



Conteúdo

1. Introdução
2. Levantamento de Requisitos
3. Análise Orientada a Objetos
4. Projeto Orientado a Objetos
5. UML
6. Métodos Ágeis

Conteúdo

1. Introdução

2. Levantamento de Requisitos

- Requisitos Funcionais e Casos de Uso
- Relacionamentos entre Casos de Uso
- Diagramas de Sequência do Sistema
- Requisitos Não-Funcionais

3. Análise Orientada a Objetos

4. Projeto Orientado a Objetos

5. UML

6. Métodos Ágeis



Levantamento de Requisitos

Requisitos

Requisitos de Sistema definem o que o sistema deve fazer e as circunstâncias sob as quais ele deve operar.

Exemplos de requisitos para um sistema de biblioteca:

- O sistema irá manter registros de todo o material da biblioteca, incluindo livros, revistas, jornais e DVDs.
- O sistema irá permitir que os usuários pesquisem por um item pelo título, autor ou ISBN.
- O sistema será implementado como uma aplicação web.
- O sistema irá dar suporte a, no mínimo, 20 transações por segundo.
- As facilidades do sistema que estarão disponíveis ao público serão demonstráveis em 10 minutos ou menos.

Requisitos

- O sistema irá manter registros de todo o material da biblioteca, incluindo livros, revistas, jornais e DVDs.

→ requisitos bem gerais

- O sistema irá permitir que os usuários pesquisem por um item pelo título, autor ou ISBN.

→ requisitos funcionais

- O sistema será implementado como uma aplicação web.

→ requisitos de implementação

- O sistema irá dar suporte a, no mínimo, 20 transações por segundo.

→ requisitos de performance

- As facilidades do sistema que estarão disponíveis ao público serão demonstráveis em 10 minutos ou menos.

→ requisitos de usabilidade

Requisitos

O que são os requisitos?

Os requisitos são, no geral, descrições de como o sistema deverá se comportar, informação sobre o domínio, restrições da operação do sistema ou especificações de uma propriedade do sistema.

Quando os requisitos são definidos?

Os requisitos são definidos durante os estágios iniciais do desenvolvimento de um sistema, especificando o que deverá ser implementado.

↖ **ultrapassado**

Modelo em Cascata: define todos os requisitos no início do desenvolvimento, considerando que eles não serão modificados.

Métodos Evolucionários: somente uma parte dos requisitos (em torno de 10 a 20%) são definidos inicialmente. A programação e os testes iniciam antes que todos os requisitos tenham sido definidos.



Requisitos Funcionais e Não Funcionais

Requisitos Funcionais

São sentenças sobre os serviços que o sistema deve fornecer, como o sistema deve reagir a entradas particulares e como o sistema deve se comportar em determinadas situações.

Requisitos Não Funcionais

São restrições sobre os serviços oferecidos pelo sistema. Inclui restrições sobre tempo, restrições sobre o processo de desenvolvimento a serem seguidos, etc.



Problemas com os Requisitos

Consequências decorrentes de problemas ocorridos durante o levantamento de requisitos:

- o sistema pode ser entregue com atraso e custar mais do que originalmente previsto;
- o cliente e os usuários não estão satisfeitos com o sistema: eles podem não usar suas facilidades ou decidir abandoná-lo;
- o sistema pode ser pouco confiável, conter erros e abortar abruptamente;
- o custo da manutenção e/ou evolução é muito alto.



Atividades do Levantamento de Requisitos

Levantamento de Requisitos

- Levantamento de Requisitos inclui as atividades que envolvem a identificação, documentação e manutenção dos requisitos de um sistema.

↙ → serve para comunicar e relembrar os requisitos

- ➡ Não existe um processo de levantamento de requisitos que seja o melhor para todas as organizações



Atividades do Levantamento de Requisitos

1. Identificação e Descrição dos Requisitos

- Visão
- Requisitos Funcionais
- Requisitos Não Funcionais
- Glossário

2. Análise dos Requisitos

3. Negociação dos Requisitos

4. Documentação Final dos Requisitos





Identificação dos Requisitos

Identificação dos Requisitos

Os analistas trabalham com os clientes e usuários para descobrir o problema a ser resolvido, os serviços do sistema, a performance requisitada do sistema, restrições de software, etc.



Os requisitos são descobertos através de:

- consultas com as pessoas envolvidas com o sistema (clientes, usuários, gerentes, etc);
- conhecimento do domínio e do mercado;
- análise dos sistemas existentes.



Técnicas de Identificação de Requisitos

1. Entrevistas
2. Cenários
3. Observação

1. Entrevista

O analista discute o sistema com diferentes clientes e usuários e chega a um entendimento dos seus requisitos.

➔ Técnica de levantamento de requisitos mais usada.


Existem dois tipos de entrevistas:

1. Entrevistas fechadas

O analista procura por respostas a um conjunto pré-definido de questões.

2. Entrevistas abertas

Não existe um conjunto pré-definido de questões e o analista discute, de maneira aberta, o que os usuários e clientes querem do sistema.




2. Cenário

Cenário: exemplo de sessão de interação entre um usuário final e o sistema.

Como funciona a identificação através de cenários?

O usuário simula o uso do sistema para realizar uma tarefa seguindo o cenário, enquanto o analista vai tomando notas dos comentários, problemas e sugestões do usuário.



3. Observação

Os usuários são observados executando o seus trabalhos normalmente.

Durante a observação, o analista deverá descrever detalhadamente as práticas de trabalho.

Também podem ser usados vídeos e áudios, mas o processamento posterior destes é muito demorado.



