

Levantamento, Análise e Gestão Requisitos

Aula 05



ENGENHARIA
DE REQUISITOS

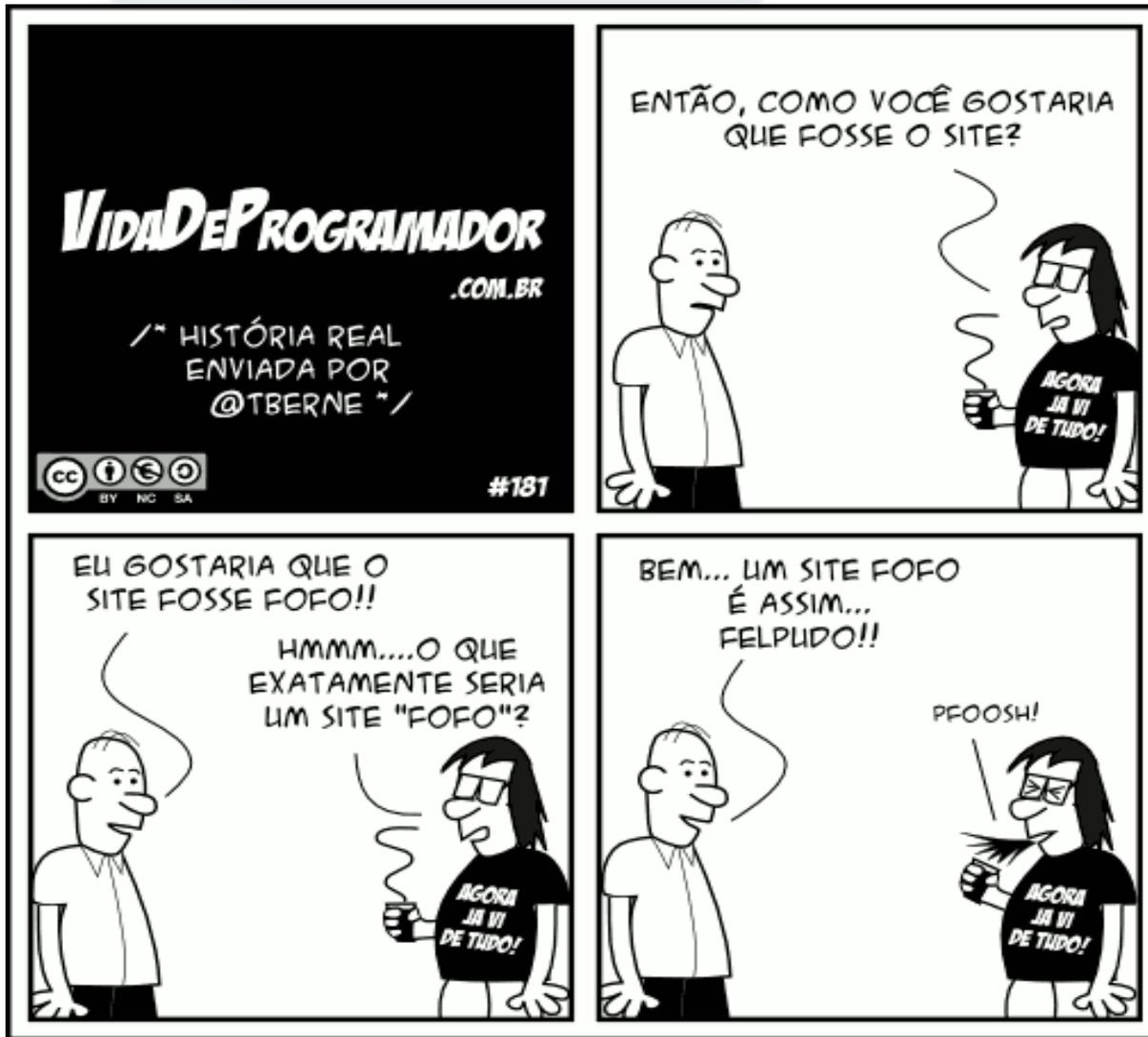


Programadores {+ Inovadores ;}
www.x25.com.br

- Requisitos de Software
 - Tipos de Requisitos: funcionais e não-funcionais
 - Definição do escopo do problema
 - Análise do problema
 - Compreensão da necessidade dos envolvidos
 - Definição do Problema
 - Identificação dos atores envolvidos
 - Requisitos do Sistema
 - Requisitos do Usuário



Tipos de Requisitos



Engenharia de Requisitos

- Processo para estabelecer os serviços que o cliente requer de um sistema
- Restrições sob as quais um sistema deve ser desenvolvido e executado
- Descrições das funções e restrições



O que é um requisito?

