# **Engenharia de Software**



Conceitos e Metodologias para Desenvolvimento de Software Cascata, Prototipação, Espiral e RUP

Prof. MSc. Edilberto Silva

prof.edilberto.silva@gmail.com http://www.edilms.eti.br

## Paradigmas de Engenharia de Software



- Principais paradigmas:
  - Cascata.
  - Prototipação.
  - Espiral
  - Fases Genéricas [Pressman]
  - RUP Rational Unified Process
- Ágeis
  - XP eXtreme Programing
  - OpenUP
  - SCRUM

#### **Modelo Cascata** Analista de Negócios Requisitos Modelo em Cascata Analista Análise funcional Chefe de projeto Projeto Analistas de Sofwietura sistemas, Codificação programadores Teste Manutenção Prof. Edilberto Silva – www.edilms.eti.br

#### **Cascata** Especificação, Analise Requisitos Arquitetura, Desenho Modelos Codificação, Implementação Desenvolvimento Testes, Teste validação, Qualidade Instalação, Suporte, Implantação Manutenção Prof. Edilberto Silva – www.edilms.eti.br

### Etapas do Modelo Cascata



#### Requisitos

- Coleta os requisitos do software.
- Gera o documento de especificação do sistema que serve de base para o orçamento, cronograma, esforço, ferramentas a serem utilizadas, etc.

#### Análise

- Compreensão clara e precisa do domínio do problema e das funcionalidades do software.
- Levantamento e revisão em conjunto com representantes do cliente, usuários chaves e outros especialistas da área de aplicação

## **Etapas do Modelo Cascata**



#### Projeto

 Concentra-se na definição das estruturas de dados, arquitetura do software, detalhes procedimentais e caracterização da interface.

#### Codificação

- Tradução do projeto para uma linguagem legível para a máquina.
- Se o projeto for bem detalhado essa tarefa pode ser automatizada.

## **Etapas do Modelo Cascata**



- Teste
  - Inicia-se logo após a geração do código.
  - Visa garantir que uma entrada do programa produz o resultado esperado.
- Manutenção
  - Ocorre em função de:
    - Ocorrência de erros
    - Adaptações para acomodar mudanças externas
    - Acréscimos funcionais
    - Problemas de desempenho

#### **Modelo Cascata**

#### **Vantagens**

- Padroniza os métodos para análise, projeto, codificação, testes e manutenção.
- Etapas semelhantes às etapas genéricas aplicáveis a todos os paradigmas.

#### **Desvantagens**



- Projetos reais raramente seguem o fluxo seqüencial que esse modelo propõe. Sempre ocorre alguma interação e/ou superposição.
- Dificilmente os clientes são capazes de relacionar todos os requisitos de uma só vez no início do projeto
- Maioria dos programas só estará disponível quando o cronograma já está bastante adiantado.
- Dificuldades para se introduzir alterações quando o processo está avançado.

## Prototipação

- Criação de um modelo do software que será implementadena
- Facilita a definição dos objetivos do software.
- Retrata a interação entre o usuário e o software.

#### **Etapas**

Coleta e refinamento dos requisitos.

Definição dos objetivos globais.

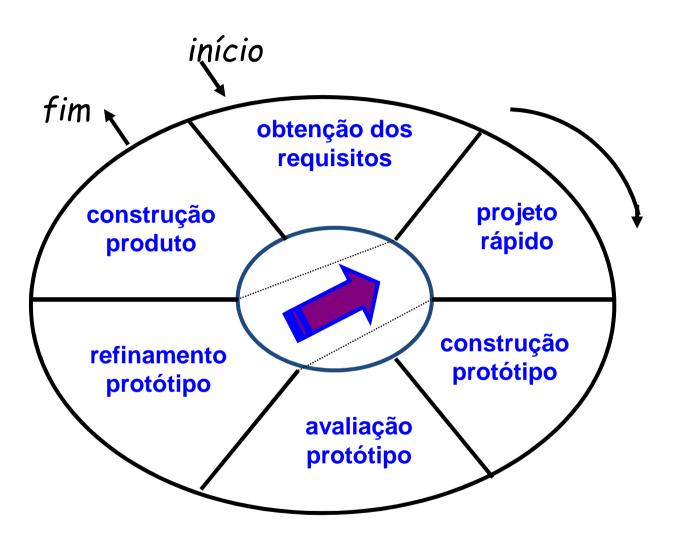
Projeto rápido.

- Identificação interações entre usuário e software.
- Concentra-se nas entradas e saídas do software.

Construção do protótipo.

