

# Modelagem de Software



---

Prof. Dr. Ronaldo C. de Oliveira

[ronaldo.co@ufu.br](mailto:ronaldo.co@ufu.br)

[www.facom.ufu.br/~ronaldooliveira](http://www.facom.ufu.br/~ronaldooliveira)

FACOM - 2019



---

# Processo Unificado de Desenvolvimento de Software



# Processo Unificado

---

- O que é:
  - Um **processo** (de engenharia) **de software** é a definição de um conjunto completo de **atividades** necessárias para transformar os **requisitos** do cliente num conjunto consistente de **artefatos** que representam um **produto** de software e, mais tarde, transformar alterações nesses requisitos num novo conjunto consistente de artefatos [Jacobson 1999]
    - define **quem** faz **o quê**, **quando** e **como** para atingir um determinado objetivo
  - A presença de um processo **bem definido e bem gerenciado** é um aspecto determinante de diferenciação entre projetos produtivos e projetos mal-sucedidos



# Processo Unificado

---

- Características do Processo Unificado:
  - Sustentado em **UML**
  - **Iterativo e incremental**
  - **Conduzido por casos de uso**
    - A identificação de casos de uso e cenários típicos de utilização é a atividade que conduz todo o processo de desenvolvimento, desde a análise de requisitos até ao teste do sistema final
  - **Centrado numa arquitetura**
    - Promove a definição inicial de uma arquitetura de software robusta, que facilita a o desenvolvimento em paralelo, a reutilização e a manutenção
  - **É um processo configurável**
    - Pode ser ajustado e redimensionado conforme as necessidades da empresa



# Processo Unificado

---

- Guiado por Casos de uso:
  - Os casos de uso não servem apenas para definir os requisitos do sistema
  - Várias atividades do processo unificado são guiadas pelos casos de uso:
    - planejamento das iterações
    - criação e validação do modelo de projeto
    - planejamento da integração do sistema
    - definição dos casos de teste



# Processo Unificado

---

- Rational Unified Process – RUP é uma instanciação do processo Unificado de Desenvolvimento d Software
- RUP é um produto de mercado mantido hoje pela IBM Software.
- RUP pode ser facilmente adaptado para o seu projeto.



# RUP - Processo Unificado

---

- **Conceitos Principais:**

- **Fases**

- cada ciclo resulta numa nova geração do produto e divide-se em fases
- cada fase divide-se em iterações a definir em cada projecto concreto

- **Trabalhadores (*workers*)**

- são perfis a que correspondem competências para a realização de atividades

- **Atividades**

- são tarefas que podem ser entregues a trabalhadores individuais

- **Artefatos**

- são as entradas e saídas das atividades

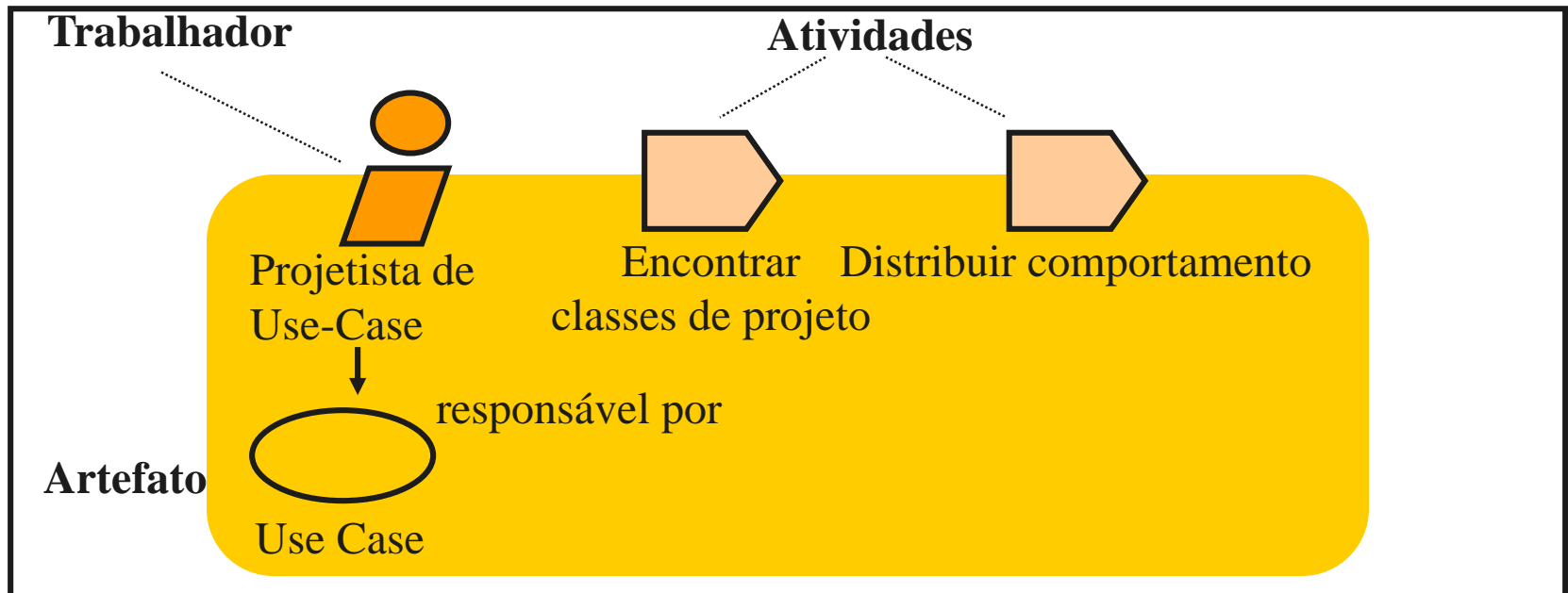
- ***Workflows***

- agrupam atividades relacionadas
- genéricos ou especializados por fases

- **Modelos**

- agrupam artefatos desenvolvidos num *workflow*

# Trabalhadores, Atividades, Artefatos



Em termos de UML:

- Trabalhador:objeto ativo
- Atividade: operação sobre um trabalhador
- Artefato: parâmetro de uma atividade





# Definição de Trabalhador

---

- Um trabalhador (Worker) define o **comportamento e responsabilidades** de um indivíduo, ou um conjunto de indivíduos, trabalhando juntos como uma equipe.
- Comportamento: um conjunto de atividades coesas.
- Responsabilidade: usualmente definida relativa para certos artefatos.
- Em termos de OO: é um objeto ativo.