Descrição dos Requisitos



Técnicas de Descrição dos Requisitos

➔ Não existe maneira única de especificar os requisitos

Algumas técnicas de descrição dos requisitos:

1. Linguagem Natural

- Casos de Uso
- Lista de Características (sentenças)

2. Protótipos

3. Protótipos de Baixa Fidelidade



1. Linguagem Natural

Na maioria das organizações, os requisitos são escritos como parágrafos de linguagem natural, adicionados por diagramas e equações.

Vantagens:

Linguagem natural é a única notação que é geralmente entendível por todos os leitores potenciais dos requisitos.

Desvantagens:

Os requisitos em linguagem natural podem ser ambíguos, pouco claros e causar mal entendidos.

2. Protótipos

Um protótipo é uma versão do sistema que está disponível no início do processo de desenvolvimento.

Tipos de protótipos:

- Protótipos “throw-away”: ajudam a levantar e analisar os requisitos. São descartados antes do desenvolvimento do sistema final.
- Protótipos evolucionários: um sistema com funcionalidade limitada é disponível aos usuários no início do processo de desenvolvimento. Este sistema é então modificado e estendido para produzir o sistema final.



o tempo de vida geral é relativamente curto, pois são pobremente estruturados e, assim, os custos de manutenção e evolução são altos

2. Protótipos

Vantagens:

- permitem aos clientes e usuários entender melhor como o sistema pode ser usado para dar suporte aos seus trabalhos;
- mal-entendidos entre projetistas e usuários podem ser identificados quando a funcionalidade do sistema é demonstrada usando o protótipo;
- funções que estão faltando e requisitos incompletos e inconsistentes podem ser detectados usando o protótipo.

Desvantagens:

- tempo e custo requerido para desenvolver um protótipo (que pode aumentar o tempo de entrega ao usuário);
- os usuários podem prestar mais atenção na interface com o usuário do que nos requisitos da aplicação;
- familiarização com uma interface com o usuário antes da versão final do sistema estar desenvolvida.

3. Protótipos de Baixa Fidelidade

São aqueles que podem gerar e mostrar rapidamente um conjunto de janelas estáticas de uma aplicação.

Não apresentam toda a funcionalidade da aplicação, não são interativos e não mostram os detalhes de como o usuário deve utilizar a aplicação.

Alguns tipos de protótipos de baixa fidelidade:

- *storyboard*
- *sketching*

3. Protótipos de Baixa Fidelidade

Storyboard: mostra a interação do usuário com o sistema através de uma sequência com o conteúdo das telas que serão acessadas pelo usuário durante a execução da tarefa.

Sketching: mostra a interação do usuário com o sistema de maneira mais informal, utilizando gráficos e desenhos.

A forma usual do *sketching* é a representação da interação no papel, mas ela também pode ser feita através do computador.

Os *sketches* são usados geralmente para representar diferentes decisões de projeto que devem ser discutidas com os usuários.

Documento de Requisitos


Importante: assegurar que os requisitos foram concordados pelo cliente

➤ Documento de Requisitos

Os requisitos do sistema são, geralmente, documentados em um documento formal que é usado para comunicar os requisitos aos clientes, engenheiros de software e gerentes do processo.

Existem diferentes maneiras de estruturar o documento de requisitos.

Exemplo de padrão para o documento de requisitos: IEEE 830/1998



Documento de Requisitos

O documento de requisitos, geralmente, descreve o seguinte:

- um capítulo introdutório que fornece uma visão geral do sistema;
- informações sobre o domínio do sistema;
- os serviços e funções que o sistema deverá fornecer;
- as restrições sob as quais o sistema deverá operar;
- uma descrição do hardware no qual o sistema deverá rodar;
- definições dos outros sistemas com os quais o sistema deverá estar integrado;
- restrições no processo usado para desenvolver o sistema;
- um glossário que define os termos técnicos usados no documento.



Atividades do Levantamento de Requisitos

1. Identificação e Descrição dos Requisitos

- Visão
- Requisitos Funcionais
- Requisitos Não Funcionais
- Glossário

2. Análise dos Requisitos

3. Negociação dos Requisitos

4. Documentação Final dos Requisitos





Visão



Visão

Apresenta de maneira concisa as idéias a respeito do porquê que o projeto está sendo proposto, o esboço dos requisitos centrais, quais os problemas, quem são os stakeholders (envolvidos no projeto), o que eles precisam e como são as soluções propostas.

A Visão do sistema também inclui brevemente:

- características do sistema (que serão detalhados nos casos de uso)
- outros requisitos (que serão detalhados nos requisitos não funcionais)

➔ Evite a duplicação de informações.

Detalhes sobre as informações podem ser registrados nos casos de uso e nos requisitos não funcionais.

Visão

Exemplo - Ponto de Venda:

Visamos uma nova geração de aplicação de ponto de venda tolerante a falhas, com a flexibilidade de dar suporte a diferentes regras de negócio, múltiplos terminais e a integração com vários sistemas auxiliares de terceiros.

Problema: Sistemas tradicionais são inflexíveis, intolerantes a falhas e difíceis de serem integrados com sistemas de terceiros. Isto leva a problemas no tempo de processamento das compras.

Visão

Exemplo - Ponto de Venda:

→ Resumo das Características do Sistema:

- Captura das compras.
- Autorização de pagamento (cartão de crédito, débito automático, cheque).
- Processamento de vendas offline automático quando componentes externos falham.
- Definição e execução de regras de negócio customizadas em pontos fixos dos casos de uso.
- ...