a partir de que conhecimento ?



Elicitação de Requisitos

- A base para a elicitação de requisitos é formada pelo conhecimento do **CONTEXTO** do software a ser desenvolvido, obtido durante a Engenharia de Requisitos.



Fontes de Requisitos – Stakeholders



Stakeholders

- Partes Interessadas (Stakeholders)
- Documentos
- Sistemas em operação



Como identificar os stakeholders ?



Levantamento de Stakeholders

- Atividade que consiste em identificar os stakeholders relevantes (gerentes, especialistas do domínio, por exemplo);
- Acatar recomendações desses para identificar outros stakeholders.



Como lidar com os stakeholders ?



Interação com os Stakeholders

- Projetos complexos e “difíceis” podem envolver muitos stakeholders;
- Documentar stakeholders com informações tais como: nome, função (papel), dados pessoais, relevância do stakeholder, área, nível de expertise, objetivos e interesses no projeto, etc.
- Stakeholders que não recebem a devida atenção do Engenheiro de Requisitos podem assumir postura crítica em relação ao projeto.
- Stakeholders podem demonstrar desmotivação por estarem confortáveis com o sistema legado.
- Recomenda-se firmar acordos formais, para se evitar mal-entendidos e disputas sobre competências.



Direitos e Deveres do Engenheiro de Requisitos



- Falar a linguagem dos Stakeholders;
- Conhecer o domínio da aplicação;
- Ser capaz de documentar os resultados do trabalho (gráficos e diagramas);
- Manter relacionamento respeitoso com os stakeholders;
- Apresentar ideias, alternativas, bem como resultados;
- Aceitar sugestões dos stakeholders para simplificar e facilitar o uso da aplicação;
- Assegurar que o sistema atenda às exigências funcionais e de qualidade dos stakeholders.

Direitos e Deveres dos Stakeholders



- Auxiliar a inclusão do Engenheiro de Requisitos no domínio da aplicação;
- Suprir o Engenheiro de Requisitos com requisitos;
- Tomar decisões em tempo hábil;
- Respeitar estimativas de custo e viabilidade feitas pelo ER;
- Priorizar requisitos;
- Inspecionar os requisitos que o ER documenta (como protótipos, etc);
- Comunicar imediatamente mudanças de Requisitos;
- Respeitar o processo de Engenharia de Requisitos implementado.

Categorização de Requisitos

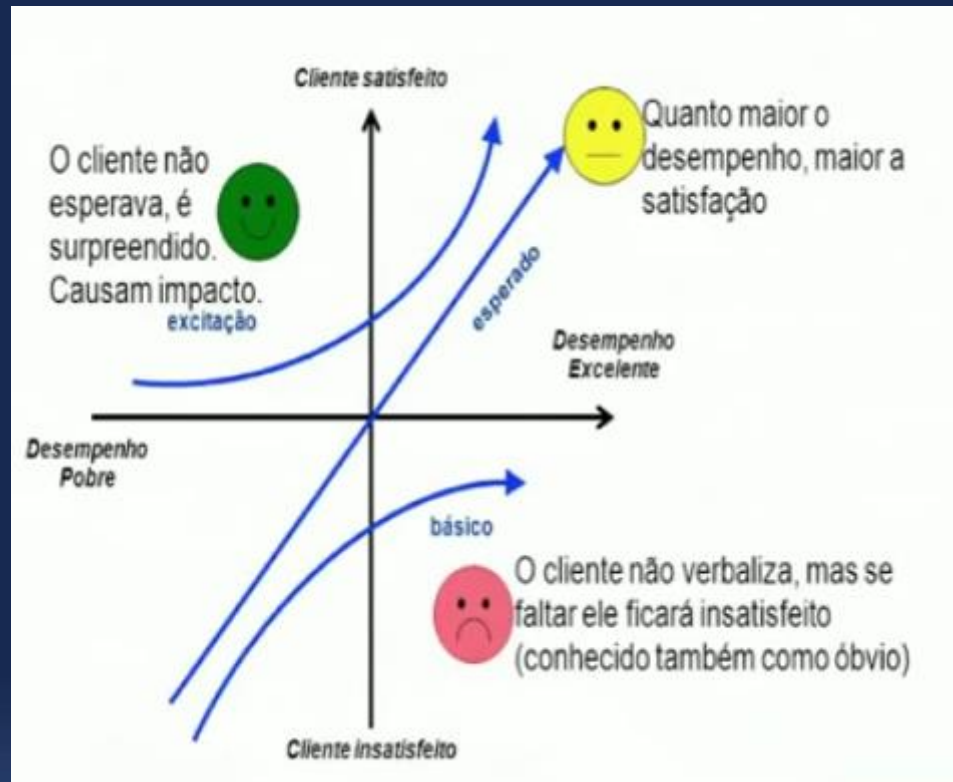
Modelo de Kano



Os fatores que determinam a satisfação do cliente são classificados nas seguintes categorias:

