

Engenharia de requisitos

Um **Requisito** é uma característica que um sistema precisa ter ou uma restrição que ele precisa satisfazer para ser aceito pelo cliente.

A **Engenharia de requisitos** tem por objetivo definir os requisitos de um sistema em construção e apresenta duas atividades principais: o levantamento e análise de requisitos.

Levantamento de requisitos

O levantamento de requisitos é desafiador. Ele exige a colaboração de vários participantes com diferentes experiências. Em geral os clientes e usuários são especialistas em seu domínio e têm uma idéia geral do que o sistema deverá ser, mas têm pouca ou nenhuma experiência no desenvolvimento de aplicações.

Erros introduzidos durante o levantamento de requisitos são tardiamente encontrados e têm um grande custo para serem corrigidos.

Tipos de requisitos

Requisitos funcionais descrevem as interações do sistema com o seu ambiente independente de sua implementação, incluindo as interações com os usuários e outros agentes externos.

Em geral os requisitos funcionais estão associados a:

1. Funcionalidades do sistema
2. Serviços que o sistema deve prover
3. Comportamento do sistema a determinadas entradas
4. Funções que o sistema não deve suportar

Exemplos: a) O aplicativo deve possibilitar o agendamento de pagamentos em conta corrente (em um sistema de bancário de contas correntes). b) O aplicativo deve possibilitar o débito parcial da fatura do cartão e pagamento antecipado. c) O pagamento *on-line* não será permitido para pagamentos em atraso.

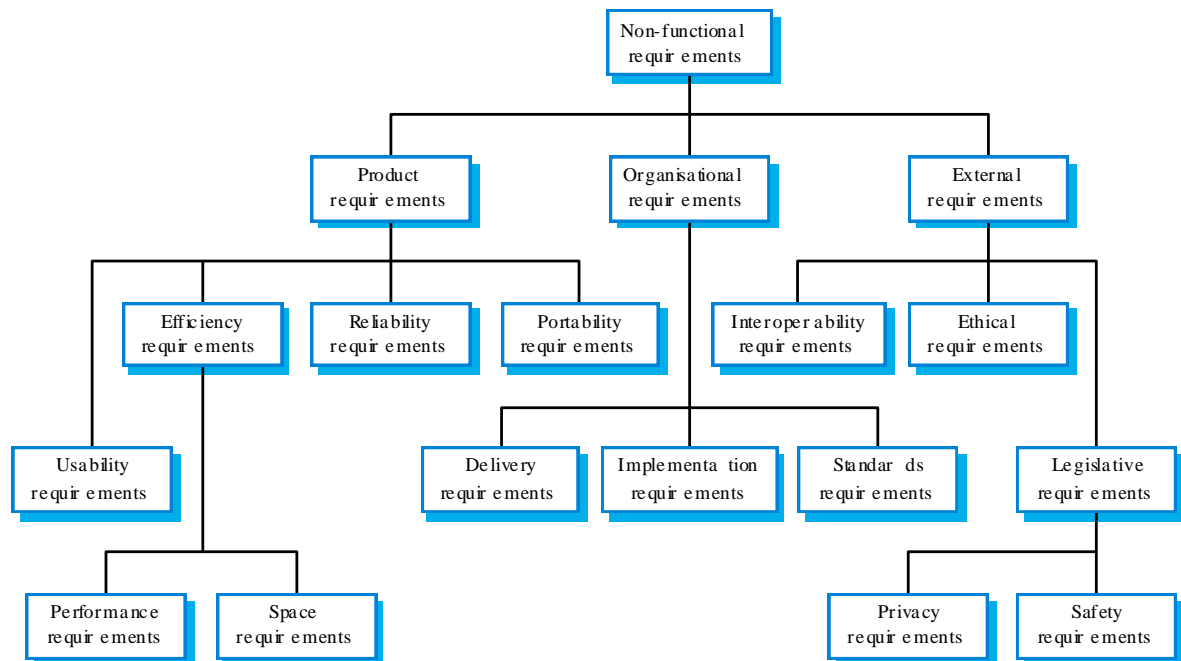
Requisitos não funcionais são características requeridas pelo usuário do sistema, mas que não estão diretamente relacionadas à sua funcionalidade.

