



# Requisitos

© Prof. Raul Sidnei Wazlawick  
UFSC-CTC-INE

2010

Fonte: Análise e Projeto de Sistemas de Informação Orientados a  
Objetos, 2ª Edição, Elsevier, 2010.

# Requisitos

- O levantamento e a análise de requisitos compõem uma parte significativa da fase de concepção dentro do UP.
- O analista pode e deve utilizar todas as informações disponíveis para identificar as fontes de requisitos (departamentos, pessoas, clientes, interfaces, sistemas etc.) e, para cada fonte, identificar as funções que o sistema deverá disponibilizar.

# Levantamento de Requisitos

- O levantamento de requisitos é o processo de descobrir quais são as *funções* que o sistema deve realizar e quais são as *restrições* que existem sobre essas funções.
- No caso do sistema Livir, por exemplo, o levantamento de requisitos vai permitir descobrir que o sistema deve controlar a compra e venda de livros, calcular automaticamente os pagamentos, permitir o registro de danos aos livros, gerar relatórios de vendas, verificar a disponibilidade de livros em estoque etc.
- Essas operações e muitas outras virão a constituir a *funcionalidade* do sistema, e por isso são chamadas também de *requisitos funcionais*.

- 
- Na fase de concepção, o levantamento de requisitos é rápido e genérico.
  - Ele é feito em extensão e não em profundidade.
  - O analista deve entender a extensão do que o sistema deve fazer, mas sem detalhar *como* ele vai fazer.
  - Somente na fase de elaboração a análise dos requisitos será aprofundada.

# Desafios

- como *descobrir* os requisitos;
- como *comunicar* os requisitos para as outras fases ou equipes do projeto;
- como *lembrar* dos requisitos durante o desenvolvimento e verificar se foram todos atendidos;
- como *gerenciar* a mudança dos requisitos.

## Requisitos Funcionais – Elementos:

- a *descrição* de uma função a ser executada pelo sistema (usualmente entrada, saída ou transformação da informação);
- a *origem* do requisito (quem solicitou) e/ou quem vai executar a função em questão (usuário);
- quais as *informações* que são passadas do sistema para o usuário e vice-versa quando a função for executada;
- que *restrições lógicas* (regras de negócio) ou *tecnológicas* se aplicam à função.

# Requisitos Não-Funcionais

- Os requisitos não funcionais aparecem sempre ligados a requisitos funcionais e podem ser basicamente de dois tipos: lógicos ou tecnológicos.

# Restrições Lógicas

- São as regras de negócio relacionadas à função em questão.
- Por exemplo, no registro de uma venda, uma série de restrições lógicas poderia ser considerada, como por exemplo: não efetuar a venda caso a operadora de cartão não autorize o pagamento ou não efetuar a venda caso a venda anterior tenha sido cancelada devido a um endereço inválido que ainda não foi corrigido.

# Restrições Tecnológicas

- Dizem respeito à tecnologia para a realização da função, como por exemplo, a interface (Web, por exemplo), o tipo de protocolo de comunicação, restrições de segurança ou tolerância a falhas etc.

# Requisitos Suplementares

- Os *requisitos suplementares* são todo tipo de restrição tecnológica ou lógica que se aplica ao sistema como um todo e não apenas a funções individuais.
- Por exemplo, um requisito suplementar pode estabelecer que o sistema deva ser compatível com um banco de dados legado ou ser implementado em uma determinada linguagem de programação, ou ainda seguir um determinado padrão de interface, ou *look and feel*.

# Documento de Requisitos

- O *documento de requisitos* registra todos os tópicos relativos ao que o sistema deve fazer e sob que condições o sistema deve fazer as coisas.
- Esse documento ainda não precisa ser totalmente estruturado, e assume-se que não será completo em profundidade, embora se possa esperar que esteja razoavelmente completo em extensão.
- Eventuais lacunas desse documento serão preenchidas durante a fase de elaboração.

# Exemplo

## **Sistema Livir — Documento de Requisitos**

### **Requisitos funcionais**

1. Registrar novos títulos a partir do catálogo das editoras.
2. Registrar vendas de livros.
3. Realizar encomendas de livros.
4. Registrar e autorizar pagamentos com cartão de crédito.
5. Registrar e aplicar promoções.
6. Calcular custos de entrega.
7. Emitir relatório de livros mais vendidos.
8. Emitir relatório de compradores mais assíduos.
9. Emitir sugestões de compra para compradores baseadas em compras anteriores.
10. ...

### **Requisitos suplementares**

1. O sistema deve operar via interface Web
2. Todos os controles de interface devem ter um campo de ajuda associado.
3. ...

# Detalhamento

## 1. Registrar novos títulos a partir do catálogo das editoras

**Descrição:** O gerente seleciona as editoras para as quais pretende fazer a atualização. O processo é automático. O sistema consulta os ISBN disponibilizados e os comprara com os existentes na base. Havendo novos ISBN, o sistema atualiza a base com as novas informações.

**Fontes:** Sr. Fulano de Tal (gerente) e manual técnico da interface de catálogo das editoras.

**Usuário:** O próprio gerente.

### **Informações de entrada:**

O gerente informa quais são as editoras para as quais pretende fazer a atualização a partir de uma lista fornecida pelo sistema.

### **Informações de saída:**

— A lista de editoras (nome).

— O relatório de atualizações efetuadas (uma lista contendo: nome da editora, ISBN, título e preço de compra).

### **Restrições lógicas:**

Não há.

### **Restrições tecnológicas:**

1. Deve ser usado o sistema de interface com as editoras, de acordo com o manual XYZ.1234.
2. A seleção feita pelo gerente se dá através de uma lista de seleção múltipla, a qual deve ser ordenada de forma alfabética.
3. ...