- ✓ **Fatores básicos de satisfação:** Propriedades evidentes e pressupostas; (auto evidentes e tidas como certas) (conhecimento subconsciente)
- ✓ **Fatores esperados de satisfação:** Propriedades explicitamente exigidas do sistema; (conhecimento consciente)
- ✓ **Fatores inesperados de satisfação:** Propriedades do sistema que o stakeholder não conhece ou espera, e que ele descobre apenas ao utilizar o sistema - encantamentos. (conhecimento inconsciente)

Modelo de Kano



Fatores **Básicos** de Satisfação



- Devem ser atendidos pelo sistema de qualquer maneira;
- Do contrário, os stakeholders ficarão decepcionados (**dissatisfiers**);
- Mesmo se atendidos, não geram uma atitude positiva por parte dos stakeholders;
- São predominantemente influenciados pelos sistemas existentes;
- Assim, técnicas de observação e técnicas centradas em análise de documentos são adequadas para se elicitar esses fatores.



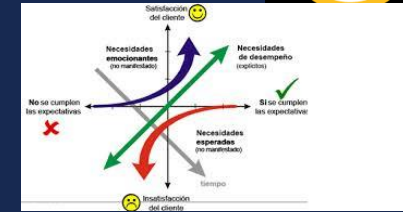
Fatores Esperados de Satisfação



- Correspondem aos requisitos conscientes (explícitos);
- Propriedades conscientemente conhecidas e explicitamente exigidas pelos stakeholders;
- Quando atendidas, os stakeholders ficam contentes e satisfeitos (daí o termo **satisfiers**), o que é desejável;
- Se algumas dessas propriedades estiverem faltando, provavelmente o stakeholder **não** aceitará o sistema;
- São comumente elicitados por meio de questionários e entrevistas.



Fatores **Inesperados** de Satisfação



- Propriedades do sistema cujo valor somente é reconhecido quando o stakeholder pode testar o sistema na prática, ou quando o ER as propõe (**delighters**);
- Correspondem aos **fatores inconscientes** de conhecimento (ocultos);
- Geram encantamentos dos stakeholders;
- Técnicas de criatividade são as mais indicadas para elicitar esses tipos de fatores.

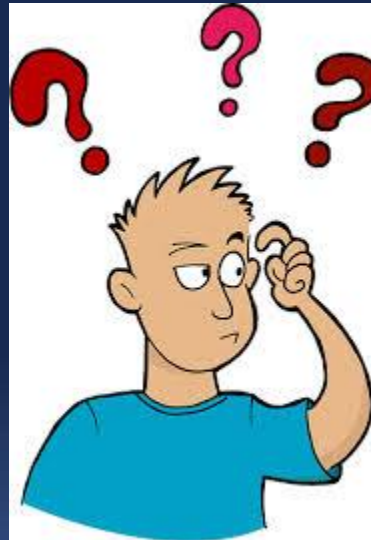


Técnicas de Elicitação

- Têm a finalidade de identificar os requisitos conscientes, inconscientes e subconscientes dos stakeholders.



Quais os fatores que influenciam na
escolha das Técnicas de Elicitação ?



Fatores para a escolha da Técnica de Elicitação

- Distinção entre requisitos conscientes, inconscientes e subconscientes;
- Restrições em termos de Tempo e Orçamento, bem como disponibilidade dos stakeholders;
- Experiência do ER;
- Oportunidades e Riscos do Projeto.



Técnicas de Pesquisa

- Partem do pressuposto de que o respondente é capaz de expressar explicitamente seu conhecimento;
- Respondente está comprometido a investir tempo e esforço para a elicitación;
- São geralmente dirigidas pelo ER;
- ER que faz as perguntas;
- Stakeholder pode se esquecer ou ignorar preocupações;
- Baseiam-se em entrevistas ou questionários.



Quais as vantagens e desvantagens da Técnica de Pesquisa baseada em Entrevistas ?