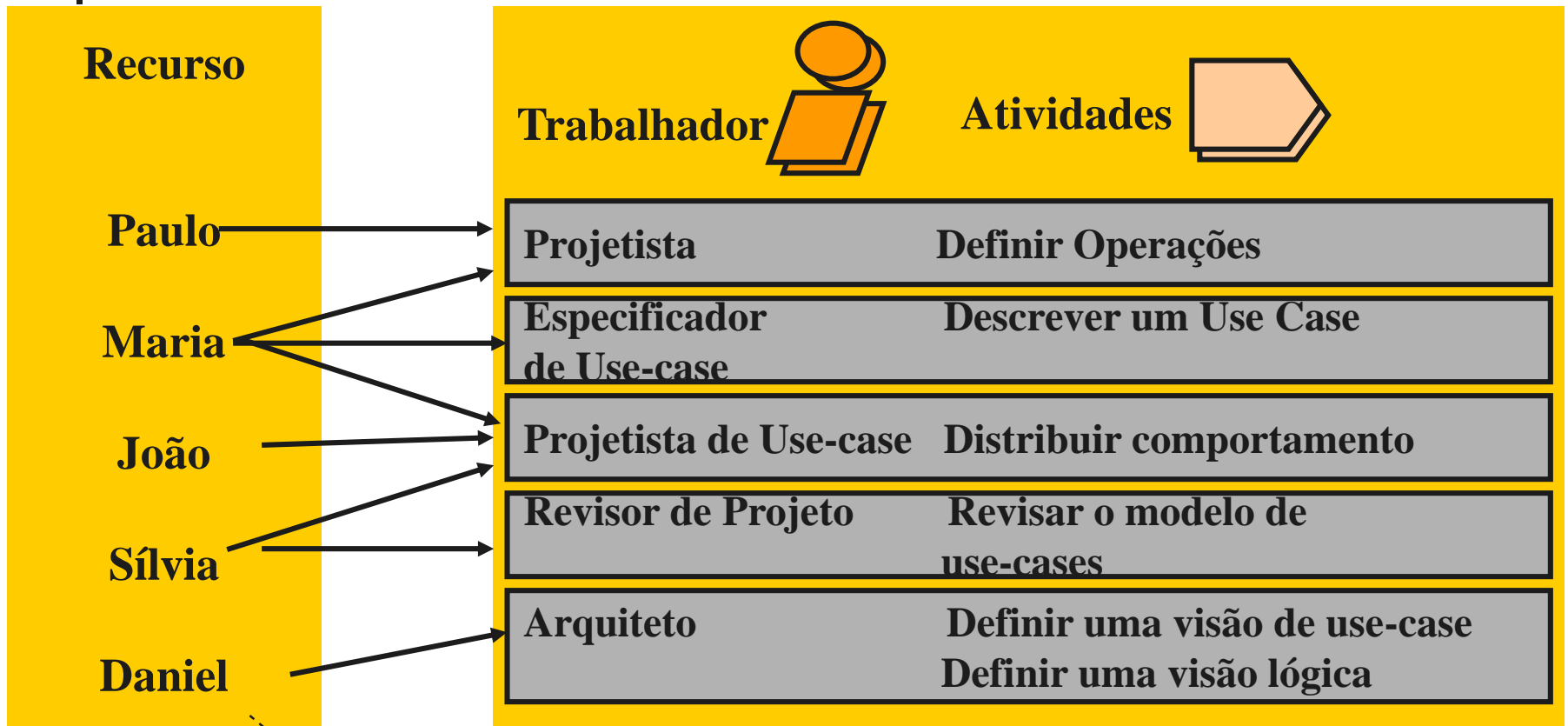


# Exemplos de Trabalhadores

---

- Analista de sistema;
- Gerente de projeto;
- Projetista de Use-Case;
- Projetista de teste;
- Instrutor de curso.

# Trabalhadores e Pessoas



Cada indivíduo no projeto é associado a um ou vários trabalhadores



# Definição de Atividade

---

- Um trabalho que um trabalhador é convidado a executar;
- Granularidade: de poucas horas a poucos dias
- Unidade de planejamento;
- Repetido, quando necessário, em cada iteração;
- Em termos de OO: uma operação sobre um trabalhador.



# Exemplos de Atividades

---

- Planejar uma iteração;
- Encontrar use cases e atores;
- Revisar o projeto;
- Executar o teste de desempenho.



# Passos de uma Atividade

---

- Atividades são decompostas em passos
- Tipos de passos
  - Passos de pensamento;
  - Passos de realização;
  - Passos de revisão.



# Exemplo de Atividade

---

- Atividade: Encontrar Use Case e Atores
  - Passo 1: Encontrar Atores;
  - Passo 2: Encontrar Use Cases;
  - Passo 3: Descrever como Atores e Use Cases interagem;
  - Passo 4: Empacotar Use Cases e Atores;
  - Passo 5: Apresentar o Modelo de Use Case em Diagramas de Use Case;
  - Passo 6: Desenvolver um detalhamento do modelo de Use Cases;
  - Passo 7: Avaliar seus resultados.



# Definição de Artefatos

---

- Alguma informação que é produzida, modificada, ou apenas utilizada por um processo.
- Define uma área de responsabilidade.
- Provavelmente é passível de controle de configuração.
- Tipos de artefatos:
  - Modelos;
  - Documentos;
  - Planos.
- Artefatos podem conter outros artefatos.





# Definição de Artefatos

---

- Exemplos de Artefatos:
  - Modelo de projeto;
  - Classe;
  - Use Case;
  - Caso de teste;
  - Plano de desenvolvimento de software;
  - *Release*;
  - Documento de avaliação de *status*;
  - Lista de risco.



