

Projetos de Sistemas de Informação





# **Material Teórico**



#### Responsável pelo Conteúdo:

Prof. Ms. Fábio Peppe Beraldo

#### Revisão Textual:

Profa. Ms. Fatima Furlan.

## UNIDADE

## Modelos de Desenvolvimento de Software



- Introdução
- Modelos de Desenvolvimento de Software





O objetivo desta Unidade é habilitar o aluno a projetar melhorias de qualidade em produtos e serviços; identificar as vantagens da modelagem de sistemas; usar o modelo de conceituar e construir sistemas; fazer um estudo interdisciplinar da utilização dos modelos; fazer a modelagem de sistemas, análise e esforços de design; fazer a modelagem e simulação de sistemas, tais como a dinâmica do sistema e sistemas de linguagem de modelagem específica.



## Atenção

Para um bom aproveitamento do curso, leia o material teórico atentamente antes de realizar as atividades. É importante também respeitar os prazos estabelecidos no cronograma.

## Contextualização

A descrição do processo se destina a cobrir as aplicações de software completos e componentes de software individuais. O desenvolvimento de software envolve entre outras coisas:

- » Os elementos gerais da engenharia de software;
- » As fases de projeto significativo de desenvolvimento de software;
- » O pessoal envolvido;
- » As atividades em curso, independentemente da fase;
- » Modelos de processos de software;
- » A importância de um processo disciplinado.



## Introdução



Como em qualquer outra área da engenharia, a engenharia de software também possui alguns modelos estruturados de desenvolvimento. Nessa unidade, vamos estudar, por meio de uma visão genérica, as diferentes metodologias de desenvolvimento de software adotadas por empresas de softwares contemporâneos.

#### **Curtain Raiser**

Como qualquer outro conjunto de produtos de engenharia, os softwares também são destinados para o cliente; ou são impulsionados pelo mercado ou impulsionam o mercado. O termo "Satisfação do Cliente" é um chavão antigo, hoje a ideia de "Co-Criação" é a nova palavra de ordem. Produtos que não são amigáveis ao cliente não têm lugar no mercado, embora eles sejam projetados usando a melhor tecnologia. A experiência do produto e a participação do cliente na criação do produto é tão importante quanto a tecnologia interna do produto.

## Pesquisa de Mercado

O estudo de mercado é feito para identificar a necessidade de um cliente potencial. Deve ser estudado nessa pesquisa a necessidade existente e as possíveis necessidades potenciais que existem em um segmento da sociedade. O estudo de mercado é feito com base em uma série de suposições, as premissas são fatores cruciais no desenvolvimento ou criação de desenvolvimento de um produto. Suposições irreais podem causar uma queda livre por todo o empreendimento, embora os pressupostos sejam abstratos, deve haver um movimento para desenvolver hipóteses concretas para chegar a um produto de sucesso.

## Pesquisa e Desenvolvimento

Uma vez que a pesquisa de mercado é realizada, a necessidade do cliente é dada à divisão de Pesquisa & Desenvolvimento (Research & Development - R&D) para conceituar um sistema que possa resolver as necessidades do cliente de uma maneira mais pratica e ou de forma mais abrangente que os concorrentes. Uma vez que o sistema conceitual é desenvolvido e testado em um ambiente hipotético, a equipe de desenvolvimento assume o seu controle, adota uma das metodologias de desenvolvimento de software (demonstradas a seguir), desenvolve o sistema proposto e o entrega ao cliente.

A divisão de Vendas e Marketing começa a vender o software para os clientes e, simultaneamente, trabalha para desenvolver um segmento de nicho com clientes que poderiam comprar o software. Além disso, a divisão também passa o feedback dos clientes e da divisão de R&D para os desenvolvedores para tornar possíveis os acréscimos de valor ao produto.

No processo de desenvolvimento de software, a empresa terceiriza as atividades non-core (não principais) para outras empresas que se especializam nessas atividades, isso acelera, em grande parte, o processo de desenvolvimento de software.

## Modelos de Desenvolvimento de Software



Veremos agora alguns dos principais modelos adotados, por muitas empresas, no desenvolvimento de softwares.

- a. Ciclo de Vida do Desenvolvimento de Software (SDLC);
- b. Modelo de Prototipagem;
- c. Modelo de Desenvolvimento de Aplicativos Rápidos;
- d. Modelo de Montagem/Construção de Componentes.