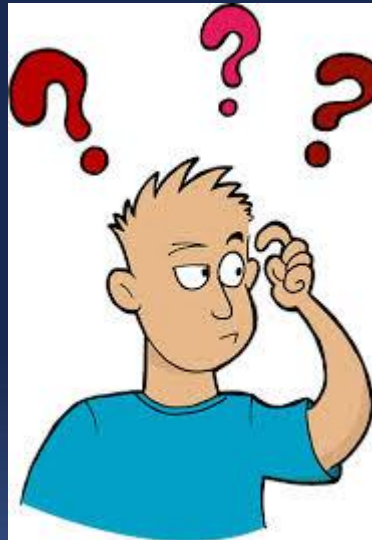


Entrevistas

- ER faz as perguntas para um ou mais Stakeholders;
- ER documenta as respostas;
- Perguntas bem formuladas podem revelar requisitos subconscientes;
- Principal desvantagem: tempo consumido para a elicitação.



Quais as vantagens e desvantagens da Técnica de Pesquisa baseada em Questionários ?



Questionários

- ER pode fazer uso de questões abertas ou fechadas (ex. Múltipla escolha) ;
- Para grande número de Stakeholders, questionário online é uma opção viável;
- Podem elicitar grande número de informações em curto espaço de tempo;
- Baixo custo;
- **Desvantagem**: coleta de requisitos que o ER já conhece ou conjectura;
- Não fornecem retroalimentação imediata entre stakeholder e ER.



Técnicas de Criatividade

- Tem a finalidade de desenvolver requisitos inovadores;
- Podem esboçar uma visão inicial do sistema;
- Podem elicitar fatores inesperados de satisfação (delighters);
- Em geral, não são adequadas para estabelecer requisitos precisos sobre comportamento do sistema;
- Dentre as técnicas, cita-se: brainstorming e mudança de perspectiva.



Brainstorming

- Ideias são coletadas durante certo período de tempo, geralmente com grupos de 5 a 10 pessoas;
- Ideias são documentadas por um moderador, sem serem discutidas, julgadas ou comentadas inicialmente;
- Participantes usam ideias de outros participantes para desenvolver ou modificar ideias existentes;
- Em outro tempo, as ideias coletadas são submetidas à uma análise rigorosa;
- Vantagem: grande número de ideias são coletadas em curto espaço de tempo;
- Desvantagem: Menos eficaz quando a dinâmica do grupo é confusa ou quando há membros muito dominantes no grupo.



Brainwriting

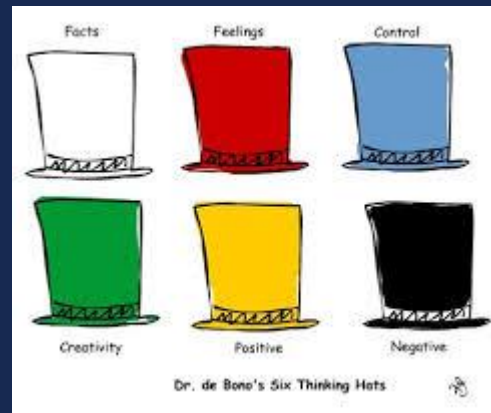
6-3-5 Brainwriting

- Corresponde ao método 6-3-5;
- Seis participantes, cada um escreve três ideias numa folha de papel;
- Essa folha é repassada aos outros membros, que comentam e desenvolvem as ideias recebidas, processo que se repete 5 vezes.



Mudança de Perspectiva

- Técnica conhecida por “Seis chapéus do pensamento” [DeBono, 2006]



- Cada chapéu representa uma perspectiva adotada, uma após a outra, por cada participante;
- As soluções daí resultantes abordam o problema a partir de perspectivas diferentes;
- Técnica benéfica quando stakeholders estão obstinadamente presos à suas opiniões;
- Difícil de ser aplicada se os requisitos exigem um nível de detalhamento muito grande, pois isso tornaria a técnica muito cansativa.

Técnicas baseadas em Documentos

- Reutilizam soluções e experiências feitas com sistemas existentes;
- Ao substituir-se um sistema legado, essa técnica assegura que a funcionalidade completa do sistema possa ser identificada;
- Devem ser combinadas com outras técnicas de elicitação para que novos requisitos possam ser identificados;
- Fazem parte dessa técnica:
 - Arqueologia do Sistema: Extração de informações a partir da documentação ou código do sistema legado;
 - Leitura baseada em perspectiva: Foca-se em partes específicas da documentação do sistema, a partir de uma perspectiva específica, por exemplo: do implementador ou do testador.
 - Reutilização: Consulta-se documentação de requisitos que foram compilados anteriormente e reutiliza-os.