# Artefatos de Modelo

---

- **Modelo do negócio:** Estabelece uma abstração da empresa;
- **Modelo do domínio:** Estabelece o contexto do sistema;
- **Modelo de use case:** Estabelece os requisitos funcionais do sistema;
- **Modelo de análise** (opcional): Estabelece o projeto de uma idéia;
- **Modelo de projeto:** Estabelece o vocabulário do problema e de sua solução;
- **Modelo de processo** (opcional): Estabelece os mecanismos de concorrência e sincronização do sistema;
- **Modelo de implantação:** Estabelece a topologia do Hardware que o sistema é executado;
- **Modelo de implementação:** Estabelece as partes utilizadas para montar e liberar o sistema físico;
- **Modelo de teste:** Estabelece os caminhos de verificação e validação do sistema.

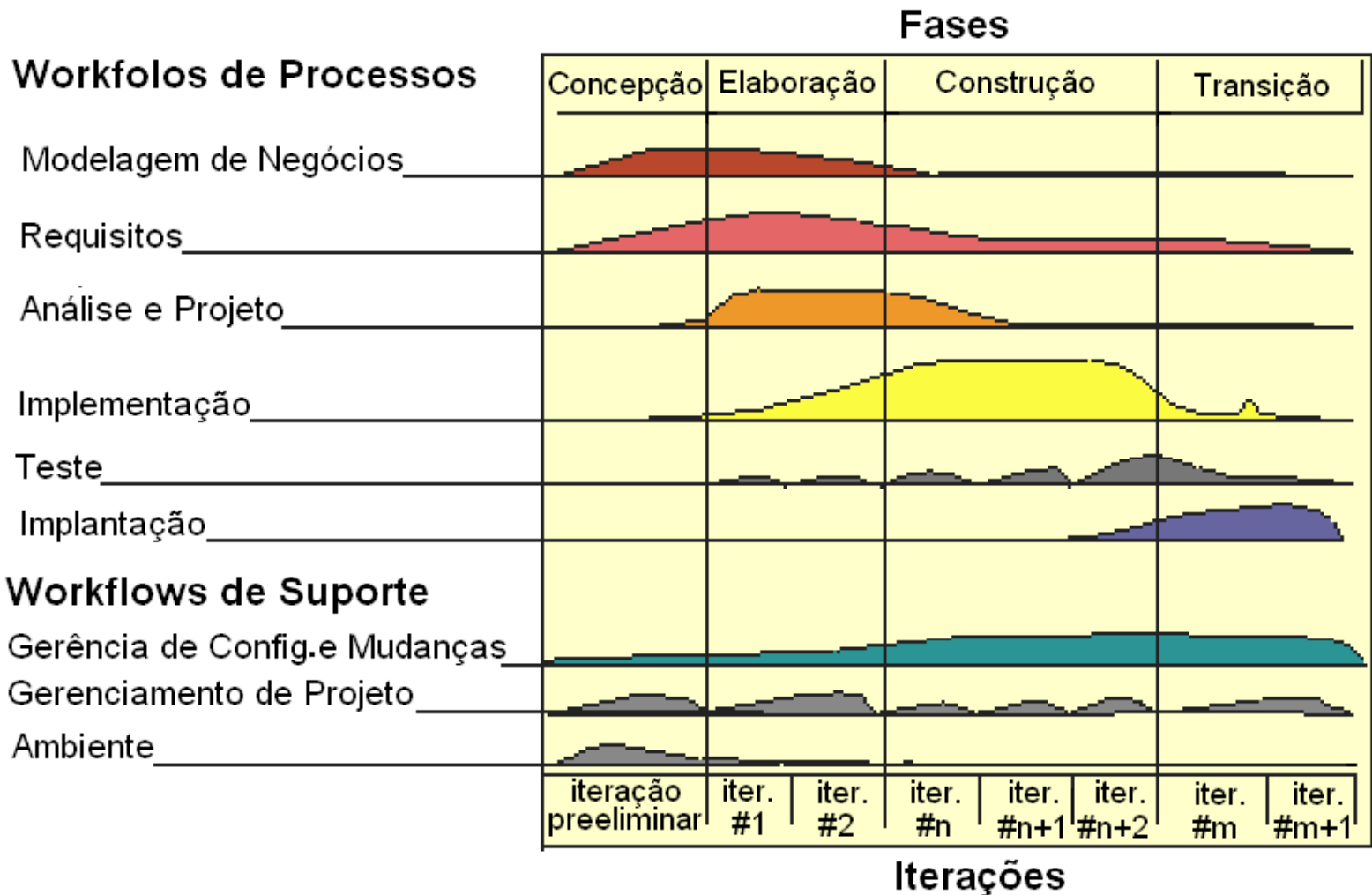
# Conjuntos: Agrupamento de Artefatos

---

- **Conjunto de gerenciamento**
  - Casos de negócio, plano de desenvolvimento de software, documento de avaliação de *status*.
- **Conjunto de requisitos**
  - documento de visão, modelo de caso de uso, especificações suplementares.
- **Conjunto de projeto**
  - modelo de projeto, modelo de teste, descrição da arquitetura.
- **Conjunto de implementação**
  - Código fonte, executáveis,...
- **Conjunto de entrega**
  - *Baseline* do produto, manual do usuário, manual de instalação,...

