

O Processo de Desenvolvimento de Software

Desenvolvimento de software é uma atividade complexa. Essa complexidade corresponde à sobreposição das complexidades relativas ao desenvolvimento dos seus diversos componentes: software, hardware, procedimentos etc.

Isso se reflete no **alto número** de projetos de software que **não** chegam ao fim, ou que extrapolam recursos de tempo e dinheiro alocados.

Um processo de desenvolvimento de software compreende todas as atividades necessárias para

definir, desenvolver, testar e manter um produto de **software**. Exemplos de processos de desenvolvimento propostos são o **ICONIX**, o **RUP** (*Rational Unified Process*), o **EUP** (*Enterprise Unified Process*), **XP** (*Extreme Programming*) e o **OPEN** (*Objectoriented Process, Environment and Notation*).

Levantamento de Requisitos

A atividade de **levantamento de requisitos** corresponde à etapa de compreensão do problema aplicada ao desenvolvimento de software.

O **principal objetivo** do levantamento de requisitos é que usuários e desenvolvedores tenham a **mesma** visão do problema a ser resolvido. Nessa etapa, os desenvolvedores, juntamente com os clientes, tentam levantar e definir as necessidades dos futuros usuários do sistema a ser desenvolvido. Essas necessidades são geralmente denominadas **requisitos**.



Formalmente, um **requisito** é uma condição ou capacidade que deve ser alcançada ou possuída por um sistema ou um de seus componentes para satisfazer um contrato, padrão, especificação ou outros documentos formalmente impostos (MACIASZEK, 2000).

Normalmente os requisitos de um sistema são identificados a partir de um *domínio*. Um domínio corresponde à parte do mundo real que é **relevante**. Também é chamado de **domínio do problema** ou **domínio do negócio**.

O produto do levantamento de requisitos é o **documento de requisitos**, que declara os diversos tipos de requisitos do sistema. É normal esse documento ser escrito em uma notação informal (em linguagem natural).

As principais seções de um documento de requisitos são:

	Definição	Exemplos
Requisitos Funcionais	Definem as funcionalidades do sistema.	"O sistema deve permitir que cada professor realize o lançamento de notas das turmas nas quais lecionou." "O sistema deve permitir que um aluno realize a sua matrícula nas disciplinas oferecidas e um semestre letivo."
Requisitos Não Funcionais	Declaram as características de <i>qualidade</i> que o sistema deve possuir e que estão relacionadas às suas funcionalidades.	Confiabilidade, Desempenho, Portabilidade, Segurança e Usabilidade.
Requisitos Normativos	Declaração de restrições impostas sobre o desenvolvimento do sistema.	Adequação de custos e prazos, plataforma tecnológica, licenciamento, componentes de hardware e software a serem adquiridos etc.

Portanto, os **requisitos** devem ser expressos de uma maneira tal que a **equipe técnica** (leitores técnicos) deve entender o documento de requisitos de forma que possa obter as soluções técnicas apropriadas. **Clientes** (leitores não técnicos) devem entender esse documento para que possam priorizar o desenvolvimento dos requisitos, conforme as necessidades da organização em que trabalham.



Um ponto importante a respeito do documento de requisitos é que ele não deve conter informações sobre as **soluções técnicas** que serão adotadas para desenvolver o sistema.

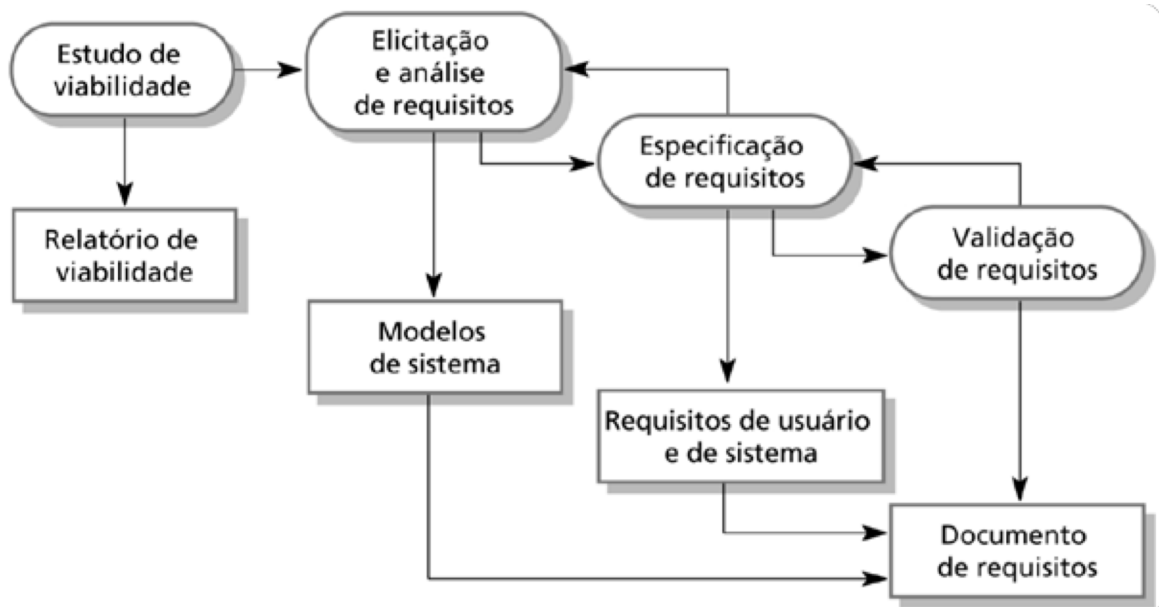
Lembre-se sempre: novos sistemas serão avaliados pelo seu grau de conformidade aos requisitos, não importa quão **complexa** a solução tecnológica tenha sido. **Requisitos** definem o **problema** a ser resolvido pelo sistema de software; eles não descrevem o software que resolve o problema.