#### I) Ciclo de Vida do Desenvolvimento de Software (SDLC)

Esse modelo é também conhecido como Modelo de Ciclo de Vida Clássico, Modelo de Sequência Linear e Modelo Waterfall. Esse modelo trabalha com as seguintes atividades.

#### a) Modelagem e Engenharia de Sistemas/Informação

Como o software geralmente faz parte de um sistema maior, o trabalho começa por estabelecer os requisitos para todos os elementos do sistema e, em seguida, alocar algum subconjunto desses requisitos ao software. Essa visão do sistema é essencial quando o software deve fazer interface com outros elementos, como hardware, pessoas e outros recursos. O "Sistema" é o requisito básico, porém crítico para a existência de software de qualquer entidade. Em alguns casos, para extrair o máximo de trabalho, o sistema deve ser reengenhado para assim se aproximar de um sistema ideal buscado pela equipe de desenvolvimento de software.

## b) Análise de Requisitos de Software

Esse processo é conhecido como estudo de viabilidade. Nesta fase, a equipe de desenvolvimento visita o cliente e estuda seu sistema/empresa. Nesse momento, é investigada a necessidade de automação do software. Ao final do estudo de viabilidade, a equipe fornece um documento que contém as diferentes recomendações específicas para o sistema candidato, incluindo também as atribuições de pessoal, custos, cronograma do projeto, prazos etc.

O processo de levantamento de requisitos é intensificado e concentrado especialmente em software. Para compreender a natureza do programa(s) a ser(erem) construído(s), o engenheiro de sistema ou "analista" deve compreender o domínio da informação para o software, bem como a função desejada, comportamento, desempenho e interface. O propósito essencial dessa fase é encontrar a necessidade e definir o problema que precisa ser resolvido. Mais para frente teremos uma unidade toda para estudarmos como ocorrem as análises de requisitos.

## c) Projeto e Análise de Sistemas

É nesta fase do processo de desenvolvimento de software que a estrutura global do software e suas nuances são definidas. Em termos de tecnologia cliente/servidor, define-se o número de níveis necessários para a arquitetura do projeto, o projeto de banco de dados, o projeto da



estrutura de dados etc. Um modelo de desenvolvimento de software é assim criado. O projeto e a analise são cruciais em todo o ciclo de desenvolvimento, qualquer falha na fase de projeto pode sair muito cara para se resolver na fase posterior do desenvolvimento de software, portanto, muito cuidado deve ser tomado. Podemos afirmar que o sistema lógico do produto (software) é desenvolvido nessa fase.

#### d) Geração do Código

O projeto deve ser traduzido em um formato legível para a máquina, também conhecido como, linguagem de máquina. O objetivo da geração de código é executar esta tarefa. Se a concepção é feita de uma maneira pormenorizada, a geração de códigos pode ser realizada sem grande complicação usando ferramentas de programação, como compiladores, interpretadores, depuradores etc... para gerar o código. Diferentes linguagens de programação de alto nível, como C, C++, Pascal, Java são usadas para a codificação. A linguagem de programação é definida a partir do tipo de aplicação/software a ser criado e da necessidade da empresa.

#### e) Teste

Uma vez que o código é gerado, seu teste começa. Diferentes metodologias de teste estão disponíveis para descobrir os erros que foram cometidos durante as fases anteriores. Ferramentas de teste de diferentes metodologias estão disponíveis. Algumas empresas costumam encomendar suas próprias ferramentas de testes que são feitas sob medida para as suas próprias operações de desenvolvimento.

## f) Manutenção

Depois de ser desenvolvido, o software definitivamente sofre mudanças. Pode haver muitas razões para que as mudanças ocorram, uma delas pode ser devido à entrada inesperada de alguns valores no sistema (alterações do sistema/empresa). Além disso, as alterações no sistema podem afetar diretamente as operações de software que deve ser desenvolvido para acomodar as mudanças que podem acontecer durante o período pós implementação.

## II) Modelo de Prototipagem

Na verdade, o Modelo de Prototipagem é a versão cíclica do modelo linear. Nesse modelo, uma vez que a análise de requisitos e o projeto para um protótipo são feitos, o processo de desenvolvimento começa. A partir do momento em que o protótipo é criado, ele é dado ao cliente para avaliação. O cliente testa o pacote e dá o seu feedback para o desenvolvedor que refinará o produto de acordo com a expectativa exata do cliente. Depois de um número finito de iterações, o pacote de software final é dado ao cliente. Nessa metodologia, o software é desenvolvido como resultado do vai-e-vem periódico de informações entre o cliente e o desenvolvedor.