## Etapas da Prototipação





## Modelo Prototipação



#### **Vantagens**

- Melhora a qualidade da especificação do software a ser desenvolvido, contribuindo para uma queda nos custos de desenvolvimento e manutenção.
- Antecipa o treinamento dos usuários.
- Partes do protótipo podem ser aproveitadas no desenvolvimento do sistema.

#### **Desvantagens**

- O custo na maioria dos casos é considerado muito alto.
- O cliente tende a confundir o protótipo com uma versão do sistema.

## **Modelo Espiral**

 Aproveita as melhores características do modelo cascata e da prototipação.

Coleta inicial dos requisitos e planejamento do projeto

Planejamento baseado nos comentários do cliente

 Acrescentando um novo elemento: a análise de riscos.

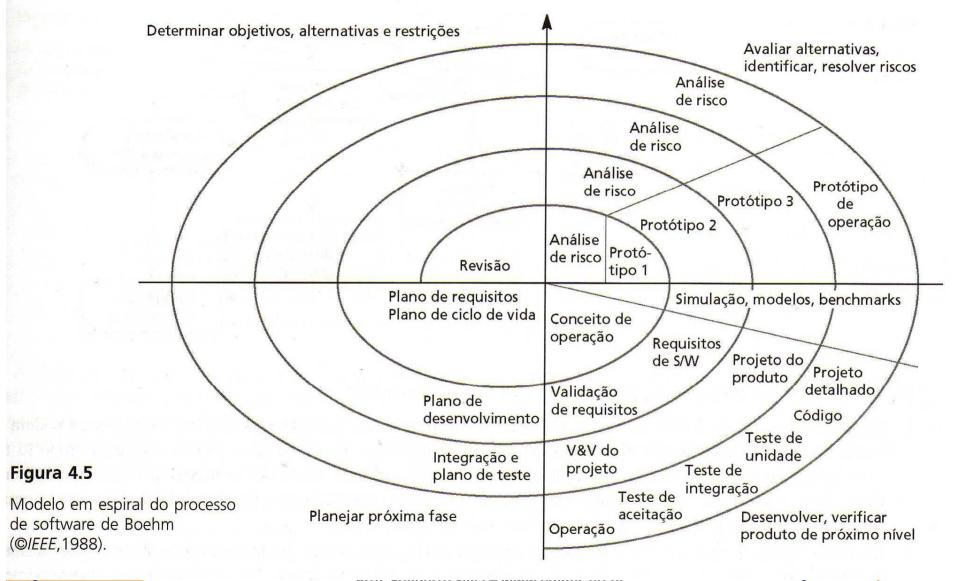
Avaliação do cliente

Planciamento Analise dos riscos Análise dos riscos baseada nos requisitos iniciais Análise dos riscos baseada na reação do cliente Decisão de prosseguir não prosseguir Na direção de um sistema concluido Protótipo de software inicial Protótipo no nivel seguinte Sistema construido Engenharia Avaliação do cliente pela Engenharia

Prof. Edilberto Silva – www.edilms.eti.br

## **Modelo Espiral**





## **Etapas do Modelo Espiral**



- Planejamento.
  - Definição dos objetivos, alternativas e restrições.
- Análise dos Riscos.
  - Análise de alternativas e identificação dos riscos sob o ponto de vista técnico e de gerência.
- Engenharia.
  - Desenvolvimento do produto.
- Avaliação do Cliente.
  - Avaliação dos resultados da engenharia.

## **Modelo Espiral**



#### **Vantagens**

- Modelo evolutivo possibilita uma maior integração entre as fases e facilita a depuração e a manutenção do sistema.
- Permite que o projetista e cliente possa entender e reagir aos riscos em cada etapa evolutiva.

#### **Desvantagens**

- Avaliação dos riscos exige muita experiência.
- O modelo é relativamente novo e não tem sido amplamente utilizado.

## Fases Genéricas: Definição

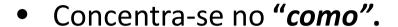


- Concentra-se no "quê".
  - Que informações devem ser processadas?
  - Que funções devem ser desempenhadas?
  - Quais são as exigências fundamentais do software?
  - Quais são as restrições do projeto?

#### **Etapas**

- Análise do Sistema:
  - Define o papel que o software deverá desempenhar.
- Planejamento do Projeto de Software:
  - Analisa os riscos, aloca os recursos, estima os custos e define as tarefas que serão realizadas.
- Análise de Requisitos
  - Define detalhadamente o domínio da informação e a função do software.

### Fases Genéricas: Desenvolvimento





- Como projetar as estruturas de dados e a arquitetura do software?
- Como implementar os detalhes procedimentais?
- Como traduzir o projeto para uma linguagem de programação?
- Como realizar os testes?