Técnicas de Observação

- ER observa os stakeholders enquanto trabalham;
- ER documenta todos os passos e elicit os processos que o sistema deverá suportar, bem como potenciais erros, riscos e questões em aberto;
- Técnica apropriada para elicit requisitos detalhados e **fatores básicos de satisfação** (**dissatisfiers**), pois o ER pode identificar fatores considerados óbvios, ou aqueles que os stakeholders apenas conhecem no subconsciente.
- Permite que o ER se familiarize com a linguagem do domínio, o que simplifica o processo de elicitção.





Técnicas de Apoio

- Mapas Mentais
- Workshop
- CRC – Class Responsibility Collaboration
- Gravações de áudio e vídeo
- Modelagem de Casos de Uso
- Protótipos



Documentação de Requisitos

- Na Engenharia de Requisitos, informações levantadas durante as diversas atividades devem, obrigatoriamente, ser documentadas.



O que é Especificação de Requisitos ?



Especificação de Requisitos

- É uma coleção de requisitos **representada de forma sistemática**, tipicamente para um sistema ou componente, atendendo a determinados critérios.



Quais as razões para se documentar requisitos ?



Razões para se documentar Requisitos

- Requisitos formam a base para o desenvolvimento do sistema;
- Requisitos tem relevância legal ;
- Documentos de requisitos são complexos;
- Requisitos devem ser acessíveis para todas as partes envolvidas.



Tipos de Documentação

- Requisitos para um sistema podem ser documentados a partir de três perspectivas diferentes. (Estrutural (dados), Funcional, Comportamental);
- Essas perspectivas são complementares (uma não substitui a outra...);
- Na prática, tanto linguagem natural (texto) quanto a modelagem conceitual são utilizadas para esse fim, ou muitas vezes, emprega-se uma combinação apropriada entre as duas.



Documentação de Requisitos com Linguagem Natural

- É a forma de documentação de requisitos mais aplicada na prática;
- Vantagem: Stakeholders não precisam aprender nova notação;
- Entretanto, pode resultar em requisitos ambíguos.



Documentação de Requisitos com Modelos Conceituais

- Estão associadas à cada perspectiva;
- Pode-se retratar os requisitos de forma mais compacta;
- Apresentam menor grau de ambiguidade;
- No entanto, exige conhecimentos específicos de modelagem de software.



Diagramas – Modelos Conceituais

- Diagrama de Caso de Uso.
- Diagrama de Classes.
- Diagrama de Atividades.
- Diagrama de Estados



Estruturas Padronizadas de Documentos

- RUP – Rational Unified Process [Kruchten,2001];
- Norma IEEE standard 830-1998;
- Modelo-V [V-Modell 2004] – Ministério do Interior da Alemanha.



Estruturas Padronizadas de Documentos

■ Padrão ISO **29148:2011** sugere que o documento de requisitos tenha 5 partes:

- ✓ Informação introdutória e descrição geral do software;
- ✓ Lista de documentos referenciados;
- ✓ Lista de requisitos especificados;
- ✓ Medidas planejadas para verificação;
- ✓ Apêndices.



Estrutura do Documento de Requisitos

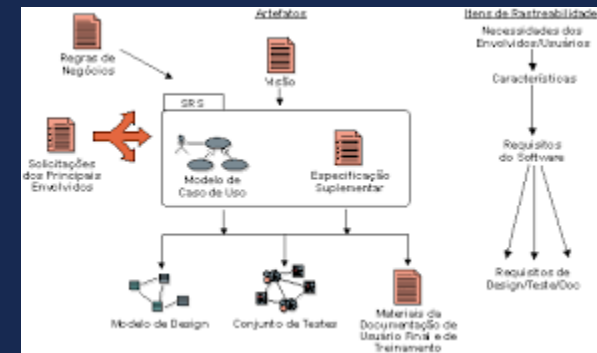
As seguintes questões deveriam ser abordadas por qualquer estrutura de documento de Especificação de Requisitos:



