

Elicitação e Análise de Requisitos

Compreendendo Requisitos

Departamento de Computação
Prof. Alberto Sales

Engenharia de Requisitos

- ❑ **Requisito:** Uma função, restrição ou outra propriedade que o sistema deve prover para suprir as necessidades dos usuários interessados no sistema.
 - ❑ **Engenharia :** implica que técnicas sistemáticas e repetível devem ser usadas
 - ❑ **Engenharia de Requisitos** significa que os requisitos para um produto são definidos, gerenciados e testados sistematicamente
-

Engenharia de Requisitos

- Os requisitos para um sistema são as descrições dos serviços que um sistema deve fornecer e as restrições ao seu funcionamento.
 - Esses requisitos refletem as necessidades de clientes para um sistema que serve a um determinado propósito, como controlar um dispositivo, colocar um pedido ou localização de informações.
 - O processo de descobrir, analisar, documentar e a verificação desses serviços e restrições é chamada de engenharia de requisitos (RE).
-

Engenharia de Requisitos

- ❑ O termo **requisito** não é usado consistentemente na indústria de software.
 - ❑ Em alguns casos, um **requisito** é simplesmente uma declaração abstrata de alto nível de um serviço que um sistema deve fornecer ou uma restrição em um sistema.
-

Engenharia de Requisitos

- ❑ É essencial que uma equipe de Engenharia de Software compreenda os requisitos dos problemas antes de tentar resolvê-los.
 - ❑ Em alguns casos engenharia de requisitos deve ser abreviada, mas nunca abandonada
 - ❑ ER são ações de Engenharia de Software que iniciam com as atividades de comunicação e continua na atividade de modelagem.
-

Engenharia de Requisitos

- ❑ ER estabelece uma base sólida para projeto e construção de software.
 - ❑ Sem ela, o software que resulta tem uma alta probabilidade de não atender as necessidades dos clientes.
-

Engenharia de Requisitos

- ❑ ER estabelece uma base sólida para projeto e construção de software.
 - ❑ Sem ela, o software que resulta tem uma alta probabilidade de não atender as necessidades dos clientes.
-

Características De Um Bom Requisito

- Claro e não ambíguo
 - Estrutura padrão
 - Tem uma única possibilidade de interpretação
 - Não mais que um requisito em uma sentença

 - Adequado
 - Um requisito contribui para uma necessidade real
-

Características De Um Bom Requisito

- ☐ Compreensível
 - Um leitor pode facilmente compreender o significado do requisito
 - ☐ Verificável
 - Um requisito pode ser testado
 - ☐ Completo
 - ☐ Consistente
 - ☐ Rastreável
 - Todo e qualquer requisito que utilize alguma outra funcionalidade deve ser rastreável
-

Análise de Requisitos: a Engenharia da Criação

- Na perspectiva do processo de software, a engenharia de requisitos é uma ação importante que se inicia durante a atividade de comunicação e continua na modelagem, construindo uma ponte para o projeto e para a construção (Pressman).
 - Um requisito é definido como “uma condição ou uma capacidade com a qual o sistema deve estar de acordo”.
 - Pode ser desde uma indicação abstrata, de alto nível, até uma especificação matemática detalhada.
-

Análise de Requisitos: a Engenharia da Criação

- Em resumo, definem o que o sistema deve fazer e sob quais limitações ele é requisitado a operar.
 - São exemplos de requisitos:
 - “O sistema deve ser capaz de dar baixa no pagamento de uma Nota Fiscal”;
 - “O sistema deve ser capaz de realizar pagamentos por transferências bancárias”;
 - “O sistema deve suportar pelo menos 20 transações por segundo”;
 - “O sistema deve realizar backup uma vez por dia.”
-

Por que está recebendo bons requisitos engessados?

- ☐ Interessados não sabem o que eles realmente querem.
 - ☐ Interessados expressam requisitos em seus próprios termos.
 - ☐ Diferentes interessados podem ter requisitos conflitantes.
 - ☐ Fatores organizacionais e políticos podem influenciar os requisitos do sistema.
 - ☐ Os requisitos mudam durante o processo de ER. Novos interessados podem surgir e mudar o ambiente do negócio.
-

Tarefas da Engenharia de Requisitos

- ❑ **Concepção** – Estabelece uma compreensão básica do problema e a natureza da solução
 - ❑ **Elicitação** – Obter os requisitos dos interessados.
 - ❑ **Elaboração (Altamente Estruturado)** – Criar um modelo de análise que representa as informações, funções e aspectos comportamentais dos requisitos.
 - ❑ **Negociação** – Acordo sobre a entrega de sistema que é realístico para desenvolvedores e clientes.
 - ❑ **Especificação** – Descrever requisitos formais e informais.
 - ❑ **Validação** — Revisar as especificações dos requisitos com relação a erros, ambiguidade, omissões e conflitos.
 - ❑ **Gerenciamento de Requisitos** — Gerenciar mudanças de requisitos.
-

Aprofundando em Requisito de Software

- Um analista de sistemas deve entender o domínio da informação antes de construir um produto de software. Ele deve conhecer as funções, comportamento, desempenho e interface necessários para serem usados no software. Os requisitos para o sistema e software são documentados e revisados com o cliente para análise posterior.
 - Conceitualmente, a análise de requisitos inclui três tipos de atividade:
 - Elicitação de requisitos
 - Analisar requisitos
 - Requisitos de registro
-

Aprofundando em Requisito de Software

- Elicitação de requisitos: A tarefa de se comunicar com clientes e usuários para determinar quais são seus requisitos. Isso às vezes também é chamado de coleta de requisitos.
 - Analisar requisitos: Determinar se os requisitos declarados não são claros, incompletos, ambíguos ou contraditórios e, em seguida, resolver esses problemas.
-