O **documento de requisitos** estabelece o **escopo do sistema** (isto é, o que faz parte ou não do sistema). O escopo de um sistema muitas vezes muda durante o seu desenvolvimento.

Outro ponto importante sobre os **requisitos** é sua característica de **volatilidade**. Um **requisito volátil** é aquele que pode sofrer modificações durante o desenvolvimento do sistema.

A menos que o sistema a ser desenvolvido seja bastante simples e estático (características raras nos sistemas atuais), é praticamente **impossível** pensar em todos os detalhes a **princípio**.

Uma característica desejável em um documento de requisitos é ter os seus requisitos **ordenados** pelos usuários em função do seu grau de **prioridade**.



Análise

O termo **análise** corresponde a “quebrar” um sistema em seus componentes e estudar como eles interagem entre si com o objetivo de entender como esse sistema funciona. No contexto dos sistemas de software, esta é a etapa na qual os **analistas** realizam um estudo **detalhado** dos **requisitos** levantados na atividade anterior. A partir desse estudo, são construídos **modelos** para representar o sistema a ser construído.

O que costuma ocorrer na prática é exatamente o contrário: equipes de desenvolvimento que passam para a construção da solução sem antes terem definido completamente o problema. Portanto, os modelos construídos na fase de análise devem ser cuidadosamente **validados** e **verificados** pela validação e verificação dos modelos, respectivamente.

O objetivo da **validação** é assegurar que as necessidades do cliente estão sendo atendidas pelo sistema. Já a **verificação** tem o objetivo de analisar se os modelos construídos estão em conformidade com os requisitos definidos.

As principais **ferramentas da UML** para realizar análise são o **diagrama de casos de uso** e o **diagrama de classes** (para a modelagem de casos de uso e de classes, respectivamente). Outros diagramas da UML também utilizados na

análise são: **diagrama de interação**, **diagrama de estados** e **diagrama de atividades**.

Projeto

Na **fase de projeto**, determina-se “como” o sistema funcionará para **atender** aos requisitos, de acordo com os recursos **tecnológicos** existentes (a fase de projeto considera os aspectos físicos e dependentes de implementação).

A fase de projeto consiste em duas atividades principais: **projeto da arquitetura** (também conhecido como *projeto de alto nível*) e **projeto detalhado** (também conhecido como *projeto de baixo nível*).

Fase de Projeto	Definição
Projeto detalhado	Especificação da configuração de componentes do <i>software</i> - funções, classes, objetos e suas interconexões.
Projeto de arquitetura	Dão modeladas as colaborações entre os objetos de cada módulo com o objetivo de realizar suas funcionalidades. Também são realizados o projeto da interface com o usuário e o projeto de banco de dados, bem como são considerados aspectos de concorrência e distribuição do sistema.

Os **diagramas da UML** que podem ser utilizados nesta fase de projeto são: **diagrama de classes**, **diagrama de casos de uso**, **diagrama de interação**, **diagrama de estados** e **diagrama de atividades**.