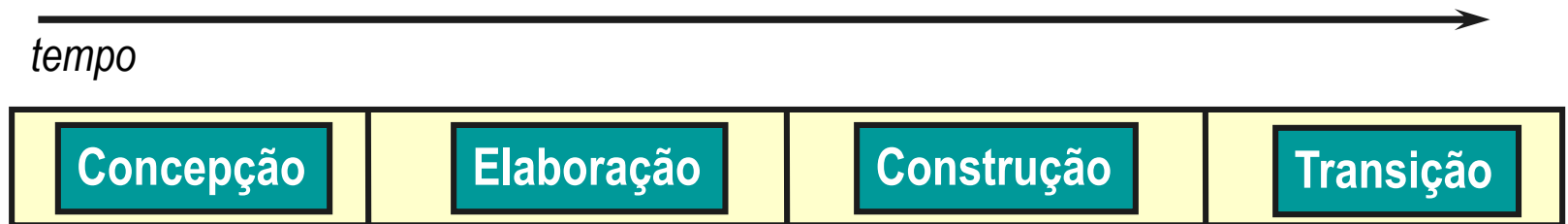
# RUP

## Fases e Workflows



# RUP

## Fases de Implementação



**Concepção**

→ Definir o escopo do projeto;

**Elaboração**

→ Planejar o projeto, especificar as características a *baseline* (linha base) da arquitetura;

**Construção**

→ Construir o produto;

**Transição**

→ Disponibilizar o produto para a comunidade de usuários.

# RUP

## Determinação das Fases

---

### ■ **Concepção**

- Delimitação dos limites do projeto e o a definição do caso de negócio, que inclui critérios de sucesso, a avaliação dos riscos, a estimativa de recursos e um plano para a fase, mostrando os principais marcos do progresso do projeto.
- Identificação dos atores e casos de uso e a descrição dos casos mais significativos.
- Construção de um protótipo executável, servindo como teste para a concepção.
- No final da Concepção, examinar os objetivos do ciclo de vida do projeto e decidir se deve prosseguir com o desenvolvimento.



# Concepção - Atividades

---

- Requisitos: capturar os requisitos mais críticos (na forma de casos de uso) e definir o escopo do sistema.
- Análise: analisar os requisitos e montar uma proposta para o modelo de classes e objetos, com foco nas classes de negócio, mais o glossário.
- Desenho: preparar o Modelo de Design ou storyboard, apresentando um rascunho preliminar da arquitetura do sistema: identificar os primeiros componentes, interfaces e subsistemas, assim como o Modelo de Implantação.
- Implementação: pode ser necessário criar um protótipo descartável para demonstrar o caminho escolhido.
- Testes: criar primeiros esboços de teste com base nas informações já adquiridas.

# RUP

## Determinação das Fases

---

### ■ **Elaboração**

- Análise do domínio do problema.
- Definição de uma arquitectura estável e robusta para todo o sistema, tendo em consideração os seus requisitos.
- Desenvolvimento do plano de projeto.
- Eliminação dos elementos de alto risco do projeto.
- Implementar um sistema que demonstre as escolhas de arquitetura e execute os casos de uso significativos.
- No final da Elaboração, deve-se examinar o escopo e os objetivos detalhados do sistema, a escolha da arquitetura e a solução para os principais riscos, além de decidir se o projeto irá continuar.





# Elaboração - Atividades

---

- Requisitos: encontrar, priorizar, detalhar e estruturar os Casos de Uso, capturando aproximadamente 80% dos requisitos.
- Análise: detalhar as classes de negócio, fazer o particionamento em pacotes, atualizar o glossário e refinar os Casos de Uso.
- Design: fazer o design dos Casos de Uso, classes e subsistemas para estabelecer uma estrutura básica do sistema. Pacotes de análise e subsistemas de design, são importantes. São considerados: sistema operacional, linguagem, banco de dados, distribuição de objetos, etc..
- Implementação: implementar e testar os componentes arquiteturalmente significantes. Eventualmente criar protótipos para testar alguma nova tecnologia.
- Testes: planejar e especificar os testes, definindo casos de teste e rotinas de teste.

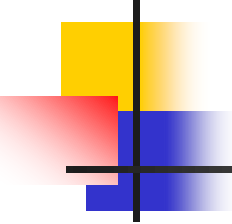
# RUP

## Determinação das Fases

---

### ■ **Construção**

- Desenvolvimento iterativo e incremental do produto completo para ser entregue aos usuários como uma primeira versão.
- Descrição dos requisitos restantes e de critérios de aceitação, dando corpo ao projeto e concluindo a implementação e teste do sistema.
- No final da Construção, deve-se decidir se o software, ambiente e usuários estão todos prontos para se tornarem operacionais.



# Construção - Atividades

---

- Requisitos: capturar os requisitos remanescentes, refinando Casos de Uso e cenários
- Análise: capturar algum detalhe que passou despercebido nas classes pertinentes ao negócio.
- Design: refinar os casos de uso e cenários remanescentes com base na tecnologia utilizada.
- Implementação: codificar e integrar componentes, priorizando os casos de uso mais importantes.
- Testes: testar funcionalidades e performance do sistema. Se necessário testar novos casos e rotinas de teste.

# RUP

## Determinação das Fases

---

### ■ Transição

- Colocar o software disponível para os usuários
- Desenvolvimento adicional para ajuste do sistema às alterações de requisitos solicitadas pelos usuários, ajustando e corrigindo problemas ou concluindo características propostas a *posteriori*.
- No final da Transição, deve-se decidir se foram alcançados os objetivos do ciclo de vida do projeto e determinar se será necessário iniciar outro ciclo de desenvolvimento.
- Lições aprendidas no projeto devem ser discutidas e assimiladas para serem aplicadas ao processo de desenvolvimento de novos projetos.



# Transição - Atividades

---

- Requisitos: eventual correção da documentação devido a bugs encontrados no sistema.
- Análise: eventual correção do modelo de análise devido a bugs encontrados no sistema.
- Design: eventual correção do modelo de design devido a bugs encontrados no sistema.
- Implementação: eventual correção do código devido a bugs encontrados no sistema.
- Testes: eventual correção do modelo de teste devido a bugs encontrados no sistema.

# RUP

## Definição dos Workflows

---

- Seqüências de atividades que produzem um resultado observável;
- Diferentes *workflows*
  - *Workflows* núcleo
  - *Workflows* de iteração
- Em termos de OO: diagramas de interação ou diagramas de atividades

# RUP

## Definição dos Workflows

---

- **Modelagem do negócio:** Descreve a estrutura e a dinâmica da organização;
- **Requisitos:** Descreve o método de identificação de requisitos baseado em casos de uso;
- **Análise e projeto:** Identificação de uma arquitectura, análise dos use cases, desenho das classes, etc;
- **Implementação:** Definir a estrutura do sistema, a implementar, teste de unidade e a integração dos componentes
- **Testes:** Descreve casos de teste, procedimentos e métricas
- **Implantação:** Promove a configuração do sistema, a produção de manuais de utilização, etc.