O modelo de desenvolvimento por prototipagem é o mais popular na indústria de TI. Muitos dos produtos de software bem-sucedidos têm sido desenvolvidos usando esse modelo.

## III) Modelo de Desenvolvimento de Aplicativos Rápidos

O Modelo de Desenvolvimento de Aplicativos Rápidos (Rapid Application Development – RAD) é um processo de desenvolvimento de software linear que enfatiza um ciclo de desenvolvimento extremamente curto. O modelo RAD é uma adaptação de "alta velocidade" do modelo sequencial linear no qual o desenvolvimento rápido é conseguido através de uma abordagem de construção baseada em componentes. Usado principalmente para aplicações de sistemas de informação, a abordagem RAD engloba as seguintes fases: (mencionados em outras unidades)

#### a) Modelagem de Negócios

O fluxo de informações entre as funções de negócio é modelado de forma que responda às seguintes perguntas:

- » Que informação dirige o processo de negócio?
- » Que informação é gerada?
- » Quem gera essa informação?
- » Para onde vai a informação?
- » Quem processa a informação?

#### b) Modelagem de Dados

O fluxo de informação é definido como parte da fase de modelagem de negócios refinado em um conjunto de objetos de dados que são necessários para apoiar o negócio. As características ou atributos de cada objeto são identificados e as relações entre esses objetos são definidas.

## c) Modelagem de Processos

Os dados definidos na fase de modelagem de dados são transformados para conseguir o fluxo de informação necessário para implementar uma função do negócio. A correta modelagem de processos para fazer o processamento desses dados é de máxima importância para adicionar, alterar, excluir ou recuperar dados dentro/para os softwares criados.

## d) Geração de Aplicativos

O modelo RAD assume o uso das ferramentas como VB, C++, Delphi etc... ao invés de criar um software utilizando linguagens de programação não convencionais e muitas vezes de abrangência menor no quesito criação e desenvolvimento. O modelo RAD trabalha para reutilizar os componentes do programa existentes ou criar componentes reutilizáveis. Em todos os casos, as ferramentas automatizadas são usadas para facilitar a construção do software.

## e) Teste e Aprimoramento

Uma vez que o modelo RAD enfatiza reutilização, muitos dos componentes do programa já foram testados. Isso minimiza o tempo de teste e desenvolvimento.



## IV) Modelo de Montagem/Construção de Componentes

Algumas tecnologias fornecem um quadro técnico para um modelo de processo baseado em componentes para a engenharia de software. O paradigma de orientação a objetos (tema discutido em unidade futura) enfatiza a criação de classes que encapsulam dados e do algoritmo que são usados para manipular os dados. Se for bem concebido e implementado, as classes orientadas a objeto são reutilizáveis em diferentes arquiteturas de sistemas. Esse modelo leva a reutilização de software e a integração ou montagem dos componentes de software já existentes acelera o processo de desenvolvimento.

## **Material Complementar**

PRESSMAN, R. S. **Engenharia de Software.** 6<sup>a</sup> ed. Porto Alegre: Grupo A, 2010. (e-book)

STEPHEN R. S. **Engenharia de Software.** 8<sup>a</sup> ed. Porto Alegre: Grupo A, 2008. (e-book)

PADUA, W. **Engenharia de Software**, 3<sup>a</sup> ed. Rio de Janeiro: Grupo GEN, 2008. (e-book)

KALINOVSKY, A. **Java Secreto:** técnicas de descompilação, patching e engenharia reversa. São Paulo: Pearson, 2009. (e-book)

PFLEEGER, S. L. **Engenharia de Software:** teoria e prática - 2º ed. São Paulo: Pearson, 2009. (e-book)



## Referências

#### Bibliografia Fundamental

SOMMERVILLE, I. **Engenharia de Software.** 9ª ed. São Paulo: Addison-Wesley, 2007. (e-book)

#### Bibliografia Básica

SCHACH, S. R. **Engenharia de Software:** Os Paradigmas Clássicos & Orientado a Objetos. 7ª ed. São Paulo: Bookman, 2009.

WAZLAWICK, R. S. Analise e Projeto de Sistemas de Informação Orientados a Objetos.  $2^a$  ed. Rio de Janeiro: Elsevier, 2011.

Anotações	



www.cruzeirodosulvirtual.com.br Campus Liberdade Rua Galvão Bueno, 868 CEP 01506-000 São Paulo SP Brasil Tel: (55 11) 3385-3000











