

Módulo II

Requisitos Arquiteturais e
Modelagem Arquitetural

Prof. Dr. João Paulo Aramuni

Tipos de requisitos e suas classificações

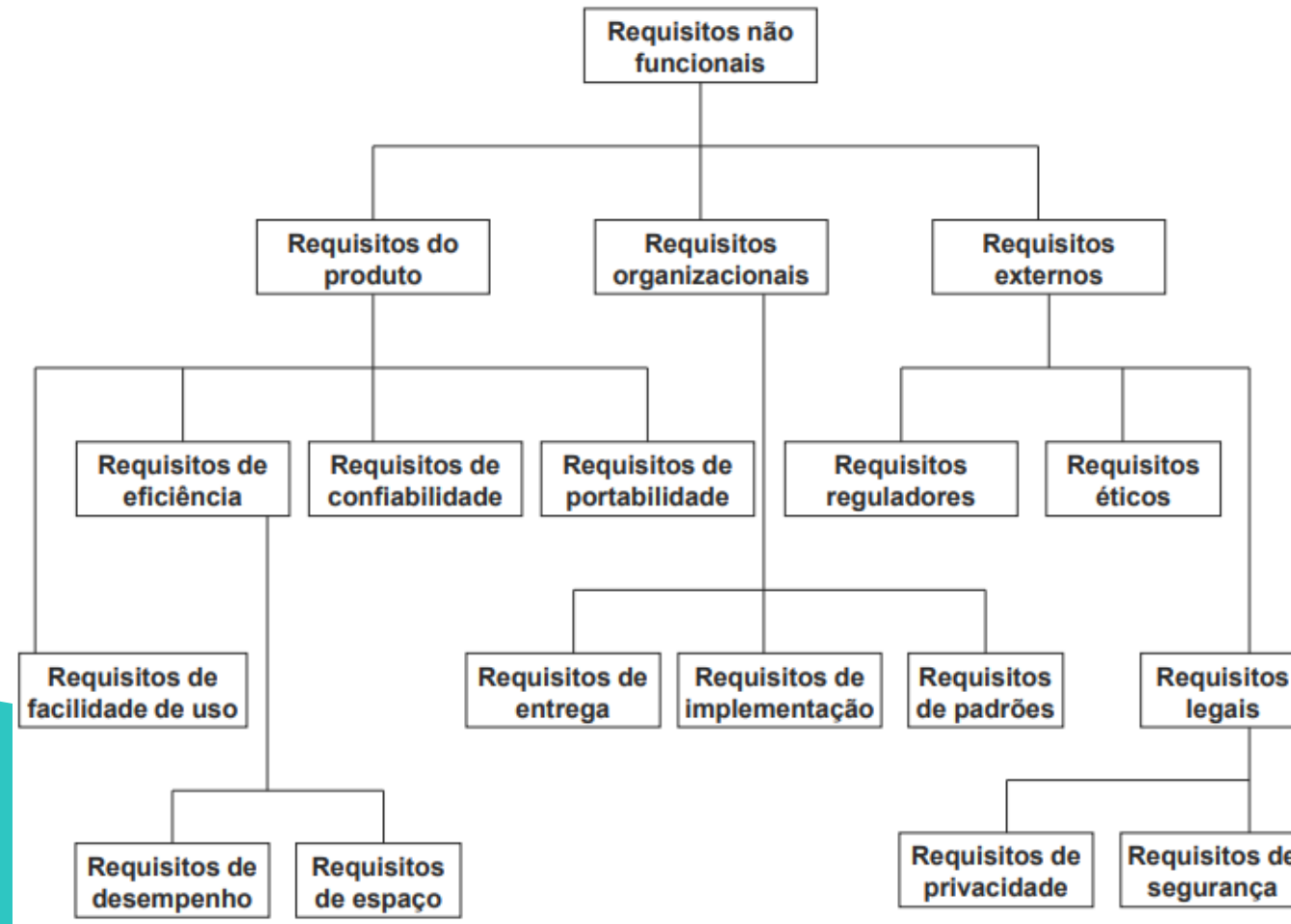


- ❑ 3.4 – Classificação de Requisitos Não Funcionais.

Classificação de Requisitos Não Funcionais

Os requisitos não funcionais ainda são classificados em três tipos, são eles: Requisitos do Produto Final, Requisitos Organizacionais e Requisitos Externos.

Classificação de Requisitos Não Funcionais



Classificação de Requisitos Não Funcionais



Requisitos do Produto Final referem-se a como o produto deve comportar-se, ou seja, a sua velocidade de execução, confiabilidade, etc.