#### **Etapas**

- Projeto de Software:
  - Traduz os requisitos do software em um conjunto de representações
- Codificação:
  - Converte as representações do projeto em uma linguagem artificial executada pelo computador.
- Realização de Testes de Software:
  - Procura por defeitos de função, lógica e implementação do software.

## Fases Genéricas: Manutenção

Concentra-se nas mudanças sofridas pelo software.



#### Tipos de Mudanças:

- Correção
  - Corrige defeitos encontrados pelo cliente (manutenção corretiva).
- Adaptação
  - Acomoda mudanças ocorridas no ambiente do software (manutenção adaptativa).
- Melhoramento Funcional
  - Estende o software para além de suas exigências funcionais originais (manutenção perfectiva).

## Desenvolvimento Incremental

- Em vez de entregar o sistema como um todo, o desenvolvimento e a entrega são divididos em incrementos, com cada incremento entregando parte da funcionalidade requerida
- Requisitos dos usuários são priorizados e os requisitos de mais alta prioridade são incluídos nas iterações iniciais
- Uma vez que o desenvolvimento de um incremento é iniciado, os requisitos são "congelados". Embora os requisitos possam continuar a evoluir para incrementos posteriores

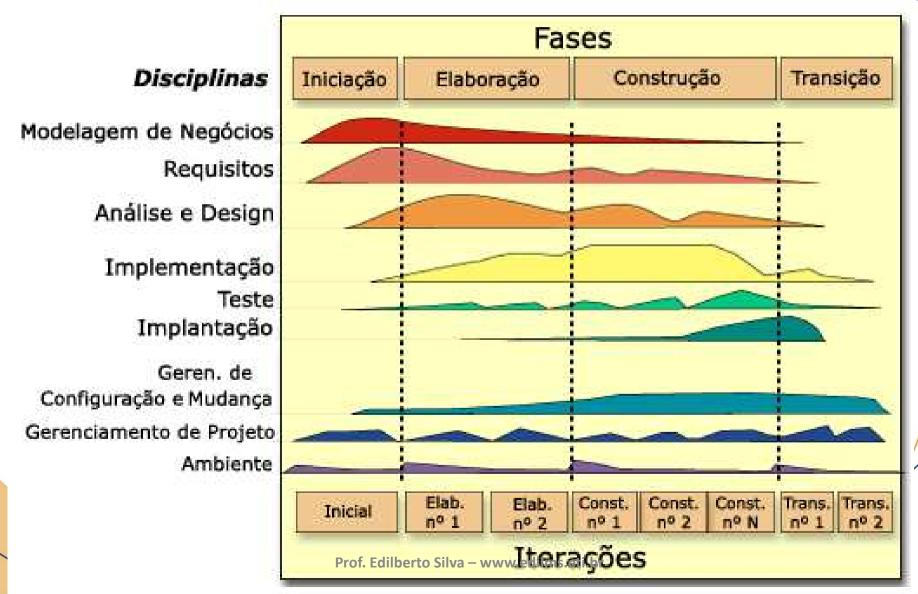
### Visão Geral do RUP



### Objetivo

 Depois desta aula você terá uma visão geral do RUP (uma metodologia para desenvolvimento de software), incluindo suas características e seus componentes principais.

# Desenvolvimento Iterativo e Incremental: RUP – Rational Unified Process senac



## O que é o RUP?



- O nome é uma abreviação de Rational Unified Process
  - mas na verdade é
    - Processo + Métodos + Linguagem (UML)
  - e os autores argumentam que é
    - Framework para gerar processos

## O que é o RUP?



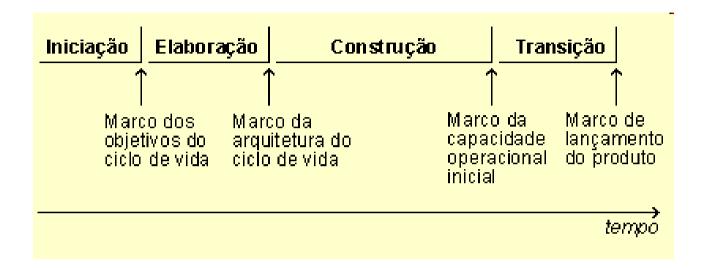
- Conjunto de atividades
  - bem definidas
  - com responsáveis
  - com artefatos de entrada e saída
  - com dependências entre as mesmas e ordem de execução
  - com modelo de ciclo de vida
  - descrição sistemática de como devem ser realizadas
  - guias (de ferramentas ou não), templates
  - utilizando diagramas de UML

## Características Principais do RUBenac

- O desenvolvimento de sistemas seguindo o RUP é
  - Iterativo e incremental
  - Guiado por casos de uso (use cases)
  - Baseado na arquitetura do sistema