Aprofundando em Requisito de Software

- Forma de Registros de Requisitos:
 - Os requisitos podem ser documentados de várias formas, como documentos em linguagem natural, casos de uso, histórias de usuários ou especificações de processos.
-

Aprofundando em Requisito de Software

- ❑ A análise de requisitos pode ser um processo longo durante o qual muitas habilidades psicológicas delicadas estão envolvidas.
 - ❑ Novos sistemas mudam o ambiente e as relações entre as pessoas envolvidas, levam em conta todas as suas necessidades e garantem que eles entendam as implicações dos novos sistemas.
-

Aprofundando em Requisito de Software

- ❑ Os analistas podem empregar várias técnicas para obter os requisitos do cliente. Isso inclui realizar entrevistas ou grupos focais e criar listas de requisitos.
 - ❑ Técnicas mais modernas incluem prototipagem e casos de uso. Quando necessário, o analista empregará uma combinação desses métodos para estabelecer os requisitos exatos das partes interessadas, de modo que seja produzido um sistema que atenda às necessidades do negócio.
-

Elicitação e Análise de Requisitos

Técnicas

Formas aplicadas na descoberta de requisitos de software

Concepção

- Fazer perguntas livres de contexto que estabelecem ...
 - Compreensão básica do problema
 - As pessoas que querem uma solução
 - A natureza da solução que é desejada, e
 - A eficácia da comunicação preliminar e colaboração entre o cliente e o desenvolvedor
-

Elicitação

- ☐ Elicita (identifica) requisitos dos clientes, usuários e outros.
 - Encontrar saída para clientes, usuários e outros que os objetivos dos produtos são
 - ☐ O que é para ser feito
 - ☐ Como o produto se encaixa às necessidades de negócios, e
 - ☐ Como os produtos são utilizados no dia a dia
-

Por que Elicitação de Requisitos é difícil?

□ Problemas de escopo:

- A fronteira do sistema é mal definida
- Clientes/usuários especificam detalhes técnicos desnecessários que pode confundir em vez de clarificar os objectivos

□ Problemas de compreensão

- Clientes não estão completamente certos do que necessitam.
 - Clientes tem uma compreensão pobre das capacidades e limitações do ambiente computacional
 - Cliente não tem uma completa compreensão do domínio dos problemas.
-

Por que Elicitação de Requisitos é difícil?

□ Problemas de compreensão

- Clientes tem problemas de comunicação com o engenheiro de sistemas.
- Clientes omitem detalhes que acreditam ser óbvios
- Clientes especificam requisitos que conflitam com outros requisitos.
- Clientes especificam requisitos que são ambíguos ou não capazes de serem testados

□ Problemas de Volatilidade.

- Requisitos mudam com o tempo
-

Elicitação



Como o cliente explicou



Como o líder de projeto entendeu



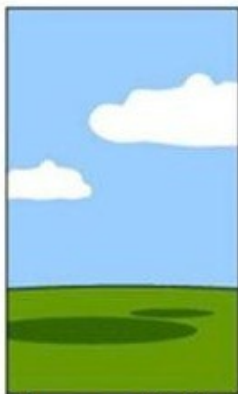
Como o analista projetou



Como o programador codificou



Como o consultor de negócio descreveu



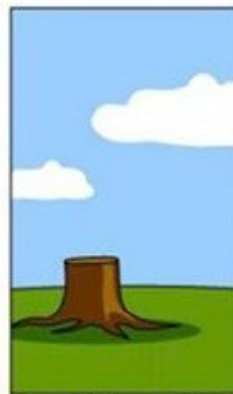
Como o projeto foi documentado



Como o produto foi instalado



Como cobraram do cliente



Como foi suportado



O que o cliente realmente precisava

Engenharia de Requisitos: Superando Problemas

- Para ajudar a superar estes problemas, os desenvolvedores devem abordar os requisitos de forma simples, prática e organizada. Os seguintes passos são recomendados:
 - Avaliar a viabilidade técnica e de negócio para o sistema proposto;
 - Identificar as pessoas que vão auxiliar a explicar os requisitos e compreender seus preconceitos organizacionais;
 - Definir o ambiente técnico no qual o sistema será instalado;
-

Engenharia de Requisitos: Superando Problemas

- Para ajudar a superar estes problemas, os desenvolvedores devem abordar os requisitos de forma simples, prática e organizada. Os seguintes passos são recomendados:
 - Identificar regras de domínio que limitam a funcionalidade ou desempenho do software a ser construído;
 - Definir um ou mais métodos de elicitação dos requisitos;
 - Solicitar a participação de várias pessoas a partir de diversos pontos de vista;
 - Identificar requisitos ambíguos que serão candidatos a prototipação.
-

Engenharia de Requisitos: Superando Problemas

- Lista de clientes/usuários que participaram da elicitação de requisitos;
 - Cenários de usos capazes de fornecer uma ideia de uso sistema;
 - Protótipo que porventura tenha sido desenvolvido.
-

Engenharia de Requisitos: Superando Problemas

- Cada documento deve ser revisado por todos que participaram desta fase do projeto.
 - Uma boa elicitação de requisitos parte das seguinte características:
 - Definir técnicas de coleta baseadas em fatores operacionais, táticos e financeiros;
 - Criar um planejamento capaz de alcançar as metas estabelecidas, como prazo, curso e qualidade;
 - Promover a integração e comprometimento de todos os envolvidos no processo;
 - Identificar os documentos e procedimentos que definem as políticas de negócio da empresa.
-