Implementação

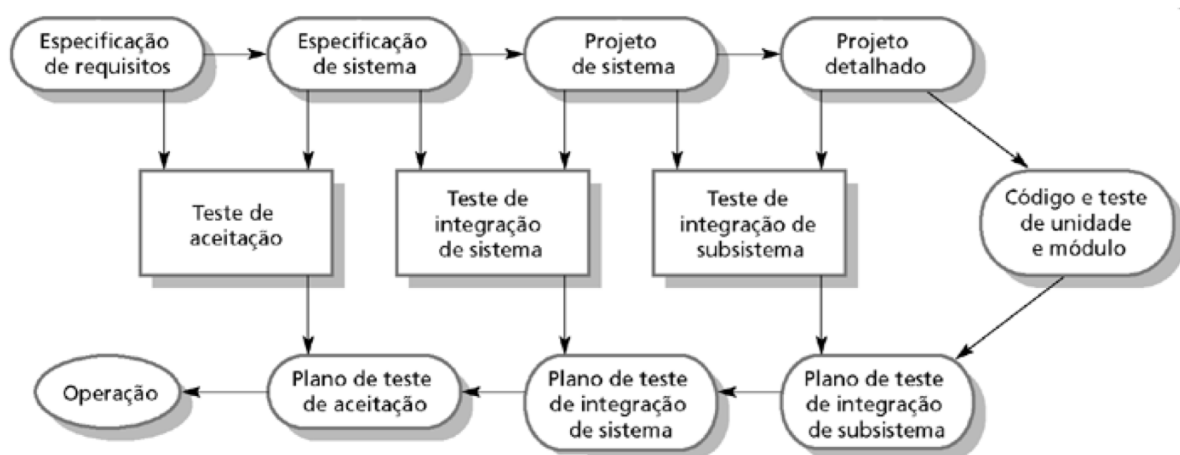
Na fase de **implementação**, o sistema é **codificado**, ou seja, ocorre a tradução da descrição computacional obtida na fase de projeto em **código executável** mediante o uso de uma ou mais **linguagens de programação**.

Teste

Diversas atividades de **teste** são realizadas para verificação do sistema construído, levando-se em conta a **especificação** feita nas fases de análise e de projeto. Um possível produto dessa fase são os **relatórios de testes**, que apresentam informações sobre erros detectados no software.

Testes de unidades são realizados sobre elementos do código-fonte do sistema. No contexto do desenvolvimento orientado a objetos, esses elementos correspondem a **classes** ou mesmo a **métodos** de uma classe. Testes de unidades normalmente são realizados pelos próprios desenvolvedores.

Nos **testes de integração**, quando uma operação de sistema é testada, também estão sendo testadas indiretamente as **interfaces** de cada objeto **participante**, assim como as **interações** entre eles.



Implantação

Na fase de implantação, o sistema é **empacotado**, distribuído e instalado no ambiente do usuário. Os manuais do sistema são escritos, os arquivos são carregados, os dados são importados para o sistema e os usuários treinados para utilizar o sistema corretamente. Em alguns casos, nesse momento também ocorre a migração de sistemas de software e de dados preexistentes.

O Componente Humano

O **desenvolvimento de software** é uma tarefa altamente cooperativa. Tecnologias complexas demandam **especialistas** em áreas específicas.

Uma **equipe** de desenvolvimento de software típica consiste em um gerente, analistas, projetistas, programadores, clientes e grupos de avaliação de qualidade.

Gerentes de Projeto	o gerente de projetos é o profissional responsável pela gerência ou coordenação das atividades necessárias à construção do sistema. Esse profissional também é responsável por fazer o orçamento do projeto de desenvolvimento, como, por exemplo, estimar o tempo necessário para o desenvolvimento do sistema, definir qual o processo de desenvolvimento, o cronograma de execução das atividades, a mão de obra especializada, os recursos de hardware e software etc
Analistas	O analista de sistemas é o profissional que deve ter conhecimento do <i>domínio do negócio</i> e entender seus problemas para que possa definir os requisitos do sistema a ser desenvolvido. Analistas devem estar aptos a se comunicar com especialistas do domínio para obter conhecimento acerca dos problemas e das necessidades envolvidas na organização empresarial. Outra característica necessária a um analista é a ética profissional. Esse profissional está frequentemente em contato com informações sigilosas e estratégicas dentro da organização na qual está trabalhando. Os analistas têm acesso a informações como preços de custo de produtos, margens de lucro aplicadas, algoritmos proprietários etc.
Projetistas	O projetista de sistemas é o integrante da equipe de desenvolvimento cujas funções são: (1) avaliar as alternativas de solução (da definição) do problema resultante da análise; e (2) gerar a especificação de uma solução computacional detalhada. A tarefa do projetista de sistemas é muitas vezes chamada de <i>projeto físico</i>
Arquiteto de Software	O objetivo desse profissional é elaborar a arquitetura do sistema como um todo. É ele quem toma decisões sobre quais subsistemas compõem o sistema como um todo e quais são as interfaces entre esses subsistemas.