Os requisitos não funcionais são associados, por exemplo, a:

1. Qualidade
2. Desempenho
3. Portabilidade
4. Precisão
5. Confiabilidade
6. Segurança

Os requisitos não funcionais definem, em geral, restrições aos sistemas propostos. Muitas dessas restrições refletem restrições do usuário ao processo de desenvolvimento do software como restrições organizacionais, restrições orçamentárias, legais etc.

Os requisitos não funcionais ainda podem ser classificados como **Requisitos de produto, organizacionais e externos**.



Requisitos não funcionais segundo Sommerville.

Requisitos de domínio são características que refletem um dado domínio da aplicação. Considere um sistema de solução de sistemas lineares, pode haver o requisito de solução do sistema através de um método específico, como Gauss-Seidel.

Pseudo Requisitos. Alguns autores ainda consideram a restrição pelo cliente do tipo de linguagem ou plataforma a ser utilizada como um pseudo-requisito do aplicativo.

Validação dos requisitos

A constante validação dos requisitos junto ao cliente e usuários é crucial. A validação envolve verificar se a especificação dos requisitos é correta, completa, consistente, unívoca e realista.

Correta	o modelo descreve a realidade do cliente e não outra
Completa	todo aspecto de interesse é descrito através de um conceito
Consistente	todos conceitos correspondem ao um aspecto concreto (real)
Unívoca	cada conceito relaciona-se a um único aspecto e vice-versa
Realista	o modelo descreve uma realidade que pode existir

Documento de Análise de Requisitos (DAR)

1. Introdução
 1. Propósito do sistema
 2. Escopo do sistema
 3. Objetivos e critérios de sucesso do projeto
 4. Definições, acrônimos, e abreviaturas
 5. Referências
 6. Visão geral
2. Sistema Atual
3. Sistema Proposto
 1. Visão geral
 2. Requisitos Funcionais
 3. Requisitos não Funcionais
 1. Interface do usuário e fatores humanos *Qual o nível de especialização dos usuários? Qual o tipo de interface requerida?*
 2. Documentação *Tipos de documentação requeridos: do usuário, técnica, nível de detalhe*
 3. Considerações de hardware *Size, tipo dos equipamentos*
 4. Considerações de software *Tipo de S.O., versões, tipo de banco de dados, arquitetura*
 5. Características de desempenho *Tempo de resposta do sistema, número de usuários, acessos concorrentes, condições extremas de carga*
 6. Tratamento de exceções *Exceções de processamento, erros do sistema*
 7. Considerações de qualidade *Robustez, confiabilidade, disponibilidade*
 8. Modificações no sistema *Escopo de modificações previstas*
 9. Ambiente físico *Locais onde o aplicativo deve ser distribuído*
 10. Considerações de segurança *Risco de violação, erros e ataques*
 11. Questões de recursos *Recursos a serem consumidos pelo sistema como espaço de armazenamento*
4. Modelos do sistema
 1. Cenários
 2. Casos de Uso
 3. Modelo de Objetos
 - i. Dicionário de dados
 - ii. Diagrama de classes
 4. Modelos dinâmicos *Diagramas de seqüência, colaboração, estados, atividades etc.*
 5. Interface do usuário *Navegabilidade, screen-shots etc.*
5. Glossário

Técnicas de extração de requisitos

As seguintes técnicas são aplicadas na extração de requisitos adaptando-se a cada caso e, na maior parte das vezes combinadas umas às outras:

Entrevistas	<ul style="list-style-type: none">- Requerem planejamento. As entrevistas devem ser planejadas: objetivo, organização da entrevista, seleção dos participantes, questionário prévio, <i>feedback</i> etc.- Gerar documento de compromisso com os participantes.- Permitem o contato direto com o cliente.
Questionários	<ul style="list-style-type: none">- Requerem conhecimento prévio do assunto e dos clientes (perfil, vocabulário etc.) para elaboração de questões claras e objetivas. Isso torna o questionário impraticável em muitos casos.- Permite padronização das perguntas/respostas e tratamento estatístico dos resultados.- Aplicável quando é necessário obter informações de um grande número de usuários.
Cenários	<ul style="list-style-type: none">- Construídos como instâncias de casos de uso que serão construídos mais adiante no sistema. Apresentam:<ul style="list-style-type: none">• Atores participantes• Condições de entrada• Fluxo de eventos principais<ul style="list-style-type: none">○ Fluxo de eventos secundários• Condições de saída• Requerimentos especiais- Cenários e casos de uso fazem parte dos recursos da linguagem UML e vem sendo amplamente empregados.
Prototipação	<ul style="list-style-type: none">- Lida basicamente em oferecer ao usuário as interfaces do sistema para que ele “experimente” o aplicativo verificando os seus requisitos e encontrando novos.- Apresenta um custo alto de desenvolvimento (protótipos são desenvolvimentos parciais que são descartados depois).- Técnicas de desenvolvimento ágil vêm sistematicamente substituindo protótipos por “entregas parciais” completas.