Engenharia de Requisitos: Fases de Elicitação

- Um projeto de Elicitação de Requisitos tem as seguinte fases:
 - Planejamento (Como deve ser feito): Identificar fontes e técnicas;
 - Elicitação de Requisitos (O que deve ser coletado): Identificar funcionalidades, identificar requisitos e riscos;
 - Documentação (Como devo documentar): Documento de visão.
-

Engenharia de Requisitos: Tipos de Requisitos



Engenharia de Requisitos: Fases de Elicitação

☐ REQUISITOS FUNCIONAIS

- Estes definem funcionalidades do sistema, ou seja, aquilo que deve fazer, as funções necessárias para atender os objetivos da aplicação. Exemplo:
 - ☐ Cadastrar Alunos;
 - ☐ Emitir Boletim de notas;
 - ☐ Fazer uma transação no banco de dados;
 - ☐ Cadastrar um registro de atendimento;
 - ☐ Imprimir relatórios;
 - Podem ser escritos em alto nível, se forem voltados ao cliente, ou podem ser especificados em detalhe, para desenvolvedores.
-

Engenharia de Requisitos: Fases de Elicitação

❑ REQUISITOS NÃO FUNCIONAIS

- Expressam restrições sob as quais o sistema deve operar ou qualidades específicas que o software deve ter, como performance, portabilidade, segurança, usabilidade, etc.
 - Descrevem também a qualidade do serviço (QoS).
 - A não consideração ou esquecimento desses fatores constitui uma das principais razões de uma eventual insatisfação dos usuários com relação ao software.
-

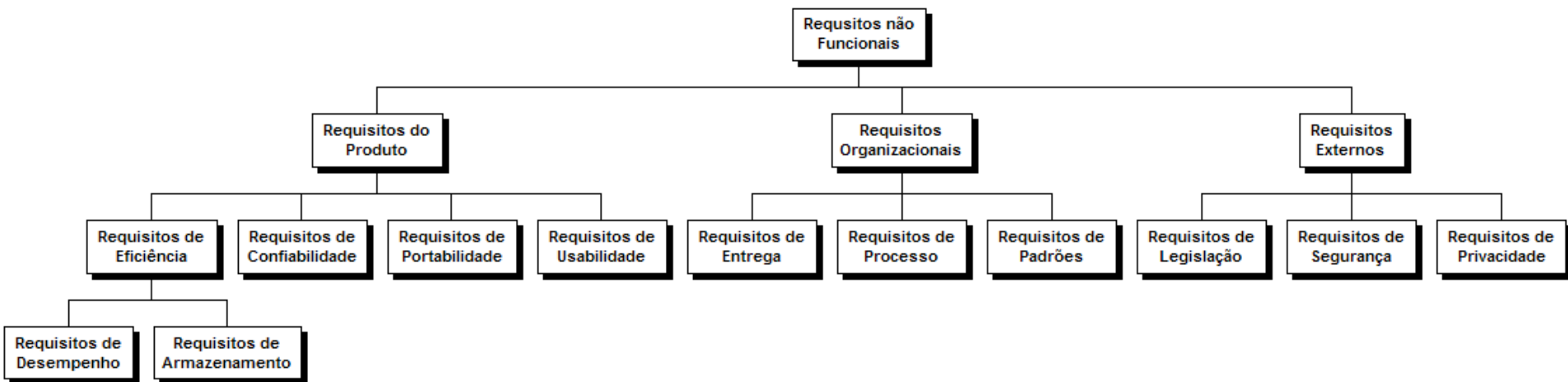
Engenharia de Requisitos: Fases de Elicitação

- Os requisitos não funcionais também são chamados de RNF ou Requisitos Suplementares.
 - Confidencialidade, Portabilidade, Confiabilidade;
 - Precisão, Performance, Integridade;
 - Qualidade, Segurança, Usabilidade;
 - Etc.
-

Engenharia de Requisitos: Fases de Elicitação

- Os requisitos não funcionais dividem-se em outros três grupos de requisitos:
 - a) Requisitos do produto: Requisitos que especificam que o software entregue deve se comportar de um determinado modo, por exemplo: ser confiável, robusto, rápido, etc.
 - b) Requisitos organizacionais: Requisitos que são consequência das políticas e procedimentos organizacionais, como padrões, processos, etc.
 - c) Requisitos externos: Requisitos que são externos ao sistema e seu desenvolvimento, por exemplo: legislação, interoperabilidade, etc.
-

Engenharia de Requisitos: Fases de Elicitação



Engenharia de Requisitos: Fases de Elicitação

- Os Requisitos não funcionais podem ser extremamente difíceis de especificar de forma precisa. Porém, devem ser verificáveis usando alguma medida que possa ser objetivamente testada.

PROPRIEDADE	MEDIDA
Desempenho	Transações por segundo; Tempo de resposta para eventos; etc.
Armazenamento	Megabytes; Número de chips ROM;
Usabilidade	Tempo de treinamento; Número de cliques de mouse;
Confiabilidade	Tempo médio entre falhas; Taxa de ocorrência de falhas; Disponibilidade;
Robustez	Tempo para recomençar depois de uma falha; Probabilidade de corrupção de dados após falha;
Portabilidade	Porcentagem de declarações dependentes de plataforma; Número de plataformas-alvo.