Requisitos Organizacionais referem-se à consequência de políticas e procedimentos organizacionais que devem ser seguidos.

Requisitos Externos referem-se a fatores externos ao sistema e ao processo de desenvolvimento como a legislação.

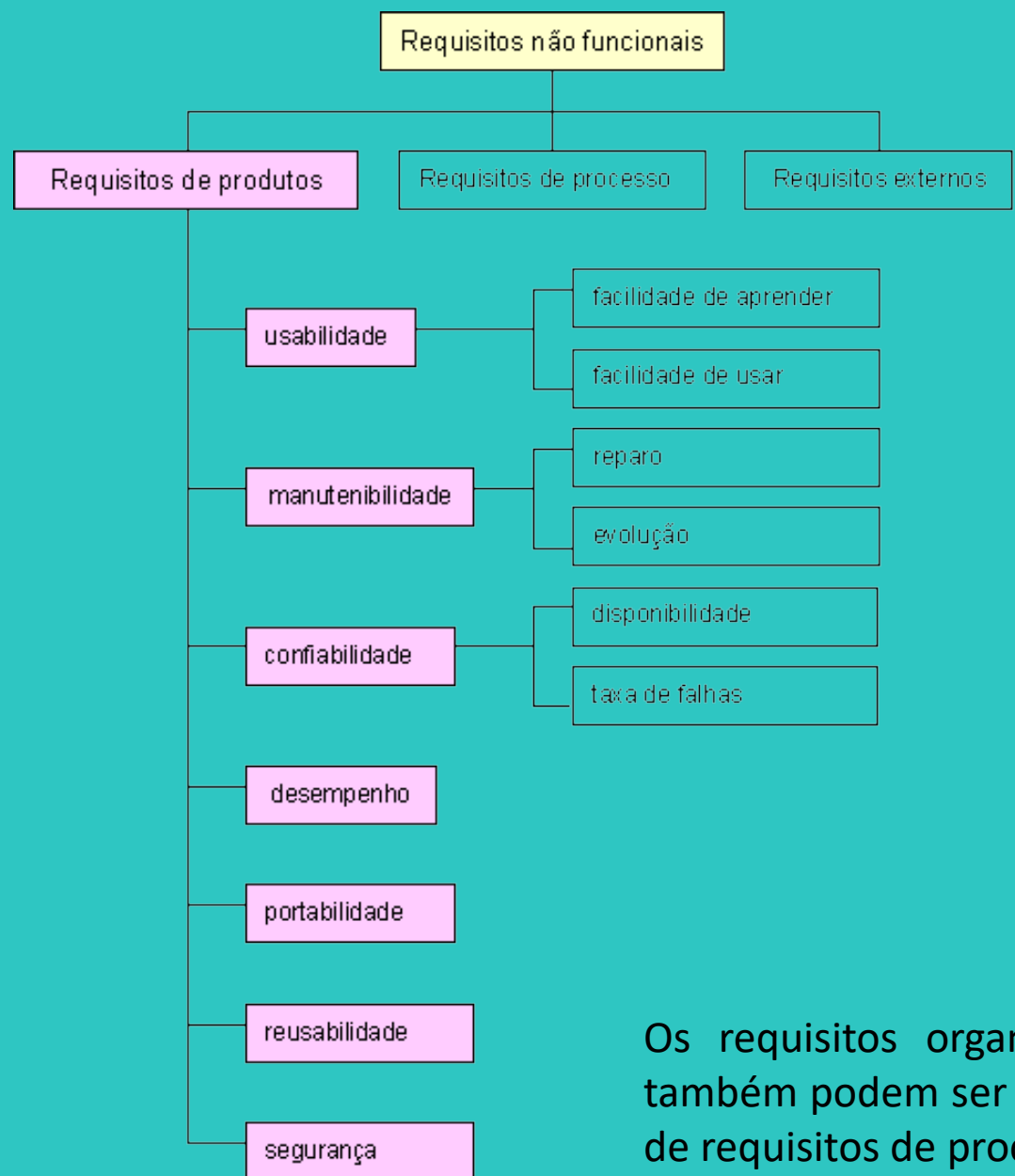
Classificação de Requisitos Não Funcionais



Exemplo de Requisito do Produto Final: A interface do usuário deve ser implementada como simples HTML.

Exemplo de Requisito Organizacional: Todos os documentos entregues devem seguir o padrão de relatórios XYZ-00.