#### **RUP – Rational Unified Process**





- Concepção (define o escopo do projeto)
- Elaboração (define os requisitos e a arquitetura)
- Construção (desenvolve o sistema)
- Transição (implanta o sistema)

## O RUP é iterativo e incrementa

 O ciclo de vida de um sistema consiste de quatro fases:

concepção	elaboração	construção	transição
-----------	------------	------------	-----------

tempo

- Concepção (define o escopo do projeto)
- Elaboração (detalha os requisitos e a arquitetura)
- Construção (desenvolve o sistema)
- Transição (implanta o sistema)

## O RUP é iterativo e incrementa

Cada fase é dividida em iterações:

Inception	Elaboration		Construction			Transition	
Preliminary iteration	Architect. iteration	Architect. iteration	Devel iteration	Devel iteration	Devel iteration	Transition iteration	Transition iteration

**Minor Milestones: Releases** 

Prof. Edilberto Silva – www.edilms.eti.br

## O RUP é iterativo e incrementa

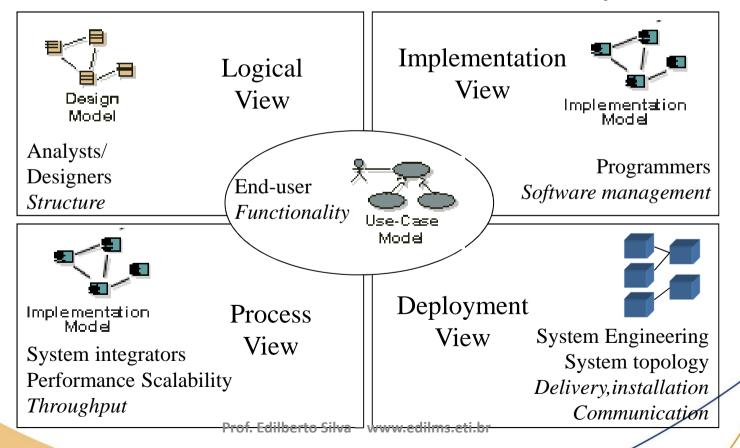
- Cada iteração
  - é planejada
  - realiza uma seqüência de atividades (de elicitação de requisitos, análise e projeto, implementação, etc.) distintas
  - geralmente resulta em uma versão executável do sistema
  - é avaliada segundo critérios de sucesso previamente definidos

# O RUP é baseado na arquitetura do sistema

- Arquitetura
  - visão geral do sistema em termos dos seus subsistemas e como estes se relacionam
- A arquitetura é prototipada e definida logo nas primeiras iterações
- O desenvolvimento consiste em complementar a arquitetura
- A arquitetura serve para definir a organização da equipe de desenvolvimento e identificar oportunidades de reuso

# O RUP é baseado na arquitetura do sistema

• Idealmente, tem-se 5 visões da arquitetura



## Organização do RUP



- Fluxos de atividades
- Atividades
  - passos
  - entradas e saídas
  - guias (de ferramentas ou não), templates
- Responsáveis (papel e perfil, não pessoa)
- Artefatos

### **Marcos**



Concep	oção	Elaboração	Construçã	Trans	ição	
1			<b>I</b>	 		
	objeti	vos do arquit	etura do c	Marco da apacidade peracional	Marco de lançamento do produto	
tempo				inicial		

O projeto poderá ser anulado ou completamente repensado caso o marco não seja atingido.

## Marco dos Objetivos do Ciclo de Vida



 Análise dos objetivos do ciclo de vida do projeto e tomada de decisão de prosseguir com o projeto ou cancelá-lo

#### Critérios de Avaliação

- Consentimento dos envolvidos sobre a definição do escopo e as estimativas de custo/programação.
- Consenso de que o conjunto correto de requisitos foi capturado e de que existe uma compreensão compartilhada desses requisitos.
- Consenso de que as estimativas de custo/programação, as prioridades, os riscos e o processo de desenvolvimento são adequados.
- Todos os riscos foram identificados e existe uma estratégia atenuante para cada um.

## Marco da Arquitetura do Ciclo de Vida

 Exame nos objetivos e o escopo do sistema, a opção de arquitetura e a resolução dos principais riscos.

#### Critárias de Avaliação

- Os planos de iteração para a fase de construção têm detalhes e fidelidade suficientes para permitir o avanço do trabalho.
- Os planos de iteração para a fase de construção são garantidos por estimativas confiáveis.

- Todos os envolvidos concordam que a visão atual poderá ser atendida se o plano atual for executado para desenvolver o sistema completo, no contexto da arquitetura atual.
- A despesa real em oposição à despesa planejada com recursos é aceitável.