Engenharia de Requisitos: Fases de Elicitação

- Além dos requisitos funcionais e não funcionais existem ainda outras definições de requisitos que não serão estudadas, mas que valem ser citadas:
 - **Requisitos de Domínio:** vêm do domínio da aplicação do sistema e refletem características do domínio. São transformados, posteriormente, em requisitos funcionais ou restrições (RNFs).
 - Exemplo: A desaceleração do Elevador deve ser computada como: $D_{\text{elevador}} = D_{\text{controle}} + D_{\text{gradiente}}$;
 - **Requisitos permanentes (estáveis):** derivados da atividade principal da organização.
 - Exemplo: em um hospital sempre haverá requisitos relativos aos médicos, aos pacientes, aos tratamentos, etc. Derivados do modelo de domínio;
 - **Requisitos Voláteis:** Requisitos que se modificam durante o desenvolvimento ou quando o sistema está em uso. Exemplo: Requisitos resultantes de políticas governamentais;
-

Engenharia de Requisitos: Identificação de Riscos

- Os riscos são a principal razão de falhas em um projeto de software. Identificando-os é possível criar um plano para reduzi-los.
 - No documento de visão é necessário identificar uma lista de riscos existentes, como:
 - **Política:** Influência de política de negócios, leis, decretos ou normas que regulam a finalidade da aplicação;
 - **Tecnologia:** Ferramentas emergentes e integração com sistemas legados;
 - **Recursos:** Orçamento restrito, contratação de terceiros;
 - **Habilidade:** Falta de domínio da tecnologia (habilidade e experiência);
 - **Requisitos:** Requisitos não plenamente conhecidos.
-

Engenharia de Requisitos: Identificação de Restrições

- São restrições impostas sobre o desenvolvimento de software, como a adequação a custos e prazos, a plataforma tecnológica, aspectos legais (licenciamento), limitações sobre a interface do usuário, componentes de hardware e softwares a serem adquiridos.
 - Exemplo:
 - O projeto requer uma tecnologia como webservices;
 - Necessita de hardware específico como um servidor exclusivo de banco de dados.
-

Engenharia de Requisitos: Documentos de visão

- Depois de concluída a identificação e elicitação de requisitos é necessário documentar o que foi feito através do Documento de Visão. Este documento tem as seguintes seções:

- Declaração dos problemas;
- Lista de Stakeholders;
- Lista de requisitos;
- Lista de Riscos;
- Lista de Restrições.

Data: _____	Autor: _____	Revisão: _____
Índice:		
1.0 - Introdução		
1.1 Objetivo do documento		
1.2 Escopo		
1.3 Abreviaturas, Siglas e etc.		
2.0 Contexto		
2.1 Declaração do Problema		
3.0 Lista de Stakeholders		
3.1 Stakeholders primários		
3.2 Stakeholders secundários		
4.0 Lista dos Requisitos		
4.1 Requisitos funcionais		
4.2 Requisitos não funcionais		
5.0 Lista dos Riscos		
6.0 Lista das Restrições		
6.1 Software		
6.2 Hardware		
6.3 Ambiente e Tecnologia		

Elaboração

- ❑ Focada no desenvolvimento de um modelo técnico refinado de funções de software , recursos e restrições utilizando as informações obtidas durante a concepção e elicitação
 - ❑ Criar um modelo de análise que identifica dados, funções e requisitos comportamentais.
-

Elaboração

- ❑ É impulsionada pela criação e refinamento de cenários de usuários que descrevem a forma de como o usuários final irá interagir com o sistema.
 - ❑ Cada evento é analisado na extração.
 - ❑ O resultado final define o domínio das informações, funções e comportamentos do problema.
-

Elaboração

- Depois que os requisitos foram coletados, os produtos de trabalho servem como base para a análise de requisitos. A análise de requisitos visa a descobrir alguns problemas e torná-los mais consistentes antes da especificação formal.
 - Neste ponto, o documento de visão é um documento importante na análise de requisitos.
-

Documentos de Requisitos

- ❑ O resultado final desta fase é um documento de requisitos cujo objetivo é classificar e descrever os requisitos de software, usuários e entidades externas.
 - ❑ O formado do documento de requisitos é sugerido pela IEEE/ANSI 830-1993 e propõe a estrutura exemplificada na figura
-

Elaboração

- A análise de requisitos deve ser:
 - **Correta:** O requisito deve ser encontrável no software;
 - **Não ambígua:** Deve ter apenas uma interpretação;
 - **Completa:** Deve incluir RF e RNF;
 - **Consistente:** Não existir conflito entre os requisitos;
 - **Verificável:** Possibilidade de verificar e validar cada requisito;
 - **Modificável:** Requisitos serem facilmente alterados.
-

Especificação de Requisitos

- O termo especificação tem vários significados, podendo ser:
 - Um documento escrito;
 - Um modelo gráfico;
 - Um modelo matemático formal;
 - Uma coleção de cenários de uso, etc.
-

Especificação de Requisitos

- A abordagem utilizada depende da necessidade específica de cada projeto, podendo ser documentos escritos combinados com modelos gráficos para sistemas maiores e cenários de uso para sistemas mais simples, etc.
 - Os diagramas da **UML** podem ser largamente empregados para esta tarefa.
-

Modelo de Documento para Especificação de Requisitos

Data: _____ | Autor: _____ | Revisão: _____

1. INTRODUÇÃO

- 1.1. Propósito do documento
- 1.2. Escopo do produto
- 1.3. Definições, acrônimos e abreviações
- 1.4. Referências
- 1.5. Visão Geral do restante do documento.