- ❖ Introdução
- ❖ Finalidade
- ❖ Cobertura do Sistema
- ❖ Stakeholder
- ❖ Definições, acrônimo e abreviações
- ❖ Referências
- ❖ Visão Geral
- ❖ Ambiente do Sistema
- ❖ Descrição da Arquitetura
- ❖ Funcionalidades do Sistema
- ❖ Usuários e público-alvo
- ❖ Restrições
- ❖ Pressupostos
- ❖ Requisitos
- ❖ Apêndices
- ❖ Índice

Usos do Documento de Requisitos

- ❖ Planejamento
- ❖ Análise e Projeto
- ❖ Implementação
- ❖ Teste
- ❖ Gerenciamento de Mudanças
- ❖ Uso do sistema e manutenção
- ❖ Gerenciamento do Contrato



Critérios de Qualidade para o Documento de Requisitos

IEEE 830-1998

- ✿ Não-ambiguidade e consistência;
- ✿ Estrutura clara;
- ✿ Modificabilidade e extensibilidade;
- ✿ Completude [IEEE-1998];
- ✿ Rastreabilidade [IEEE-1998].



Critérios de Qualidade para o Documento de Requisitos

IEEE 830-1998

Os critérios de qualidade definidos na norma IEEE 830-1998 podem ser aplicados tanto para requisitos individuais como para documentos completos de requisitos



- ✿ Acordado
- ✿ Priorizado
- ✿ Não-ambíguo
- ✿ Válido e atualizado
- ✿ Correto
- ✿ Consistente
- ✿ Verificável
- ✿ Realizável
- ✿ Rastreável
- ✿ Completo
- ✿ Compreensível

Regras fundamentais para a Legibilidade de Requisitos

- ❖ Frases Curtas e parágrafos curtos;
- ❖ Formular apenas um requisito por frase.



Glossário

- ❖ Uma causa frequente de **conflitos** na Engenharia de Requisitos ocorre quando pessoas envolvidas no processo de desenvolvimento de Software têm diferentes interpretações de termos.
- ❖ Um **glossário** é uma coleção de definição de termos, apresentando os seguintes elementos:
 - ✓ Termos técnicos do contexto
 - ✓ Abreviações e acrônimos
 - ✓ Conceitos do dia-a-dia
 - ✓ Sinônimos
 - ✓ Homônimos





Regras para utilizar o Glossário

- ❖ Tem que ser gerenciado de forma centralizada;
- ❖ Deve-se eleger quem fará a manutenção do glossário;
- ❖ Deve ser atualizado de forma constante;
- ❖ Deve ser acessível;
- ❖ Seu uso deve ser obrigatório;
- ❖ Deve conter a origem dos termos;
- ❖ Precisa ser acordado entre as partes interessadas;
- ❖ Deve ter uma estrutura consistente.

Documentação de Requisitos com Linguagem Natural

- ❖ Requisitos elicitados são frequentemente documentados usando linguagem natural;
- ❖ É mais facilmente compreendida pelos stakeholders;
- ❖ Não exige preparação dos stakeholders para compreender a especificação.



Efeitos da Linguagem Natural

- ✿ É inerentemente ambígua.
- ✿ Requisitos podem se manifestar de forma diferente em cada pessoa (“Efeitos Transformacionais”)



Efeitos Transformacionais

- ❖ Nominalização;
- ❖ Substantivos sem indicador de referência;
- ❖ Quantificadores universais;
- ❖ Condições especificadas de forma incompleta;
- ❖ Verbos de processo especificados de forma incompleta.



Nominalização

- ❁ Um processo (às vezes de longa duração) é convertido em um evento (singular)
- ❁ Dessa forma, todas as informações necessárias para descrever o processo acuradamente são perdidas.

Exemplo: Nominalização

“Em caso de **crash** do sistema, será realizado um **restart** do sistema”



- ❁ Os termos **crash** do sistema e **restart** descrevem, respectivamente, processos que deveriam ser analisados com maior precisão...
- ❁ Termos nominalizados não devem deixar qualquer margem de interpretação dos processos e deve representar o processo de forma precisa, incluindo quaisquer exceções que possam ocorrer, bem como todos os parâmetros de entrada e saída.
- ❁ Portanto, as nominalizações devem ser evitadas. Se presentes, deve-se examinar se foram suficientemente detalhadas em outra parte do Documento de Requisitos e, portanto, devem estar claras para todos os Stakeholders.