Exemplo de Requisito Externo: Informações pessoais dos usuários não podem ser vistas pelos operadores do sistema.



Os requisitos organizacionais também podem ser chamados de requisitos de processo.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Usabilidade:**

Usabilidade é um dos atributos de qualidade ou requisitos não funcionais de qualquer sistema interativo, ou seja, no qual ocorre interação entre o sistema e seres humanos.

A noção de usabilidade vem do fato que qualquer sistema projetado para ser utilizado pelas pessoas deveria ser fácil de aprender e fácil de usar, tornando assim fácil e agradável a realização de qualquer tarefa.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Usabilidade:**

Requisitos de usabilidade especificam tanto o nível de desempenho quanto a satisfação do usuário no uso do sistema.

Dessa forma, a usabilidade pode ser expressa em termos de:

- Facilidade de aprender: Associado ao tempo e esforço mínimo exigido para alcançar um determinado nível de desempenho no uso do sistema.
- Facilidade de uso: Relacionado à velocidade de execução de tarefas e à redução de erros no uso do sistema.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Usabilidade:**

Critérios de medição de usabilidade:

1. Tempo para realizar um tarefa.
2. Percentual de tarefa concluído.
3. Percentual de tarefa concluído por unidade de tempo.
4. Taxa de sucessos/falhas.
5. Tempo consumido com erros.
6. Percentual de erros.
7. Número de comandos utilizados.
8. Número de comandos disponíveis não utilizados.
9. Frequência de uso de *ajuda* (help) ou documentação.
10. Número de vezes que o usuário expressa satisfação ou frustração.

Fonte: Tyldesley (1988)

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Manutenibilidade:**

De um modo geral, a manutenibilidade é um dos requisitos mais relacionados com a arquitetura de um sistema de software.

A facilidade de fazer alteração no sistema existente, seja adicionando ou modificando alguma funcionalidade, depende muito da arquitetura do mesmo.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Manutenibilidade:**

Adição de novos componentes deve ocorrer sem a necessidade de modificar a arquitetura existente e ainda comprometer pouco (ou nada) o desempenho atual do sistema.

Tal suporte à manutenibilidade (seja para correção ou evolução do sistema) deve ser facilmente acomodada pela arquitetura de software.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Confiabilidade:**

Confiabilidade de software é a probabilidade de o software não causar uma falha num sistema durante um determinado período de tempo sob condições especificadas.

A probabilidade é uma função da existência de defeitos no software. Assim, os estímulos recebidos por um sistema determinam a existência ou não de algum defeito.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Confiabilidade:**

Em outras palavras, a confiabilidade de software, geralmente definida em termos de comportamento estatístico, é a probabilidade de que o software irá operar como desejado num intervalo de tempo conhecido.

Também, a confiabilidade caracteriza-se um atributo de qualidade de software o qual implica que um sistema executará suas funções como esperado.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Confiabilidade:**

Exemplos de métricas utilizadas para avaliar a confiabilidade de software compreendem:

- ☐ Disponibilidade
- ☐ Taxa de ocorrência de falha
- ☐ Probabilidade de falha durante fase operacional
- ☐ Tempo médio até a ocorrência de falha

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Desempenho:**

Desempenho é um atributo de qualidade importante para sistemas de software.

Considere, por exemplo, um sistema de uma administradora de cartões de crédito. Em tal sistema, um projetista ou engenheiro de software poderia considerar os requisitos de desempenho para obter uma resposta de tempo para autorização de compras por cartão.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Desempenho:**

Adicionalmente, desempenho é importante porque afeta a usabilidade de um sistema.

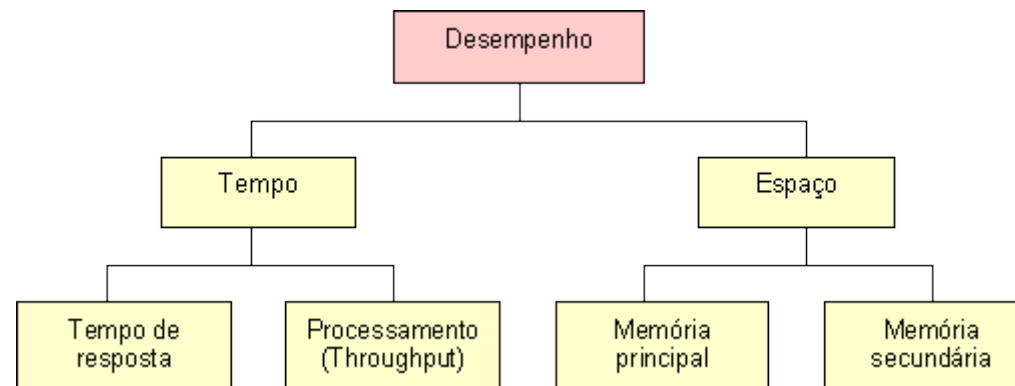
Se um sistema de software é lento, ele certamente reduz a produtividade de seus usuários ao ponto de não atender às suas necessidades.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Desempenho:**