2. DESCRIÇÃO GERAL

- 2.1. Perspectiva do produto
- 2.2. Funções do produto
- 2.3. Características do usuário
- 2.4. Restrições Gerais
- 2.5. Suposições e dependências

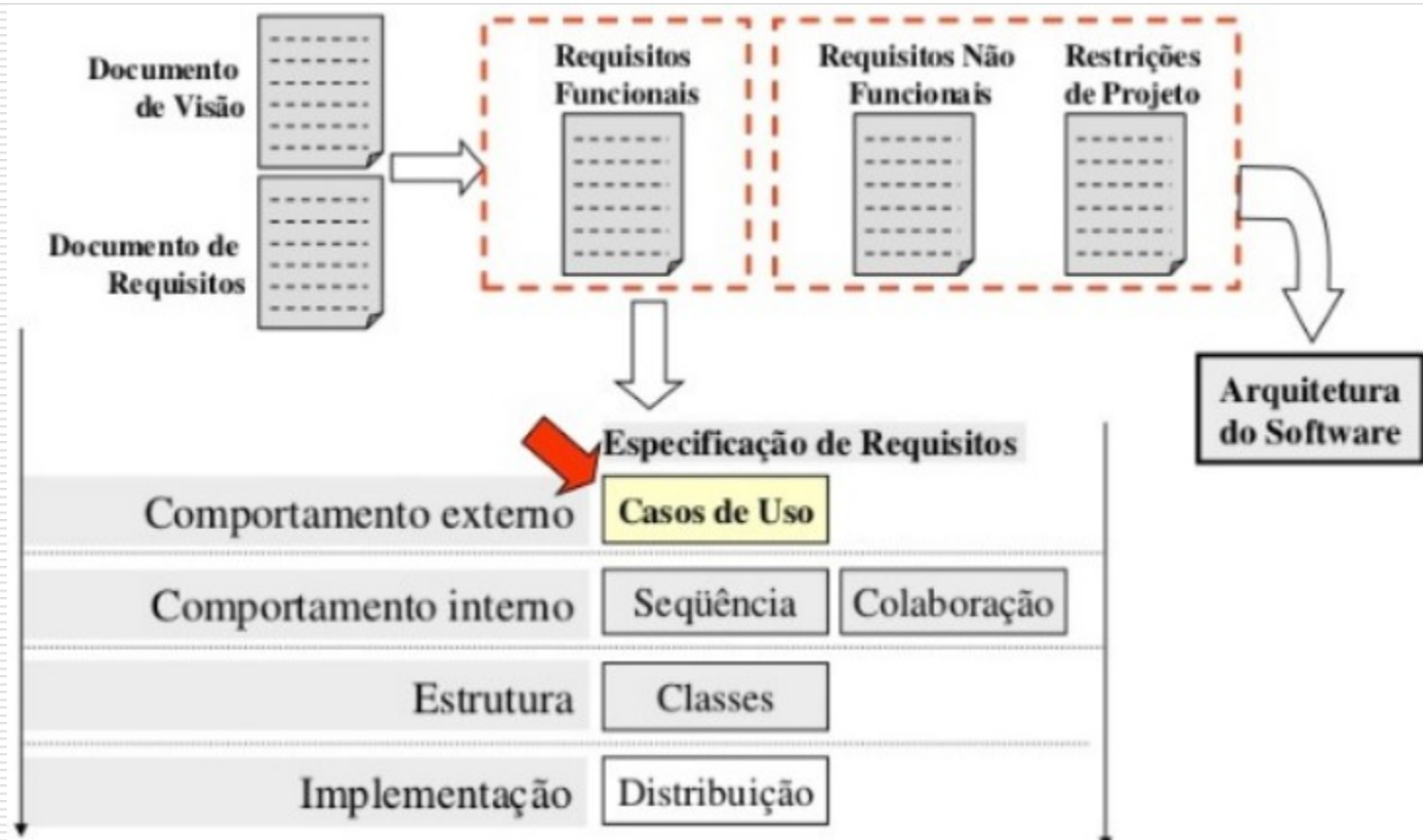
3. REQUISITOS (Específicos)

Podem documentar interfaces externas, descrever funcionalidade e desempenho do sistema, especificar requisitos lógicos de banco de dados, restrições de projeto, características de qualidade, etc.

4. APÊNDICE

5. ÍNDICE

Especificação de Requisitos



Especificação

- A formalidade e formatos da especificação varia com o tamanho e a complexidade do software a ser construído.
 - Para **sistemas grandes**,
 - documentos escritos, linguagem de descrição e modelos gráficos podem ser a melhor abordagem.
 - Para **pequenos sistemas ou produtos**,
 - utilizam-se cenários
-

Negociação

- Chegar a um acordo sobre um sistema de entrega que é realista para desenvolvedores e clientes
 - Requisitos são categorizados e organizados em subconjuntos
 - As relações entre os requisitos identificados
 - Requisitos revisados para correção
-

Negociação

- Requisitos são priorizados com base nas necessidades dos clientes
 - Negociação sobre requisitos, custo do projeto e cronograma do projeto
 - A negociação deve ser aceita por ambas as partes
-

Validação de Requisitos

- Mecanismo de revisão técnica formal que procura
 - Erros no conteúdo ou de interpretação;
 - Áreas onde podem ser necessários esclarecimentos;
 - Falta de informações;
 - Inconsistências (um problema maior quando sistema ou produtos grandes são projetados);
 - Requisitos conflitantes ou não realísticos (inalcançável).
-

Elicitação e Análise de Requisitos

Gerenciamento de Requisitos

Gestão de requisitos é importante para a descoberta de requisitos

Gerenciamento de Requisitos

- Um conjunto de atividades que ajuda a equipe do projeto a identificar, controlar e encontrar requisitos e mudanças à medida que o projeto prossegue.
 - Requisitos iniciam com a **identificação**.
 - Cada requisito é atribuído com um identificador único. Uma vez identificados os requisitos, a **tabela de rastreabilidade** são desenvolvidas.
-