# Marco da capacidade operacional inicial

- Produto pronto para ser passado para a Equipe de Transição.
- Toda a funcionalidade desenvolvida e os testes alfa (se houver algum) foram concluídos.
- Manual do usuário desenvolvido e uma descrição do release atual.

#### Critérios de Avaliação

- Envolvem respostas para as questões:
  - Este release do produto é estável e desenvolvido o suficiente para ser implantado na comunidade de usuários?
  - Todos os envolvidos estão prontos para a transição para a comunidade de usuários?
  - As despesas reais com recursos ainda s\u00e3o aceit\u00e1veis se comparadas com as planejadas?

## Marco do Release do Produto



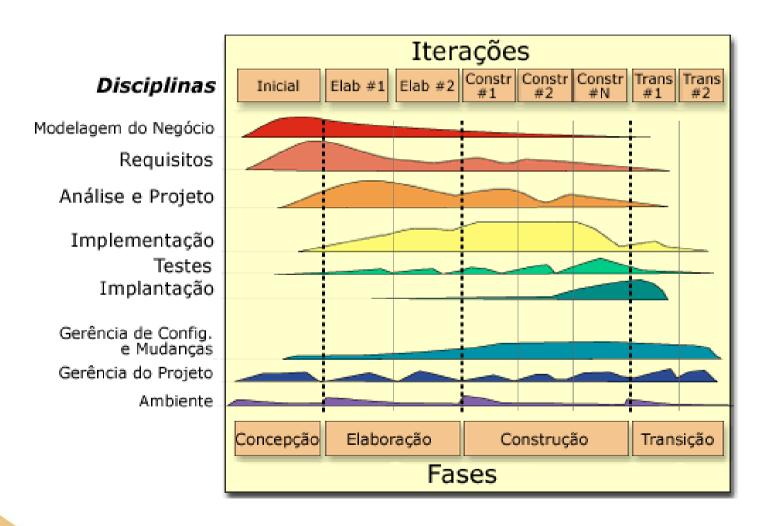
- Decisão se os objetivos foram atendidos e se outro ciclo de desenvolvimento deve ser iniciado.
- Pode coincidir com o fim da fase de iniciação do próximo ciclo.
- O Marco do Release do Produto é o resultado da conclusão com êxito dos Artefatos

#### Fluxos de Atividades do RUP



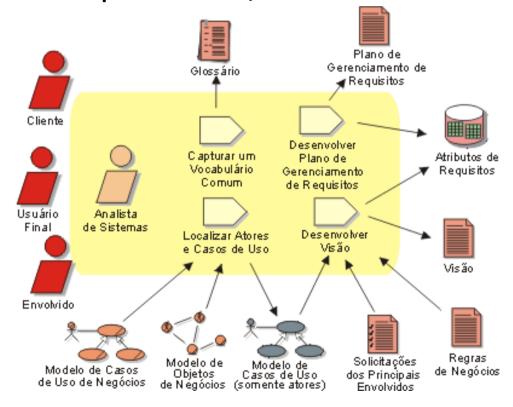
- Agrupam atividades correlacionadas
- Fluxos de atividades básicos:
  - modelagem do negócio
  - requisitos
  - análise e projeto
  - implementação
  - testes
  - distribuição
- Fluxos de atividades de suporte:
  - gerência de configuração e mudanças
  - gerência do projeto
  - configuração do ambiente

# Fases, Iterações e Fluxos de Atividades



### Responsáveis, Atividades e Artefatos

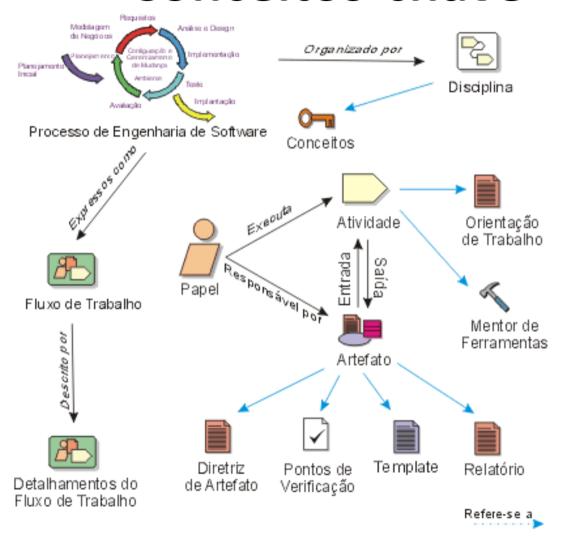
 Os fluxos de atividades do RUP são descritos através de responsáveis, atividades e artefatos