- Pode variar:
 - De uma declaração abstrata de alto nível
 - De uma função que o sistema deve fornecer
 - De uma restrição do sistema, a uma definição detalhada, matematicamente formal
 - De uma função do sistema
- Requisitos possui diferentes níveis de descrição
 - Base para uma licitação de um contrato deve ser aberto à interpretações
 - Base para o próprio contrato deve ser definido em detalhes
 - Ambas declarações podem ser chamada requisitos



Por que os requisitos são importantes?

- Pesquisa foi realizada em:
 - Mais de 350 empresas sobre os seus mais de 8.000 Projetos de software
 - 30% desses projetos foram cancelados
- Dos concluídos:
 - 9% entregues dentro do prazo e do valor estimado (Standish Group – 1994)



Fatores Principais de Falhas em Projetos

Fatores principais relatados como as causas das falhas:

1. Requisitos incompletos (13,1%)
2. Falta de envolvimento por parte do usuário (12,4%)
3. Falta de recursos (10,6%)
4. Expectativas não realistas (9,9%)
5. Falta de apoio dos executivos (9,3%)
6. Modificações nos requisitos e nas especificações (8,7%)
7. Falta de planejamento (8,1%)
8. O sistema não era mais necessário (7,5%)



Classificação dos Requisitos

- Entendemos requisitos de software como sentenças que expressam as necessidades dos clientes e que condicionam a qualidade do software
- Classificamos requisitos como:
 - **Requisitos Funcionais** – diretamente ligado a funcionalidade do software
 - **Requisitos Não-funcionais** – reflete os requisitos que expressam restrições que o software deve atender ou qualidade específicas que o software deve ter
- Um terceiro grupo é denominado “Requisitos Inversos” que trata de situações que não devem ocorrer



Requisitos Funcionais e Não-Funcionais

- Requisitos funcionais
 - Declarações de serviços que o sistema deve fornecer
 - Como o sistema deve reagir a entradas específicas
 - Como deve se comportar em determinadas situações
- Requisitos não funcionais
 - Restrições sobre os serviços
 - As funções oferecidas pelo sistema
- Requisitos de domínio
 - Requisitos que se originam do domínio da aplicação do sistema e que refletem características desse domínio
 - Podem ser requisitos funcionais e não funcionais



Requisitos Funcionais

- Descrever as funcionalidades do sistema desejadas pelos clientes (o que ele faz)
- O QUE esperar que o software realize
- Descrever a funcionalidade ou os serviços do sistema
- Depender do tipo de software, das expectativas dos usuários e do tipo de sistema que está sendo desenvolvido
- Podem ser dos seguintes tipos:
 - Do usuário, descritos de forma bem geral
 - De sistema, descrevem a função de sistema detalhadamente



Exemplos de Requisitos Funcionais

- Possibilitar o cadastramento dos dados pessoais dos clientes
- Emitir relatórios gerenciais
- Permitir a baixa automática do estoque quando da venda de um produto
- Possibilitar o cálculo dos gastos diários, semanais, mensais e anuais com pessoal
- Emitir relatórios de compras a cada quinze dias
- Obter o número de aprovações, reprovações e trancamentos em todas as disciplinas por um período de tempo



Requisitos Não-Funcionais

- Descrever as qualidades do sistema (como ele é)
- A qualidade afeta diretamente a satisfação do cliente e envolvidos com o sistema
- A qualidade de um software pode ser avaliada de duas maneiras
- A qualidade visível para o usuário final e a qualidade interna visível em tempo de desenvolvimento (mas que permite ou não evoluções do software)



Exemplos de Requisitos Não-Funcionais

- Tempo de resposta não deve ultrapassar 10 segundos
- Software deve ser operacionalizado no sistema Windows
- Banco de dados usado será o PostgreSQL
- Banco de dados protegido para o acesso apenas de usuários autorizados
- Tempo de desenvolvimento não deve ultrapassar seis meses