Gerenciamento de Requisitos

■ **Tabela de Rastreabilidade**

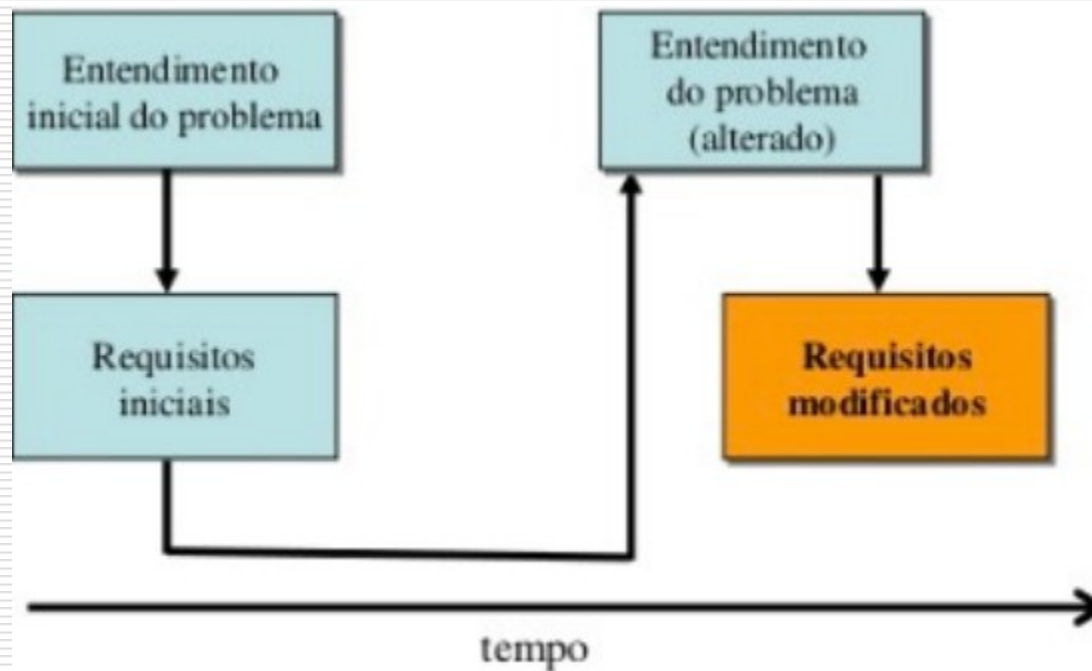
- **Tabela de recursos de rastreabilidade** – mostra características observáveis sobre requisitos relacionados ao cliente
- **Tabelas rastreáveis das fontes** – identificam fontes de cada requisito
- **Tabela da rastreabilidade das dependências** – indicam relações entre os requisitos
- **Tabela de Rastreabilidade dos Subistema** – requisitos são categorizados por subsistemas
- **Tabela de Rastreabilidade da Interface** – apresenta requisitos relacionados para interfaces internas e externas

- A tabela irá ajudar a controlar, se a mudança em um requisito afetará os diferentes aspectos do sistema.
-

Gerenciamento de Requisitos

- É o processo de gerenciar as mudanças nos requisitos durante o processo de Engenharia de Requisitos.
 - Os requisitos são, inevitavelmente, incompletos e inconsistentes.
 - Novos requisitos surgem à medida que as necessidades de negócios mudam e há um melhor entendimento do sistema.
 - Diferentes pontos de vista normalmente têm requisitos diferentes (e conflitantes).
 - Diante disso é possível afirmar que os requisitos sempre mudam, bem como a prioridade de cada um ao longo do projeto
-

Gerenciamento de Requisitos



Gerenciamento de Requisitos

- ❑ Considerando que o ambiente de negócio e tecnológico do projeto mudam durante seu desenrolar é necessário gerenciar tudo isso.
 - ❑ Durante o processo de engenharia de requisitos, será necessário planejar:
 - A identificação dos requisitos: Como os requisitos são individualmente identificados;
 - Um processo de mudança de gerenciamento;
 - Políticas de rastreabilidade: Manter o histórico dos requisitos, rastrear suas mudanças e seus possíveis impactos;
 - Suporte de ferramenta: Necessário para auxiliar no processo de gerenciamento.
-

Rastreabilidade

- A rastreabilidade preocupa-se com as relações entre requisitos, suas fontes e o projeto do software.
 - Rastreabilidade de Fonte: Ligação entre o requisito e o stakeholder que o propôs (e sua necessidade original);
 - Rastreabilidade de Requisitos: Ligações entre requisitos que dependem entre si;
 - Rastreabilidade de Projeto: Ligação entre o requisito e o projeto (arquitetura, módulos, código) do software.
-

Matrizes de Rastreabilidade

- São geralmente utilizadas para exibir os relacionamentos entre a elicitação de requisitos e a representação destes requisitos em um método particular da engenharia de software.
 - Mesmo em projetos pequenos ou de tamanho moderado, o estabelecimento de elos de rastreabilidade, entre artefatos-chave e modelos, continua sendo uma tarefa desafiadora e cara entre artefatos-chave e modelos, continua sendo desafiadora e cara.
-

Matrizes de Rastreabilidade

- Elas exibem o mapeamento entre fonte e alvo.
 - Tais mapeamentos são apresentados em um tipo especial de matriz, chamada de matriz de dependência, que representa a relação de dependência entre elementos da fonte e do alvo (fonte x alvo)
-

Matrizes de Rastreabilidade

- Nas linhas, ficam os elementos fonte, e nas colunas, os elementos alvo. Nesta matriz, uma célula com o valor 1 denota que o elemento fonte (na linha) é mapeado para o elemento alvo (na coluna).
- Reciprocamente, isto significa que o elemento alvo depende do elemento fonte
- Em sua forma mais simples, a rastreabilidade se manifesta em tabelas cruzadas, nas quais os elementos de um projeto são relacionados aos requisitos que satisfazem
- Nesta matriz, elementos-fonte são mapeados para elementos-alvo.