Fonte: Rational

### **Conceitos-chave**





# Modelagem do Negócio



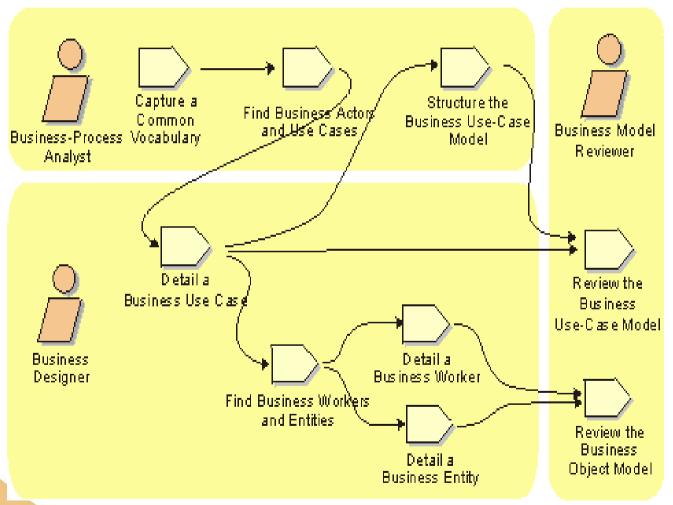
### Objetivos:

- descobrir "o problema por trás do problema"
- entender a estrutura e dinâmica da organização
- assegurar que os clientes, usuários e desenvolvedores têm a mesma visão do negócio
- descobrir os requisitos do sistema necessários para suportar o negócio

#### Desenvolver um modelo do negócio

### Modelagem do Negócio





- Glossário
- Modelo de Casos de Uso do Negócio
- Modelo de Objetos do Negócio

# Requisitos

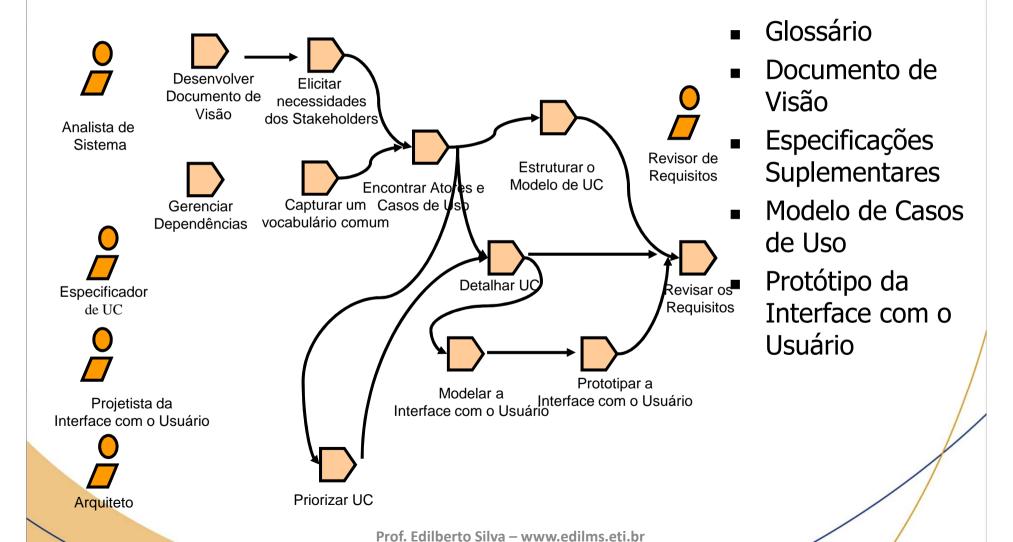


#### Objetivos:

- descrever o quê o sistema deve fazer, em acordo com o cliente e usuários
- definição de como gerenciar escopo e mudanças de requisitos
- delimitar o escopo do sistema e prover uma base para o planejamento das iterações
- definir a interface com o usuário

### Requisitos





# **Análise e Projeto**



#### Objetivos:

- transformar os requisitos em um modelo para implementação do sistema
- encontrar uma arquitetura robusta para o sistema