Projetistas e empresas tendem a misturar vários desses processos criando técnicas de extração de requisitos específicas. As técnicas de **JAD (*Joint Application Design*)** são bastante populares entre os desenvolvedores de software. Basicamente ela envolve a colaboração de clientes, usuários e desenvolvedores, incluindo-se os executivos patrocinadores do projeto (stakeholders) e gerentes do projeto, através de reunião de trabalho ao longo de toda uma

semana para a extração dos requisitos de um sistema. Nesse trabalho são efetuadas discussões, entrevistas, atividades de planejamento, verificações de requisitos e elaborados documentos do sistema.

10 Small Steps, to Better Requirements

Não obstante o esforço despendido no adequado levantamento de requisitos a falta de especificações corretas de requisitos ainda aparece como um dos principais problemas no desenvolvimento de software. Ian Alexander aponta no texto abaixo algumas boas práticas que buscam minimizar esse problema.

Ian Alexander, Requirements Column, IEEE Software, 23, 2, pages 19-21, Mar/Apr, 2006.
<http://easyweb.easynet.co.uk/~iany/index.htm>

Step 1.	Mission & Scope
Step 2.	Stakeholders
Step 3.	Goals
Step 4.	Goal Conflicts
Step 5.	Scenarios
Step 6.	Shall-Statements
Step 7.	Justifications
Step 8.	Assumptions
Step 9.	Agreed Priorities
Step 10.	Acceptance Criteria

Exercícios

1. Cite 3 dificuldades encontradas no levantamento de requisitos.
2. Defina de forma geral do que é a técnica de JAD?
3. **Estudo de Viabilidade.** Antes do levantamento de requisitos devemos nos questionar sobre a Viabilidade do projeto. Que questões principais devem ser verificadas antes de prosseguirmos com o levantamento de requisitos?

(Resposta) Basicamente o estudo de Viabilidade busca responder às perguntas:

- O sistema contribui para os objetivos gerais da organização e encontra-se alinhado com eles?
- O sistema é viável na tecnologia, prazos e recursos (financeiros, humanos etc.) disponíveis para sua realização?
- O sistema se ajusta aos aplicativos e processos atuais (análise de impacto).

4. Cite 6 requisitos não funcionais.
5. Requisitos não funcionais requerem métricas que permitam que sejam verificados. a) Justifique essa afirmativa. b) Exemplifique.

(Resposta item b somente). Seguem-se exemplos de requisitos não funcionais e formas de medida.

Requisito não funcional	Métricas potenciais
Desempenho	Tempo de resposta ao usuário Número de operações/transações por segundo Tempo de carga de uma tela Taxa de transferência de arquivos
Capacidade	Gigabytes de armazenamento requeridos Número de usuários concorrentes
Facilidade de uso	Quantidade de horas de treinamento requeridas Número de ações requeridas para executar uma dada operação (clicks, telas etc.)
Confiabilidade	Percentual de indisponibilidade do sistema Número de falhas/incidentes por um período
Portabilidade	Relação de sistemas e versões para as quais se espera portabilidade
Robustez	Tempo de recuperação do sistema para diferentes cenários desastre (perda do banco de dados, perda da máquina etc.)

6. Cite ao menos 4 boas práticas a serem observadas para o sucesso do levantamento de requisitos.

7. Um dos maiores problemas na engenharia de requisitos é que os requisitos mudam. a) Justifique essa afirmativa. b) Não sendo possível garantir que os requisitos não mudam como devemos proceder para garantir o sucesso de um projeto.