		Alvos			
		a1	a2	a3	a4
Fontes	f1	1	0	1	1
	f2	0	1	0	0
	f3	0	0	1	0

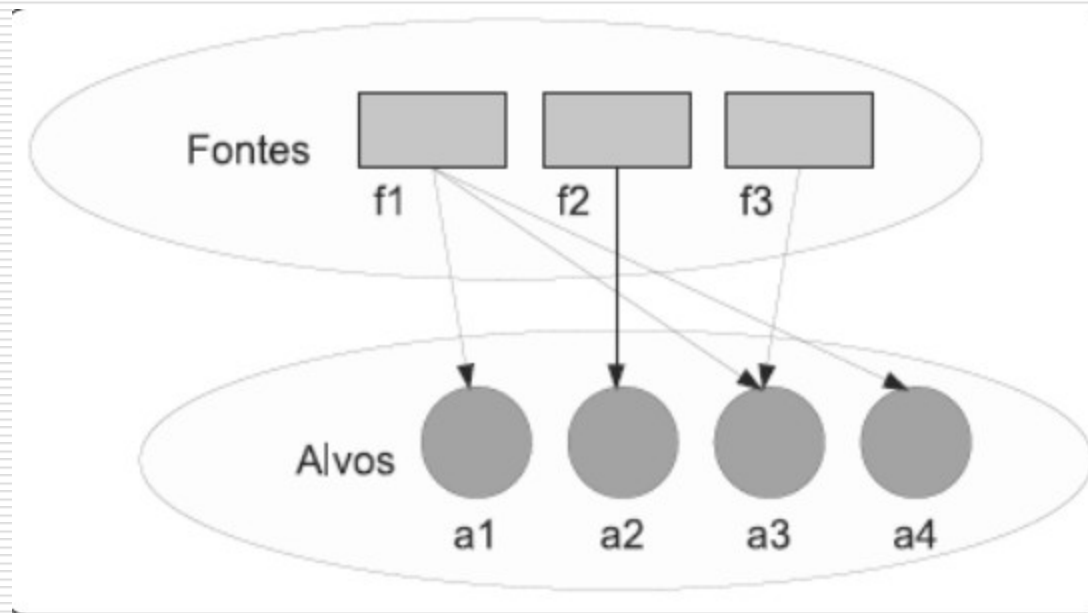
Matrizes de Rastreabilidade

- ❑ A Tabela apresenta o mapeamento do elemento-fonte **f1** para os elementos-alvo **a1**, **a3** e **a4**, o que indica que **a1**, **a3** e **a4** dependem de **f1**. Analogamente, pode-se dizer que **f1** dá origem a **a1**, **a3** e **a4**.
- ❑ Esta representação permite visualizar, por exemplo, que vários requisitos são implementados por uma mesma classe, permitindo que a intersecção entre classes responsáveis pela satisfação de um requisito seja não-vazia (ou seja, *n* classes podem ser comuns entre dois ou mais requisitos)

		Alvos			
		<i>a1</i>	<i>a2</i>	<i>a3</i>	<i>a4</i>
Fontes	<i>f1</i>	1	0	1	1
	<i>f2</i>	0	1	0	0
	<i>f3</i>	0	0	1	0

Matrizes de Rastreabilidade

- É possível visualizar as várias classes que podem ser necessárias para a implementação de um requisito, podendo ser representada graficamente, conforme a Figura a seguir.



Matrizes de Rastreabilidade

- ❑ Elos de rastreabilidade são geralmente estabelecidos pelo relacionamento explícito entre dois artefatos, armazenando-os em tabelas, bancos de dados, ou ferramentas de gerenciamento de requisitos, e esta ainda é a prática atual.
 - ❑ Existem diversos modelos de rastreabilidade que diferem nos tipos de artefatos que rastreiam.
 - ❑ Porém, a maioria dos projetos não utiliza um método sistemático de rastreamento, mas delega, para determinados indivíduos, a atividade de realizar um rastreamento ad hoc, quando for necessário
-

Matrizes de Rastreabilidade

- Formalmente, as estruturas tradicionais de rastreabilidade são criadas seguindo a definição $Elo(a, a')$.
 - Uma matriz de rastreabilidade (MR) é formada pelo conjunto $\{Elo(a, a')\}$, **a** e **a'** são artefatos, em qualquer nível de abstração
 - (eg. caso de uso, especificação de requisitos, diagrama de classe, código fonte de uma linguagem específica, etc.
 - Cada artefato possui um nível de abstração, que pode ser mais ou menos completo do que os outros, e apresenta diferentes pontos de vista sobre as necessidades do sistema.
-

Matrizes de Rastreabilidade

- As relações entre estes, do menos completo para o mais completo, até a implementação, deve ser mantida para garantir que a estrutura de rastreabilidade permita:
 - O mapeamento dos requisitos para um modelo posterior, até o código fonte, garantindo que a satisfação dos requisitos esteja atribuída a componentes do sistema (Forward from Requirements);
 - O Mapeamento de um modelo qualquer de volta para os requisitos, evitando o gold-plating (Backward to Requirements);
-