#### Análise:

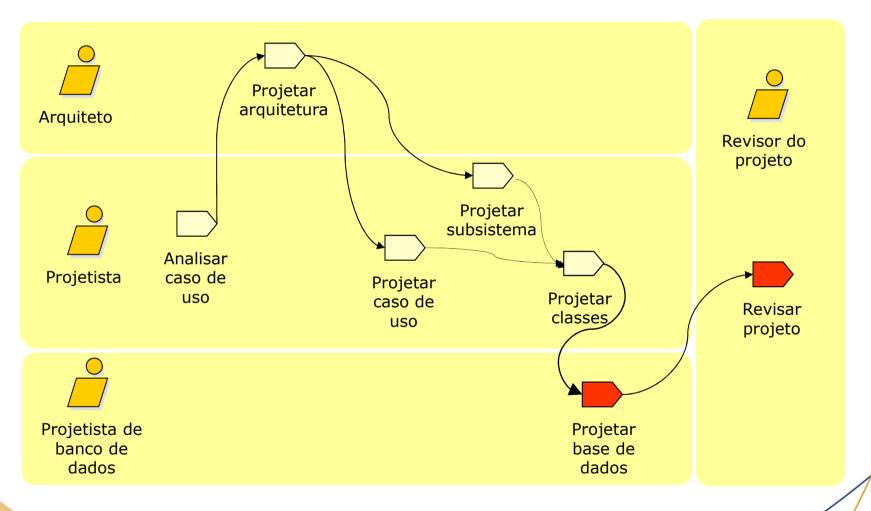
assegura que os requisitos funcionais são tratados

#### • Projeto:

 adapta os resultados da análise aos requisitos não funcionais e ambiente de implementação

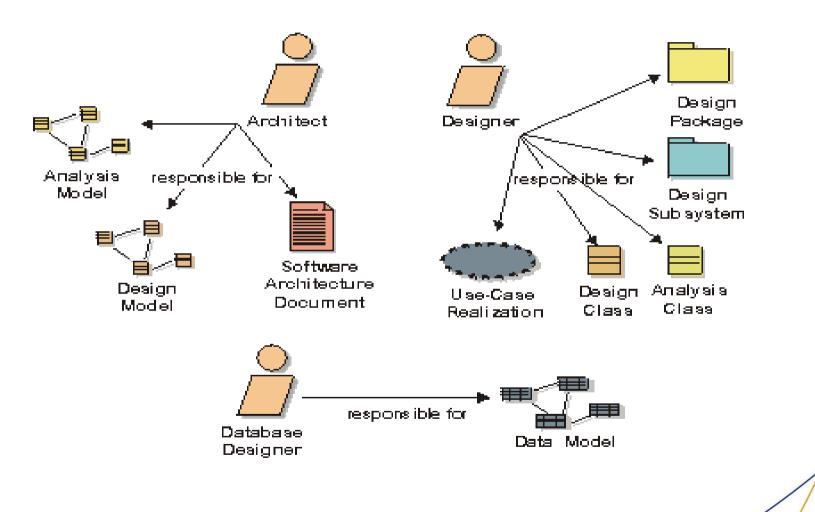
# Análise e Projeto





# Análise e Projeto





# Implementação

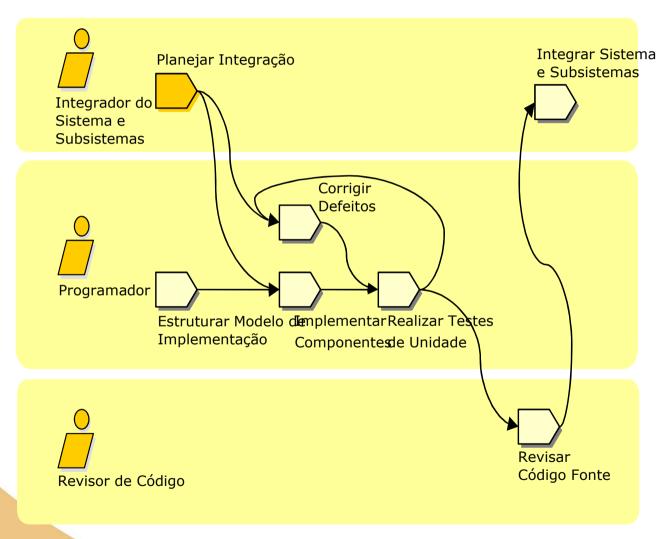


### Objetivos:

- implementar os componentes necessários
- testar os componentes implementados como unidades
- integrar os componentes implementados em um sistema executável

# Implementação





- Modelo de Implementação
- Componentes implementados
- Subsistemas implementados

### **Testes**

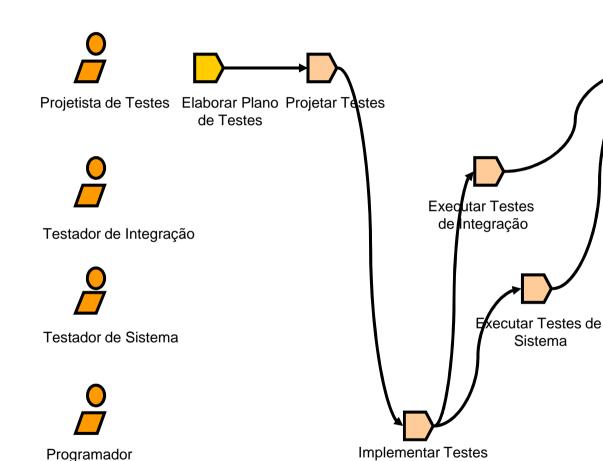


### Objetivos:

- verificar a interação e integração dos componentes
- verificar se todos os requisitos foram corretamente implementados
- identificar defeitos e assegurar as correções de acordo com as prioridades de entrega de cada componente