Programadores	São os responsáveis pela <i>implementação do sistema</i> . É comum haver vários programadores em uma equipe de desenvolvimento. Um programador pode ser proficiente em uma ou mais linguagens de programação, além de ter conhecimento sobre bancos de dados e poder ler os modelos resultantes do trabalho do projetista.
Especialistas do domínio (Cliente)	Outro componente da equipe de desenvolvimento é o <i>especialista do domínio</i> , também conhecido como <i>especialista do negócio</i> . Esse componente é o indivíduo, ou um grupo deles, que possui conhecimento acerca da área ou do negócio em que o sistema em desenvolvimento estará inserido.
Avaliadores de Qualidade	O desempenho e a confiabilidade são exemplos de características que devem ser encontradas em um sistema de software de boa qualidade. Avaliadores de qualidade asseguram a adequação do processo de desenvolvimento e do produto de software sendo desenvolvido aos padrões de qualidade estabelecidos pela organização.

Exercícios da Aula

1. Um sistema automático de emissão de passagens vende passagens de trem. A partir de uma lista de possíveis destinos, os usuários escolhem seu destino e apresentam um cartão de crédito e um número de identificação pessoal. Os destinos possíveis devem ser organizados de modo a facilitar a escolha. Após a escolha do destino, o sistema deve responder prontamente se há espaço disponível no trem. A passagem é emitida e o custo dessa passagem é incluído em sua conta do cartão de crédito. Quando o usuário pressiona o botão para iniciar, uma tela de menu com os possíveis destinos é ativada, juntamente com uma mensagem para que o usuário selecione um destino. Uma vez selecionado um destino, pede-se que os usuários insiram seu cartão de crédito. A validade do cartão é checada e o usuário então deve fornecer um número de identificação pessoal. Quando a transação de crédito é validada, a passagem é emitida. O formato do bilhete de passagem deve seguir ao padrão definido pelo Sistema Nacional de Tráfego Ferroviário

Solução:

RF1: Quando o usuário pressiona o botão para iniciar, uma tela de menu com os possíveis

destinos é ativada, juntamente com uma mensagem para que o usuário selecione um destino.

RF2: Uma vez selecionado um destino, pede-se que os usuários insiram seu cartão de crédito.

RF3: O sistema deve informar se existem vagas no destino escolhido

RF4: A validade do cartão é checada e o usuário então deve fornecer um número de identificação pessoal.

RF5: Quando a transação de crédito é validada, a passagem é emitida e o custo dessa

passagem é incluído em sua conta do cartão de crédito.

RNF1: As telas devem facilitar a escolha do destino

RNF2: O tempo de resposta sobre vaga no trem deve ser adequado

RNF3: O formato do bilhete de passagem deve seguir ao padrão definido pelo Sistema

Nacional de Tráfego Ferroviário".

2. Suponha que um Analista de TI, participando da etapa de análise de requisitos de um sistema de emissão de certidão negativa para o TRF4, tenha elencado os requisitos apresentados abaixo: Utilizar interface responsiva para que possa ser executado em dispositivos móveis e na web.2. Validar o tipo de certidão solicitado.3. Emitir certidão negativa após verificação de situação do requerente.4. Solicitar o CPF do requerente.5. Responder ao clique único do usuário em qualquer botão da interface.6. Validar o CPF do requerente.7. Restaurar os dados automaticamente após falhas não programadas.8. Solicitar o nome do requerente.9. Oferecer dois tipos de certidão: para fins gerais e para fins eleitorais.10. Emitir aviso de impossibilidade de emissão da certidão.

3. Um Analista de Informática levantou os requisitos para desenvolver um sistema de gestão. Dentre os requisitos levantados:

- a. o sistema deve apresentar a tela de login e senha antes de cada transação e validar o acesso com base nas políticas de segurança organizacional.
 - b. o sistema deve estar disponível para a diretoria em tempo integral, ou seja, 24 × 7.
 - c. o tempo de resposta de uma consulta da alta administração não pode exceder a 5 milissegundos.
 - d. cada Diretor que usa o sistema deve ser identificado apenas por sua matrícula de cinco dígitos seguidos do código de segurança.
 - e. o sistema deverá gravar um log de autenticação a cada transação completada, contendo a identificação do usuário, data e equipamento utilizado.
 - f. os backups do sistema deverão ser feitos diariamente a fim de evitar a eventual perda de dados sem capacidade de recuperação.
4. Escolha um dos temas abaixo e crie um documento de requisitos para o seu sistema. Inclua requisitos funcionais e não funcionais.
- Oficina Mecânica
 - Aluguel de Filmes
 - Disque de Táxi
 - Entrega de Encomendas
 - Assinatura de Revista
 - Sistema Escolar
 - Lan House
 - Reserva de Buffet

Um aluno será o CLIENTE, os demais são os entrevistadores/desenvolvedores.