# RUP

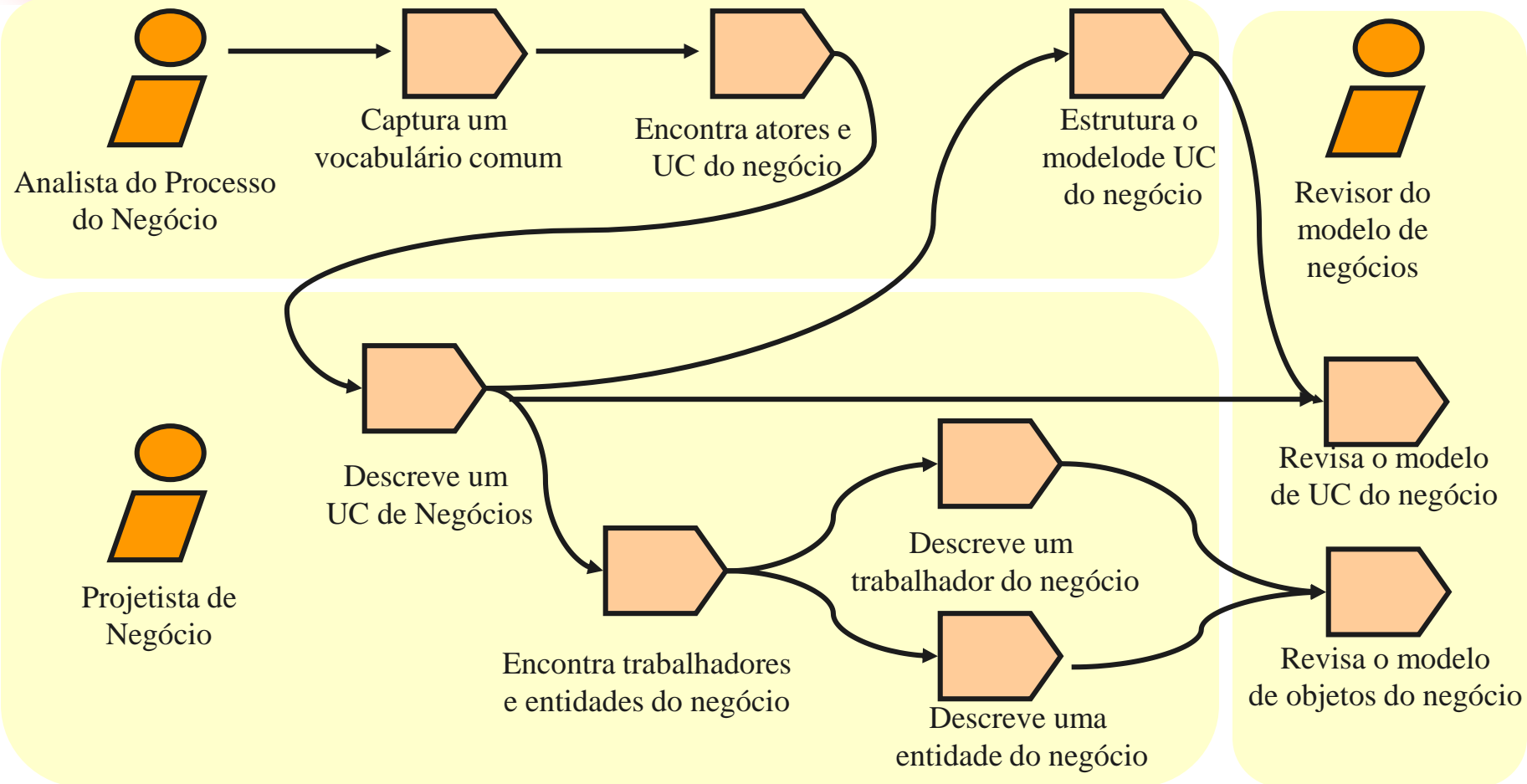
## Definição dos Workflows

---

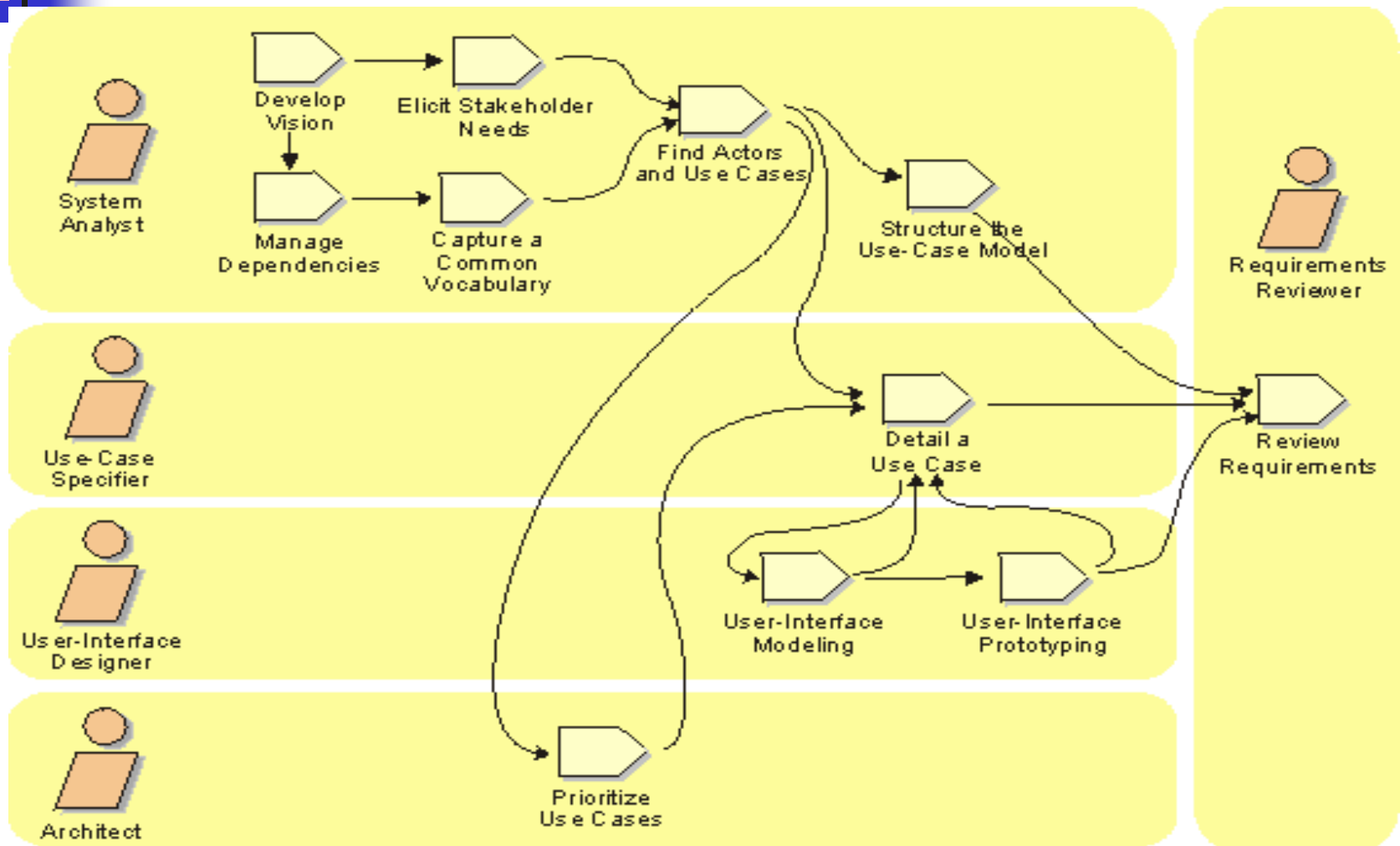
- **Gestão de projetos:** Especifica um conjunto de princípios a serem aplicados na gestão do projeto a nível de alocação de recursos, planeamento, identificação e controle de riscos, etc.
- **Gestão de configuração e versão:** Controla a mudança e mantém a integridade dos artefatos do projeto.
- **Definição do ambiente:** Cobre a infra-estrutura necessária para desenvolver um sistema (seleção de ferramentas, e definição dos regras para modelar o negócio, interface, testes)