### **Testes**





■ Plano de Teste

Avaliar Testes

- Modelo de Teste
- Casos de Teste
- Procedimentos de Teste
- Lista de defeitos
- Avaliação dos Testes

# Implantação



- Objetivo:
  - entregar o produto aos usuários finais
- Muito dependente do contexto do negócio e do projeto => precisa ser configurado

### **Implantação**



- Possíveis atividades:
  - produzir o (que falta do) software
    - scripts de instalação, documentação para o usuário, programas para conversão de dados, etc.
  - embalar e distribuir o software
  - instalar o software
  - realizar migração
    - troca de sistema antigo pelo novo, conversão de dados
  - treinamento do usuário
  - aceitação formal pelo cliente
  - planejamento e condução de beta testes

# Gerência de Projeto



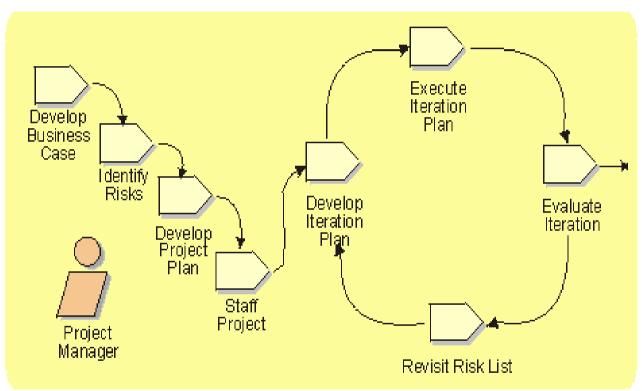
#### Objetivos:

- prover um framework para gerenciar projetos e riscos
- prover orientações para o planejamento de atividades,
   definição da equipe, execução e monitoração de projetos

#### Planejamento e monitoração das iterações!

# Gerência de Projeto





- Plano de Negócios
- Plano do Desenv. do Software:
  - plano de métricas
  - lista de riscos
  - plano do projeto
  - plano da iteração

### Gerência de Configuração e Mudanças

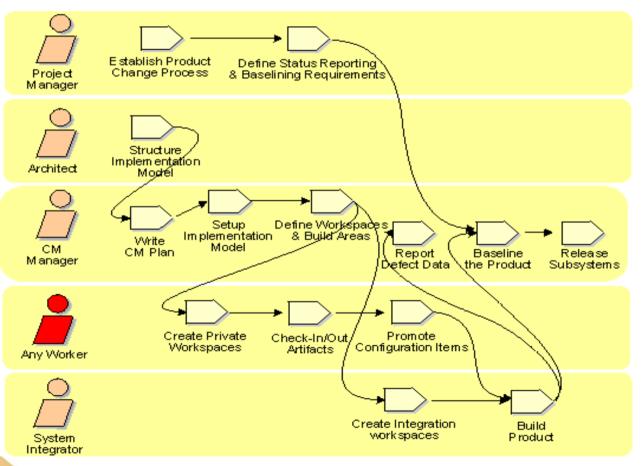


#### Objetivos:

- identificar, definir e manipular itens de software
- controlar modificações e versões destes itens
- reportar e armazenar a situação dos itens e as solicitações de mudanças
- garantir a completude, consistência e corretude dos itens
- controlar o armazenamento, manipulação e entrega destes itens



### Gerência de Configuração e Mudanças Senac



 Plano de gerência de configuração

Fonte: Rational

#### Gerência de Ambiente



#### Objetivo:

prover o processo e as ferramentas necessárias ao desenvolvimento

#### Possíveis Atividades:

- configurar o RUP
- desenvolver guidelines
- selecionar e adquirir ferramentas
- adaptar ou desenvolver ferramentas
- suportar o ambiente de desenvolvimento (backups, administração de contas, etc.)
- treinamento
- implantação do RUP na organização

#### Gerência de Ambientes



- Processo para um projeto x Processo para a organização
- Processo para um projeto
  - considerar tamanho, reuso, tipo do ciclo (inicial x de evolução)
- Processo para a organização
  - considerar cultura, pessoas, tecnologias e aplicações chave, ...

# Obrigado!





### **Edilberto Silva**

www.edilms.eti.br