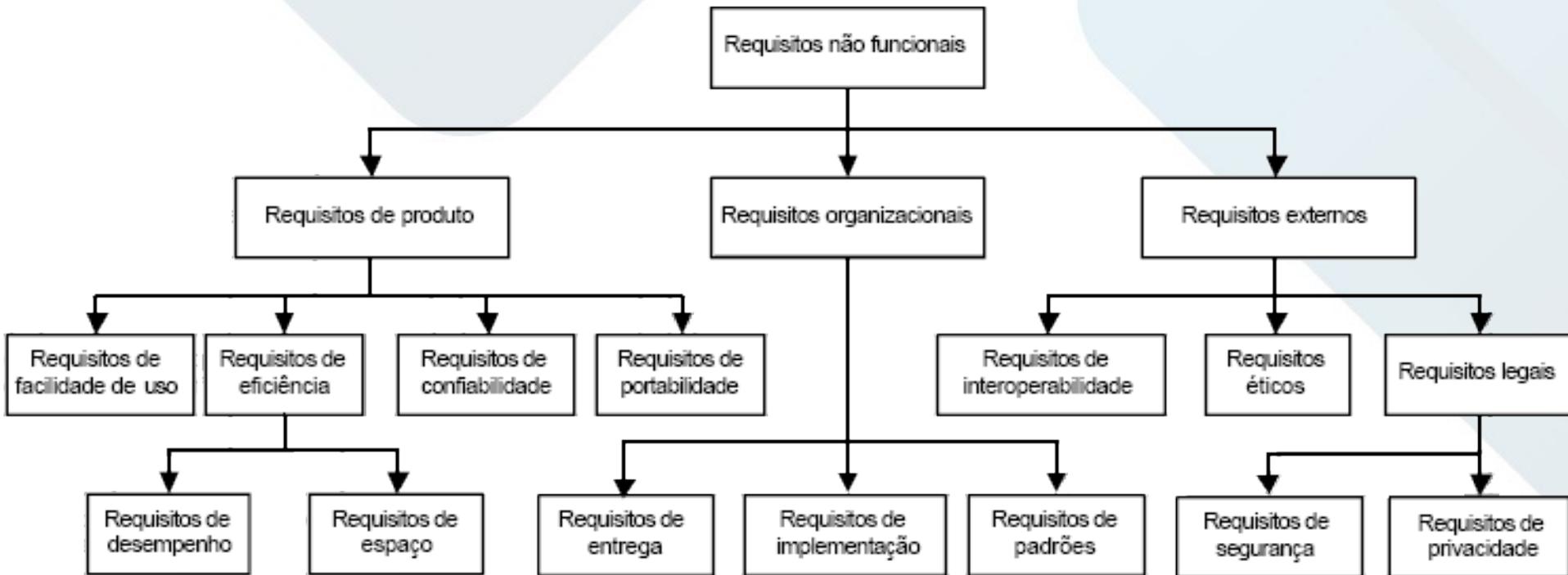


Tipos de Requisitos Não-Funcionais

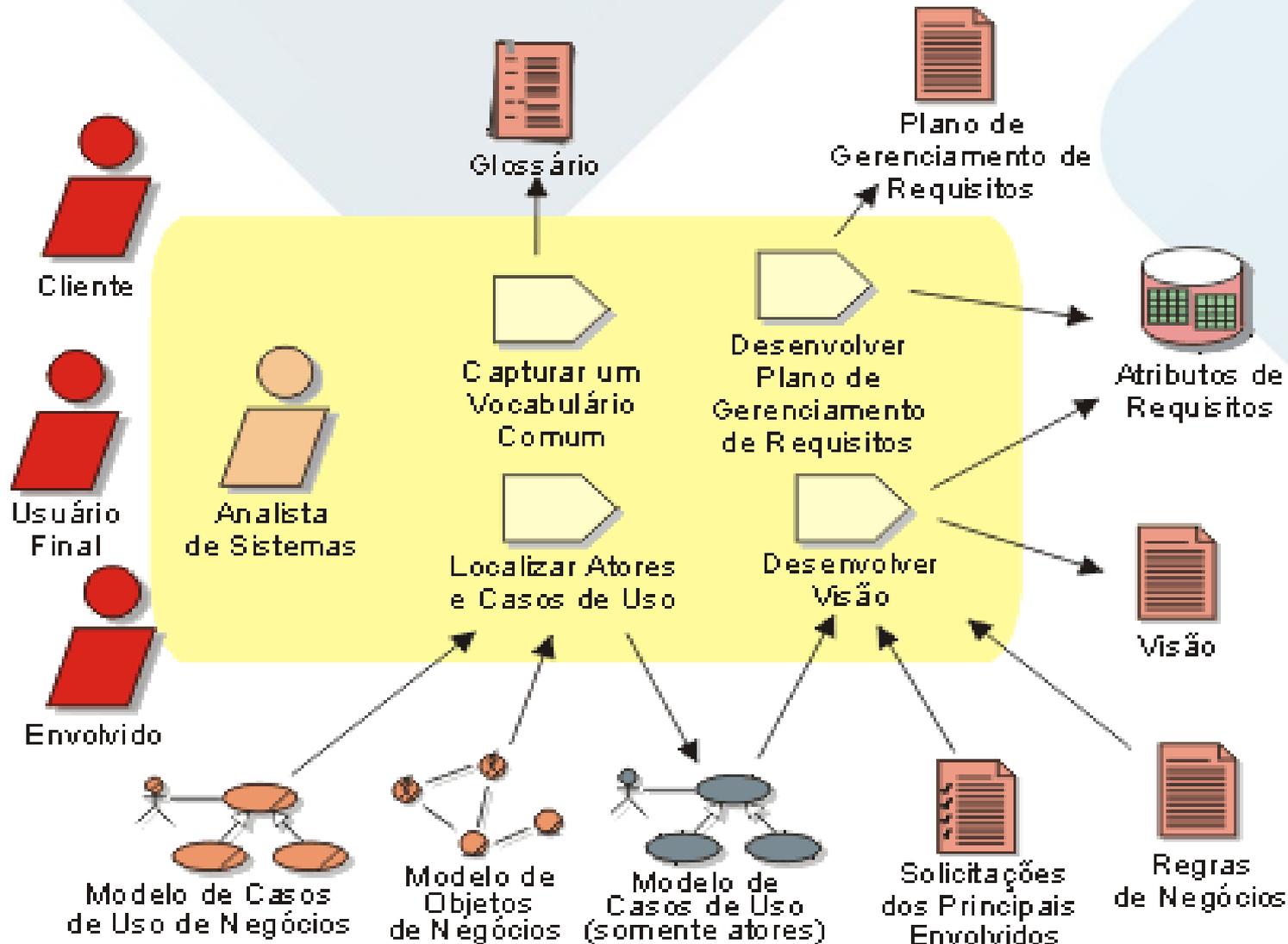
- Requisitos de produto
 - Especificam o comportamento do produto
 - Ex: velocidade de execução, confiabilidade, portabilidade, facilidade de uso
- Requisitos organizacionais
 - Consequência de políticas de procedimentos nas organizações do cliente e do desenvolvedor
 - Ex: padrões de processos que devem ser utilizados, requisitos de implementação
- Requisitos externos
 - Procedentes de fatores externos ao sistema e a seu processo de desenvolvimento
 - Ex: requisitos de interoperabilidade, requisitos legais e os requisitos éticos



Tipos de Requisitos Não-Funcionais



Análise do Problema



Análise do Problema

- A finalidade desse detalhamento do fluxo de trabalho é:
 - Estabelecer acordo sobre o problema a ser resolvido
 - Identificar os envolvidos
 - Definir as fronteiras do sistema
 - Identificar as restrições impostas ao sistema
- O primeiro passo é verificar se todas as partes envolvidas estão de acordo com a definição do problema que tentamos resolver com o nosso sistema

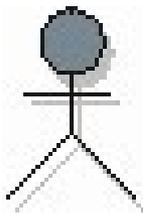


Análise do Problema

- Para evitar mal-entendidos:
 - Concordar com uma terminologia comum a ser usada no decorrer do projeto
 - Definir os termos de nosso projeto em um glossário que será mantido durante todo o ciclo de vida do projeto
 - Compreender completamente o(s) problema(s) tratado(s) e conhecer os envolvidos
 - Alguns desses envolvidos serão representados por atores em nosso Diagrama de Casos de Uso



- Elemento externo que interage com o sistema:
 - “externo”: atores não fazem parte do sistema
 - “interage”: um ator troca informações com o sistema
- Casos de uso representam uma sequência de interações entre o sistema e o ator
- No sentido de troca de informações entre eles
- Normalmente um agente externo inicia a sequência de interações com o sistema, ou um evento acontece para que o sistema responda



Representação Gráfica da UML

Categorias de Atores

- **Pessoas**
 - Empregado, Cliente, Gerente, Almoxarife, Vendedor, ...
- **Organizações**
 - Empresa Fornecedora, Agência de Impostos, Administradora de Cartões, ...
- **Outros sistemas**
 - Sistema de Cobrança, Sistema de Estoque de Produtos, ...
- **Equipamentos**
 - Leitora de Código de Barras, Sensor, ...