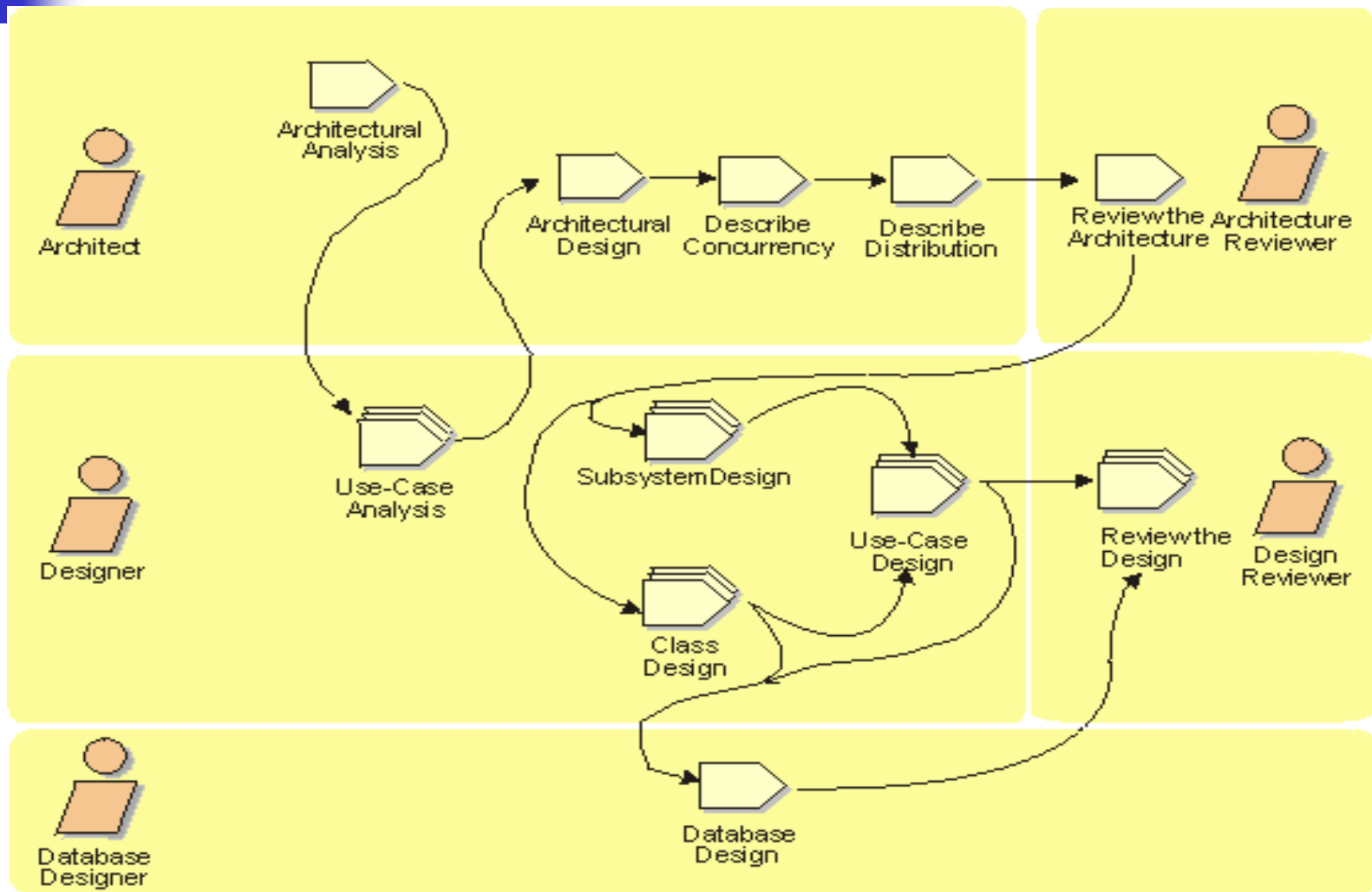
# Workflow de Modelagem de Negócio



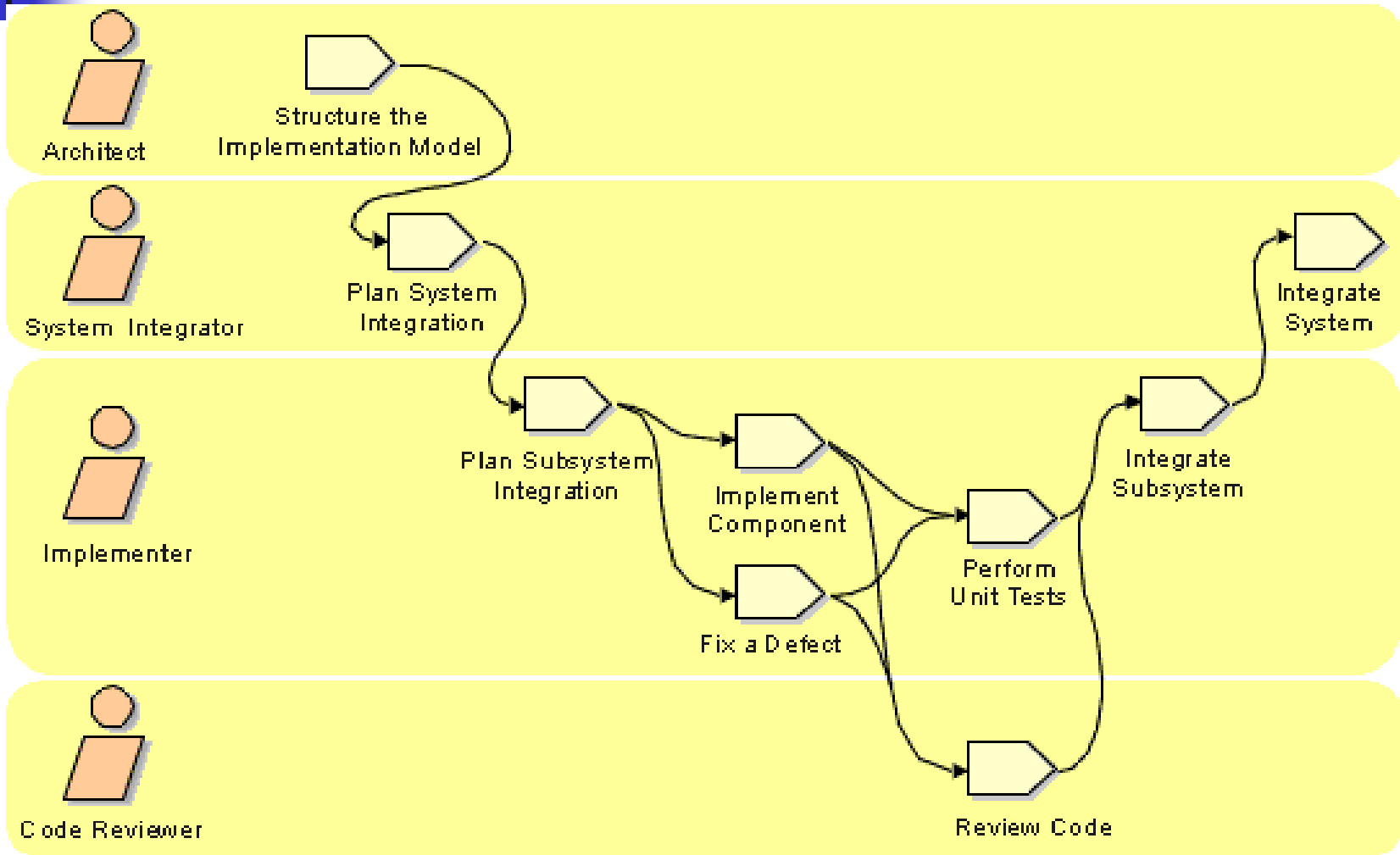
# Workflow de Levantamento de Requisitos