# Análise de Requisitos

- Na análise de requisitos, o analista vai procurar caracterizar certas propriedades dos requisitos já levantados:
  - Permanentes e transitórios
  - Evidentes e ocultos
  - Obrigatórios e desejados

# Permanência e Transitoriedade

- Requisitos não funcionais podem ser considerados permanentes (não vão mudar) ou transitórios (podem mudar) de acordo com uma decisão tomada pelo analista em conjunto com o cliente.
- O requisito não tem a propriedade de permanência ou transitoriedade por si: ela é decidida de acordo com a conveniência.
- Um mesmo requisito pode ser considerado permanente ou transitório dependendo do que se queira em relação ao tempo de desenvolvimento ou custo da manutenção do software.

# Se o requisito for permanente

- fica mais barato e rápido desenvolver o sistema em si;
- fica mais caro e demorado mudar o sistema caso o requisito, por algum motivo, mude no futuro.

# Se o requisito for transitório

- fica mais caro e complexo desenvolver o sistema (ele deverá acomodar funcionalidades para a mudança);
- fica mais barato e rápido fazer a manutenção no sistema (o sistema já está preparado para acomodar a mudança com uma simples reconfiguração).

# Requisitos Evidentes

- São funções efetuadas com conhecimento do usuário.
- Esses requisitos usualmente correspondem a trocas de informação, como consultas e entrada de dados, que ocorrem com o meio exterior através da interface do sistema.

# Requisitos Ocultos

- São funções efetuadas pelo sistema sem o conhecimento explícito do usuário.
- Usualmente são cálculos ou atualizações feitas pelo sistema sem a solicitação explícita do usuário, mas como consequência de outras funções solicitadas por ele.

# Requisitos Obrigatórios e Desejados

- Os requisitos ainda podem ser classificados em *obrigatórios* e *desejados*, ou seja, aqueles que devem ser obtidos de qualquer maneira e aqueles que podem ser obtidos caso isso não cause maiores transtornos no processo de desenvolvimento.

- 
- *Os requisitos não funcionais e suplementares são bem mais imprevisíveis do que os funcionais para efeito de estimativa de esforço.*
  - *Assim, em alguns casos, pode ser necessário efetivamente classificar esses requisitos por graus de prioridade.*

# Classificação de requisitos não funcionais e suplementares

- Usabilidade
- Confiabilidade
- Desempenho
- Configurabilidade
- Segurança
- Implementação
- Interface
- Empacotamento
- Legais

# Usabilidade

- Quais fatores humanos estão envolvidos no sistema?
- Que tipo de ajuda o sistema vai prover?
- Quais as formas de documentação ou manuais estarão disponíveis?
- Como esses manuais vão ser produzidos?
- Que tipo de informação eles vão conter?
- Seria interessante definir esses tópicos na fase de concepção, visto que o contrato com o cliente deve especificar muitas dessas questões.

# Confiabilidade

- Que tipo de tratamento de falhas o sistema vai ter?
- O analista não é obrigado a produzir um sistema totalmente tolerante a falhas, mas deve estabelecer que tipo de falhas o sistema será capaz de gerenciar:
  - falta de energia,
  - falha de comunicação,
  - falha na mídia de gravação etc.
- Não se deve confundir *falha* com erro de programação, visto que erros de programação são elementos que nenhum software deveria conter.
- As falhas são situações anormais que podem ocorrer mesmo para um software implementado sem nenhum erro de programação;

# Desempenho

- Que tipo de eficiência e precisão o sistema será capaz de apresentar?
- Pode-se estabelecer, por exemplo, como requisito de eficiência, que nenhuma consulta à base de dados de compradores vai demorar mais de cinco segundos.
- Na fase de concepção não se define *como* o sistema fará para cumprir o requisito, apenas se diz que de alguma forma ele terá de ser cumprido no projeto.
- Cabe ao projetista e programador garantir que o requisito seja satisfeito.
- Se o analista por algum motivo conclui que o requisito dificilmente poderá ser implementado, o requisito passa a ser um *risco* do sistema e eventualmente necessitará de um estudo mais aprofundado ainda na fase de concepção, para verificar a possibilidade de sua realização.

# Configurabilidade

- O que pode ser configurado no sistema?
- Deve-se definir os elementos que poderão ser configurados pelo usuário sem que seja necessário recompilar o sistema.
- Exemplos de itens configuráveis são:
  - o tipo de impressoras,
  - a moeda do país,
  - políticas da empresa,
  - fontes e cores da interface,
  - idioma etc.

# Segurança

- Quais são os tipos de usuários e que funções cada um pode executar?
- Sugere-se a implantação de um sistema de permissões, de forma que possam ser criados dinamicamente perfis de usuários com diferentes conjuntos de permissões.

# Implementação

- Qual linguagem deve ser usada?
- Por que motivo?
- Que bibliotecas estarão disponíveis?  
Quais bancos de dados serão acessíveis?
- Há necessidade de comunicação com sistemas legados?

# Interface

- Como deve ser a interface?
- Vai ser seguida alguma norma ergonômica?

# Empacotamento

- De que forma o software deve ser entregue ao usuário final?

# Legais

- Muitas vezes, uma equipe de desenvolvimento deve contar com uma assessoria jurídica para saber se está infringindo direitos autorais ou normas específicas da área para a qual o software está sendo desenvolvido.

- 
- Embora essa lista seja extensa, o analista deve ter em mente que se trata apenas de uma forma de classificação para que ele possa mais facilmente avaliar quais requisitos são relevantes para cada um dos tipos listados.
  - Não há necessidade de procurar requisitos que não existem, por exemplo, estabelecendo complicados requisitos de empacotamento para um cliente para o qual não faz a menor diferença a forma como o software será entregue.