Como Definir um Ator

- Papel representado em relação ao sistema:
 - **Cliente** que compra mercadorias
 - **Vendedor** que processa vendas
 - **Funcionário** de uma instituição bancária que realiza a manutenção de um caixa eletrônico
 - **Banco** que interage com um cliente
- O nome de um ator deve fazer referência ao seu papel, ao invés de quem ele representa



Identificação dos Atores

- Perguntas úteis:
 - Que órgãos, empresas ou pessoas irão utilizar o sistema?
 - Que outros sistemas irão se comunicar com o sistema a ser construído?
 - Alguém deve ser informado de alguma ocorrência no sistema?
 - Quem está interessado em um certo requisito funcional do sistema?
- O desenvolvedor deve ainda continuar a pensar sobre atores quando passar para a identificação dos casos de uso



Dicas de Identificação dos Atores

- Você localizou todos os atores? Ou seja, considerou e modelou todos os papéis no ambiente do sistema?
- Cada ator está envolvido com pelo menos um caso de uso?
- Você pode apontar pelo menos duas pessoas que poderiam atuar como um ator específico?
- Existem atores que desempenham papéis similares em relação ao sistema?
- Existem dois atores executando o mesmo papel em relação a um caso de uso?
- Um determinado ator usará o sistema de diversas maneiras ou ele tem várias finalidades para utilizar o caso de uso?
- Tanto os usuários quanto os clientes entendem os nomes?
- Os atores têm nomes intuitivos e descritivos?



Tipos de Especificações dos Requisitos

- **Requisitos do Sistema**

- Documento estruturado que estabelece detalhadamente as funções e as restrições de sistema
- Escrito como um contrato entre o cliente e o desenvolvedor do software

- **Requisitos do Usuário**

- Declarações em linguagem natural e/ou diagramas sobre as funções que o sistema deve fornecer e as restrições sob as quais deve operar

- **Especificação do Software**

- Descrição detalhada do software que serve como base para projeto e a implementação
- Escrito para os desenvolvedores



Tipos de Especificações dos Requisitos

Requisitos do Sistema

Usuários finais de sistemas
Engenheiros do cliente
Arquitetos de sistemas
Desenvolvedores de software

Requisitos do Usuário

Gerentes de clientes
Usuários finais de sistemas
Engenheiros do cliente
Gerentes do fornecedor
Arquitetos de sistemas

Especificação do Software

Engenheiros do cliente (talvez)
Arquitetos de sistemas
Desenvolvedores de software



Tipos de Especificações – Requisitos do Sistema

- Descrevem mais detalhadamente os requisitos de usuário
- Base para um contrato de implementação do sistema
- Problemas na identificação dos requisitos de sistema:
 - Requisitos de sistema ainda são descritos em linguagem natural acompanhada de diagramas ilustrativos (a ambiguidade persiste)
 - Idealmente, os requisitos de sistema não devem conter decisões de projeto, mas requisitos impostos pela arquitetura e sistemas legados acabam sendo incorporados nos requisitos de sistema



Tipos de Especificações – Requisitos do Usuário

- Especificam o comportamento externo do sistema sob a perspectiva do usuário (humano ou não)
- Problemas na identificação dos requisitos de usuário:
 - Falta de clareza ou ambiguidades, por serem descritos em Linguagem natural (ex.: o usuário deve ser alertado sobre operações perigosas)
 - Confusão entre requisitos funcionais, não funcionais e objetivos do sistema (ex.: o sistema deve facilitar a solicitação de declarações)
 - Fusão de requisitos onde um único requisito é na verdade uma condensação de vários requisitos (Ex.: O sistema deve permitir ao usuário escolher a imagem a ser processada (dentre os diversos formatos permitidos))