Substantivos sem indicador de Referência

- ❖ Substantivos especificados de forma **incompleta**;
- ❖ Linguistas referem-se a isso como a falta de um indicador de referência.

Exemplo: Substantivos sem indicador de referência

“Os dados deverão ser exibidos para o usuário no terminal”



- ❖ Quais dados ?
- ❖ Qual usuário ?
- ❖ Qual terminal, exatamente ?
- ❖ Requisito deveria ser: “O sistema deve exibir os dados de faturamento para o usuário cadastrado no terminal em que ele estiver logado”

Quantificadores Universais

- ❖ Especificam quantidades e frequências. Agrupam um conjunto de objetos e fazem uma declaração sobre o comportamento desse grupo; Utilizam os seguintes termos: **Nunca, Sempre, Não, Nenhum, Cada, Todos, Alguns, Nada**;
- ❖ Existe o risco do comportamento ou da propriedade especificada, **não** se aplicar à todos os objetos do conjunto especificado.

Exemplo: Quantificadores universais

“ O sistema deverá mostrar **todos** os dados em **cada** submenu ”



- ❖ Nesse casos, é preciso fazer as seguintes perguntas:
- ❖ Realmente em cada submenu ?
- ❖ Realmente todos os conjuntos de dados ?
- ❖ Quantificadores universais podem facilmente ser identificados por meio de “palavras-gatilho” (trigger words) tais como: nunca, sempre, nenhum, cada, todos, alguns ou nada.

Condições especificadas de forma incompleta

- ❖ Requisitos contendo condições especificam o comportamento que deve ocorrer quando a condição for atendida. Empregam os termos Se.. Então, No caso de , Se ... , Em função de ;
- ❖ Mas, também devem especificar o comportamento quando a condição não for atendida (essa parte costuma faltar nas especificações)

Exemplo: Condições especificadas de forma incompleta

“ O sistema do restaurante deverá oferecer todas as bebidas para um cliente registrado com mais de 20 anos”



- ❖ Que bebidas o sistema vai oferecer para um cliente que tenha **20 anos ou menos** ?
- ❖ Se essa questão for esclarecida, o requisito poderá ser formulado como segue:
- ❖ “ O sistema do restaurante deverá oferecer:
Todas as bebidas não alcoólicas para qualquer usuário registrado com menos de 21 anos. Todas as bebidas, inclusive as alcoólicas, para qualquer usuário registrado acima de 20 anos.”

Verbos de processo especificados de forma incompleta

- ❖ Alguns verbos de processo requerem mais de um substantivo para serem especificados de forma completa. Por exemplo, o verbo “**transmitir**” exige pelo menos três suplementos para ser considerado completo: **O que é transmitido, de onde é transmitido e para onde é transmitido;**
- ❖ Isso pode ser evitado, na maioria das vezes, se os requisitos passarem a ser formulados na voz ativa e não na voz passiva.

Exemplo: Verbos de processo especificados de forma incompleta

“ Para logar um usuário, os dados de login são inseridos”



- ❖ O requisito foi formulado na voz passiva. Não está claro quem vai inserir os dados de login.
- ❖ Também não está claro onde e como isso será feito.
- ❖ Se esse requisito fosse especificado na voz ativa, a formulação poderia ser:

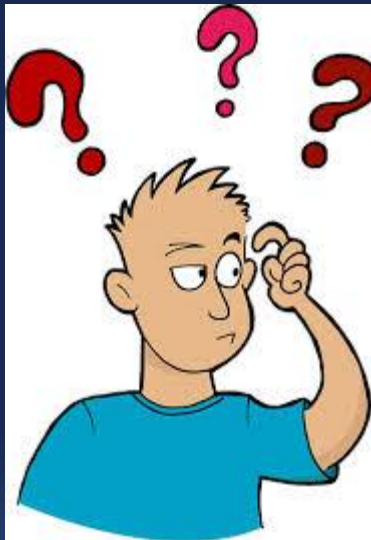
“O sistema deve permitir ao usuário inserir seu username e senha usando o teclado do terminal”

Requisitos com o uso de Templates

- ❖ Fornecem uma abordagem simples para se reduzir os efeitos transformacionais de linguagem ao se documentar requisitos.
- ❖ Fornecem o apoio para que o ER possa obter alta qualidade e não-ambiguidade sintática, com baixo custo e tempo otimizado.



O que é Template de Requisitos ?



Template de Requisitos

- ✿ É um padrão para a estrutura sintática de requisitos individuais.