Matrizes de Rastreabilidade

- Grande parte das propostas de rastreabilidade possui estas características, no entanto:
 - Algumas suportam o rastreamento somente vertical (relacionamentos entre artefatos no mesmo nível de abstração),
 - Outras somente o horizontal (relacionamentos entre artefatos em níveis diferentes de abstração).
-

Matrizes de Rastreabilidade

- Para exemplificar a utilização de matrizes no rastreamento vertical, considere as seguintes regras de negócio (requisitos):
 - **R1** → “Todo cliente tem uma conta, com um determinado limite de crédito”;
 - **R2** → “O limite de crédito de qualquer conta está restrito a um determinado valor, estabelecido segundo a lei federal XYZ”;
 - **R3** → “A verificação do limite de crédito é realizada de acordo com os seguintes critérios...”;
 - **R4** → “Universitários têm limite de R\$ 200,00, invariavelmente”.
-

Matrizes de Rastreabilidade

- Não é necessário muito esforço de interpretação para perceber que os requisitos estão de fato relacionados, pois tratam de interesses comuns, formando a possível matriz de rastreabilidade, representada pela seguinte Tabela:

Requisitos	R1	R2	R3	R4
R1		X		X
R2			X	
R3				X
R4				

- Note que apesar do pequeno número de requisitos e relacionamentos, não é trivial interpretar a matriz da Tabela e identificar o motivo por trás da existência dos elos e a inexistência de outros. No caso do exemplo, estes foram criados pelas seguintes interpretações:
-

Matrizes de Rastreabilidade

- ❑ **R1-R2** → “A lei federal XYZ estabelece o limite da conta, citado em R1”;
- ❑ **R1-R4** → “O limite da conta também pode ser definido pelo valor citado em R4”;
- ❑ **R2-R3** → “A verificação deve considerar as regras impostas pela legislação”;
- ❑ **R3-R4** → “A verificação deve considerar o caso das contas universitárias”.

Requisitos	R1	R2	R3	R4
R1		X		X
R2			X	
R3				X
R4				

Matrizes de Rastreabilidade

- ❑ Porém, várias interpretações diferentes podem ser tomadas a partir dos requisitos apresentados, formando relacionamentos diferentes.
- ❑ Assim, para evitar este esforço de re-interpretação, é necessário anexar a cada elo as decisões, problemas, suposições e argumentos que motivaram sua criação.
- ❑ Analogamente, é fundamental que as mesmas informações sejam mantidas para os elos que não foram criados, evitando o esforço desnecessário de justificar novamente uma decisão já estabelecida

Requisito	Classes	Métodos	Atributos
R1	Cliente, Account		Cliente.account.limit
R2	Account		Account.limit
R3		System.limit.Verification	
R4	Account		Account {limit, type}

Matrizes de Rastreabilidade

- A Tabela representa as possíveis relações entre os requisitos e partes do código apresentado.
- O requisito **R1** é satisfeito pela implementação das classes **Client** e **Account**, bem como pelo atributo limit em **Account**. **R2** trata do limite da conta, então a implementação de **Account.limit** pode ser afetada caso **R2** seja alterado (e vice-versa).
- Já **R3** é satisfeito pela implementação de **System.limitVerification**, e **R4** pelo tratamento de um tipo de conta (**Account.type**, que pode ser “conta de pessoa física”, “conta de pessoa jurídica”, “conta universitária”, etc.).
- Note que a matriz de rastreabilidade depende muito da interpretação humana, sendo que outras interpretações e necessidades podem originar uma matriz completamente diferente.

Matrizes de Rastreabilidade

- ❑ Embora a rastreabilidade seja legalmente requerida na maioria das aplicações de segurança crítica, e seja reconhecida como um componente de muitas iniciativas de melhoria em processos de software,
 - ❑ As organizações ainda lutam para implementá-la de modo que ofereça um bom custo-benefício
 - ❑ Padrões aceitos como o CMMI nível 3 e ISO 15504 exigem que práticas básicas de rastreabilidade sejam utilizadas,
-

Matrizes de Rastreabilidade

- ❑ Mas as atividades de rastreamento e gerenciamento de equisitos podem trazer diversos custos inesperados, que em alguns casos excedem os benefícios.
 - ❑ Infelizmente, a tarefa de construir uma matriz de **RR** leva tempo, é árdua, cara, e sujeita a erros. Por conta disto, com o desenvolvimento do software e sua manutenção, geralmente as informações de rastreabilidade não representam mais a realidade, ou nem existem mais.
-

Iniciando Requisitos na Engenharia de Processos

□ **Identificando Interessados**

- Interessado pode ser "qualquer um que se beneficia de uma forma direta ou indireta a partir do sistema que está sendo desenvolvido".

Ex. Gerente de Negócios, gerente de projetos, pessoal de marketing, engenheiro de software, engenheiro de suporte, usuários finais, clientes internos e externos, etc

- Cada um destes tem uma visão diferente do sistema

□ **Reconhecendo múltiplos pontos de vistas**

- O pessoal do Marketing se preocupa com recursos e funções para analisar mercados em potencial
- Gerente de negócios se preocupa com recursos construídos dentro do orçamento e se estarão prontos para atender o mercado.
- Usuário Final - Facilidade de aprender e utilizar
- Engenheiro de Software - Produto funcionando com várias infraestruturas de suporte
- Engenheiro de Suporte - Manutenibilidade do software.

- Papel do RE é categorizar todas as informações dos interessados de uma forma que não poderia haver nenhuma exigência inconsistente ou conflitantes entre si
-