Tipos de Especificações – Exemplos

- Requisitos de Sistema

- O usuário deve dispor de recursos para definir o tipo dos arquivos externos
- Cada tipo de arquivo externo pode ter uma ferramenta associada que pode ser aplicada a ele
- Cada tipo de arquivo externo pode ser representado como um ícone específico na tela do usuário
- Devem ser fornecidos recursos para o ícone que representa um arquivo externo, a ser definido pelo usuário
- Quando um usuário seleciona um ícone que representa um arquivo externo, o efeito dessa seleção é aplicar a ferramenta associada com o tipo de arquivo externo ao arquivo representado pelo ícone selecionado

- Requisitos do usuário

- O software deve oferecer um meio de representar e acessar arquivos externos criados por outra ferramenta



- A Norma ISO / IEC 9126 define seis características de qualidade de software que devem ser avaliadas:
 - Funcionalidade (finalidade do produto)
 - Usabilidade (esforço para utilizar, aprender o produto)
 - Confiabilidade (frequência de falhas, recuperabilidade)
 - Eficiência (característica relacionada ao desempenho)
 - Manutenibilidade (esforço necessário para modificar)
 - Portabilidade (capacidade de transferir o produto para outros ambientes)



- Uma boa especificação de requisitos deve ser:
 - Clara e não-ambígua
 - Completa
 - Correta
 - Compreensível
 - Consistente
 - Concisa
 - Confiável



Documento de Requisitos – Padrão IEEE 830/1998

Sugere a seguinte estruturação para o documento de requisitos:

1. Introdução

- 1.1. Propósito do documento de requisitos
- 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

Apêndices

Índice



Documento de Requisitos – Padrão IEEE 830/1998

O padrão IEEE 830-1998 sugere 7 formas de organizar o item 3 do documento de requisitos. Cada forma organiza os requisitos por um critério específico:

1. Modo (de operação, utilização, etc.)
2. Classe de Usuário
3. Objeto
4. Facilidade
5. Estímulo
6. Hierarquia Funcional
7. Múltiplos Critérios (Ex.: facilidade + classe de usuário)



Documento de Requisitos – Padrão IEEE 830/1998

3. Requisitos Específicos

3.1. Requisitos de interfaces externas

3.1.1. Interfaces com o usuário

3.1.2. Interfaces de hardware

3.1.3. Interfaces de software

3.1.4. Interface de comunicação

3.2. Requisitos Funcionais

3.2.1. Critério #1

3.2.1.1. Requisito funcional #1

3.2.1.k. Requisito funcional #k

3.2.2. Critério #n

3.3. Requisitos de desempenho

3.4. Restrições de projeto

3.5. Atributos do sistema de software

3.6. Outros Requisitos



Questionário Prático

1. Quantas pessoas vão utilizar o software? Desse número, quantas utilizarão simultaneamente?
2. Dos relatórios previstos, quais podem ser gerados por processamento batch e quais devem ser online? Qual o tempo aceitável para processar e gerar um relatório online?
3. Qual o tempo de resposta esperado para as principais funcionalidades do sistema? E para as outras?
4. Qual tipo de acesso a aplicação vai ter? Somente via intranet? Internet?
5. Qual o perfil dos usuários que vão acessar a aplicação? Possuem conhecimento de internet? São usuários avançados?
6. É desejável que a maior parte das funcionalidades da aplicação possam se acessadas via teclado (sem auxílio do mouse)?
7. A aplicação deve ser compatível com quais versões do navegador? sistema operacional?
8. Quais os padrões de implementação esperados? Os desenvolvedores podem escrever o código em qualquer idioma? Podem utilizar qualquer banco de dados e qualquer tecnologia?
9. Qual a segurança esperada para o tráfego de dados? Toda comunicação entre o servidor e o navegador tem que ser criptografada usando SSL? Será adquirido o certificado SSL? Ou a aplicação não tem dados críticos e confidenciais vai ser executada em uma rede segura?
10. Qual a disponibilidade a aplicação deve ter? O tempo médio entre falhas, tempo máximo para acertar os problemas? Número máximo de problemas em cada versão? Nesse caso a resposta pode ser que aplicação deve obedecer um acordo de SLA ou que existem regras específicas para esse software de acordo com o negocio.



Dúvidas? Agradecimentos

Home Page

<http://fernandoans.site50.net>

Blog

<http://fernandoanselmo.blogspot.com>

X25 Home Page

<http://www.x25.com.br>



Fernando Anselmo

fernando.anselmo@x25.com.br



Programadores {+ Inovadores ;}

www.x25.com.br