Fatores de desempenho:



Classificação de Requisitos Não Funcionais



Requisitos do produto: **Desempenho:**

O requisito de desempenho restringe a velocidade de operação de um sistema de software.

Isto pode ser visto em termos de:

- ☐ Requisitos de resposta
- ☐ Requisitos de processamento
- ☐ Requisitos de temporização
- ☐ Requisitos de espaço

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Portabilidade:**

Portabilidade pode ser definida como a facilidade na qual o software pode ser transferido de um sistema computacional ou ambiente para outro.

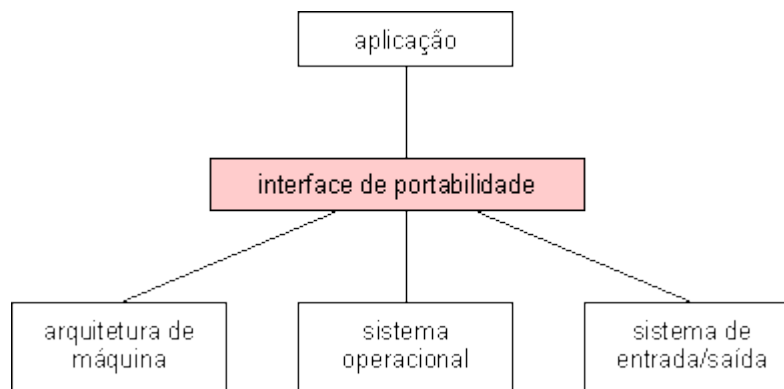
De um modo geral, a portabilidade refere-se à habilidade de executar um sistema em diferentes plataformas. É importante observar que à medida que aumenta a razão de custos entre software e hardware, a portabilidade torna-se cada vez mais importante.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Portabilidade:**

Uma abordagem geral que poderia ser adotada para obter um sistema portátil tentaria separar as partes do sistema que dependem do ambiente externo numa camada ou interface de portabilidade:



Classificação de Requisitos Não Funcionais



Requisitos do produto: **Portabilidade:**

Dessa forma, quando o sistema de software muda de hardware ou sistema operacional, apenas a interface de portabilidade precisaria ser alterada.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Reusabilidade:**

Uma característica das engenharias é fazer uso de projetos existentes a fim de reutilizar componentes já desenvolvidos, objetivando minimizar o esforço em novos projetos.

O reuso pode ser visto sob diferentes perspectivas. Ele pode ser orientado a componentes, orientado a processos ou orientado ao conhecimento específico de um domínio.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Reusabilidade:**

Exemplos de reuso de componentes:

- ❑ **Aplicação:** Toda a aplicação poderia ser reutilizada.
- ❑ **Subsistemas:** Os principais subsistemas de uma aplicação poderiam ser reutilizados.
- ❑ **Objetos ou módulos:** Componentes de um sistema, englobando um conjunto de funções, podem ser reutilizados.
- ❑ **Funções:** Componentes de software que implementam uma única função (como uma função matemática) podem ser reutilizados.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Segurança:**

Em um sistema de software, este requisito não funcional caracteriza a segurança de que acessos não autorizados ao sistema e dados associados não serão permitidos.

Portanto, é assegurada a integridade do sistema quanto a ataques intencionais ou acidentes. Dessa forma, a segurança é vista como a probabilidade de que a ameaça de algum tipo será repelida.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Segurança:**

Exemplos de requisitos de segurança são:

- ☐ Apenas pessoas que tenham sido autenticadas por um componente de controle acesso e autenticação poderão visualizar informações .
- ☐ As permissões de acesso ao sistema podem ser alteradas apenas pelo administrador de sistemas.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Segurança:**

Exemplos de requisitos de segurança são:

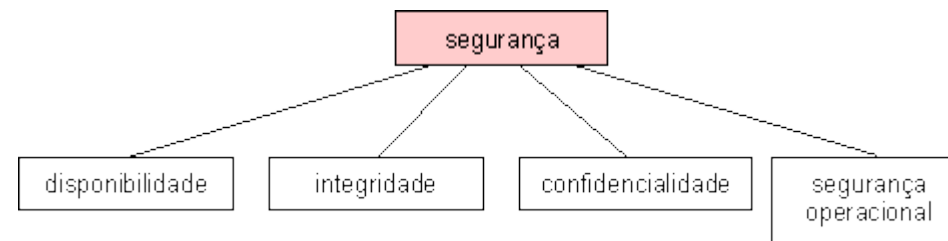
- ☐ Deve ser feito cópias (backup) de todos os dados do sistema a cada 24 horas e estas cópias devem ser guardadas em um local seguro, sendo preferencialmente num local diferente de onde se encontra o sistema.
- ☐ Todas as comunicações externas entre o servidor de dados do sistema e clientes devem ser criptografadas.

Classificação de Requisitos Não Funcionais



Requisitos do produto: **Segurança:**

Tipos de segurança:

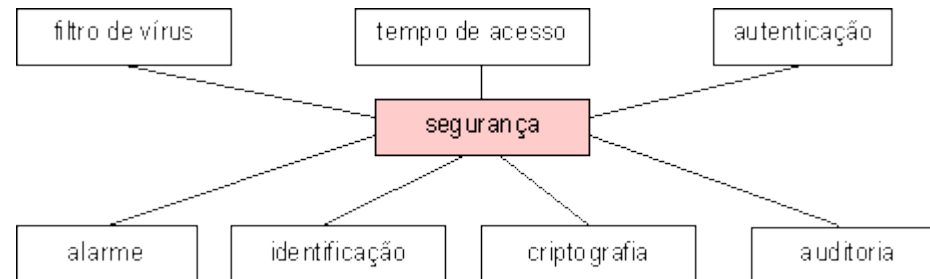



Classificação de Requisitos Não Funcionais



Requisitos do produto: **Segurança:**

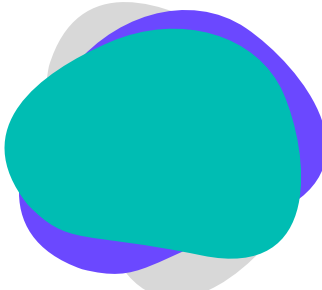

Métodos usados para prover segurança:



A teal and purple abstract shape, resembling a brain or a cloud, is positioned to the left of the text.

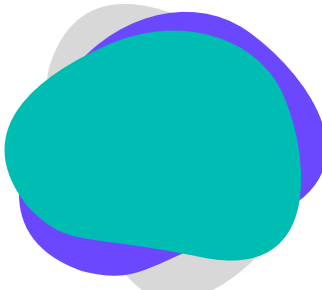
Problema: A especificação quantitativa de requisitos não funcionais é difícil.

Problema: Ocorre mistura de requisitos funcionais e não funcionais.

A teal and purple abstract shape, resembling a brain or a cloud, is positioned to the right of the text.A teal and purple abstract shape, resembling a brain or a cloud, is positioned to the left of the text.

Problema: Requisitos não funcionais podem conflitar com outros requisitos (funcionais ou não)

Problema: Nem sempre o arquiteto de software está disponível para auxiliar nesta tarefa.

A teal and purple abstract shape, resembling a brain or a cloud, is positioned to the right of the text.

Referências



- ❑ Acquisition Practices: Good and Bad
www.sei.cmu.edu/programs/acquisition-support/conf/2003-presentations/oberndorf.pdf
- ❑ Are All Quality Goals Created Equal? Functional vs. Non-Functional
www.sei.cmu.edu/architecture/saturn/2005/quality_steven.pdf
- ❑ Non-Functional Requirements in Software Engineering
<http://www.utdallas.edu/~chung/BOOK/book.html>
- ❑ PRESSMAN, R. S.; Software Engineering: A Practitioner's Approach, 7 ed., McGraw Hill, 2010.
- ❑ SEI's Software Architecture Technology Initiative
www.sei.cmu.edu/architecture/sat_init.html
- ❑ SOMMERVILLE, I.; Software Engineering, 8. ed., Addison-Wesley, 2007.
- ❑ The Software Architecture Portal
<http://www.softwarearchitectureportal.org/>
- ❑ Tyldesley, D. A. (1988): Employing usability engineering in development of office products. Computer Journal, 31(5), pp. 431-436.

Obrigado!

IGTi