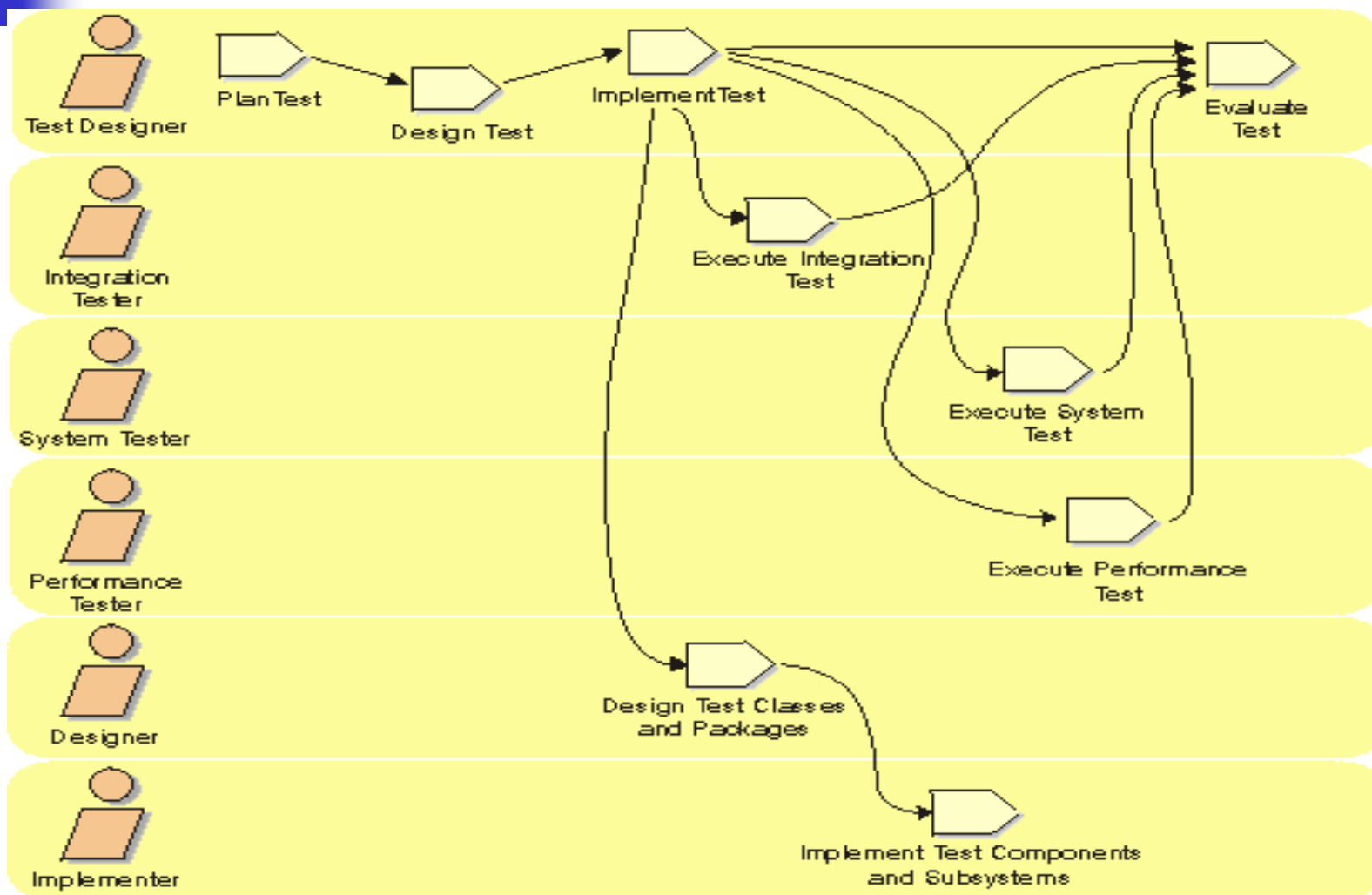
# Workflow de Análise e Projeto



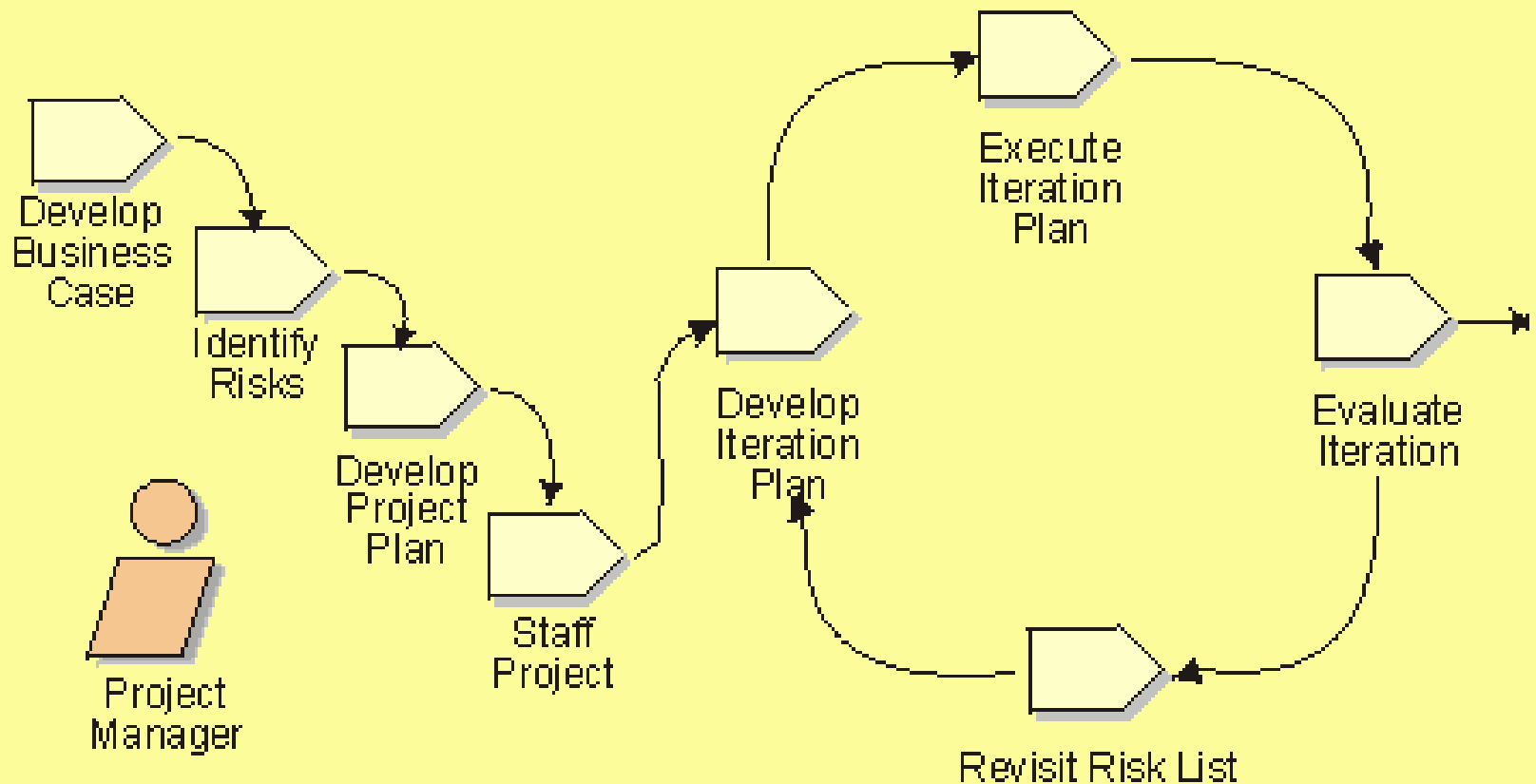
# Workflow de Implementação



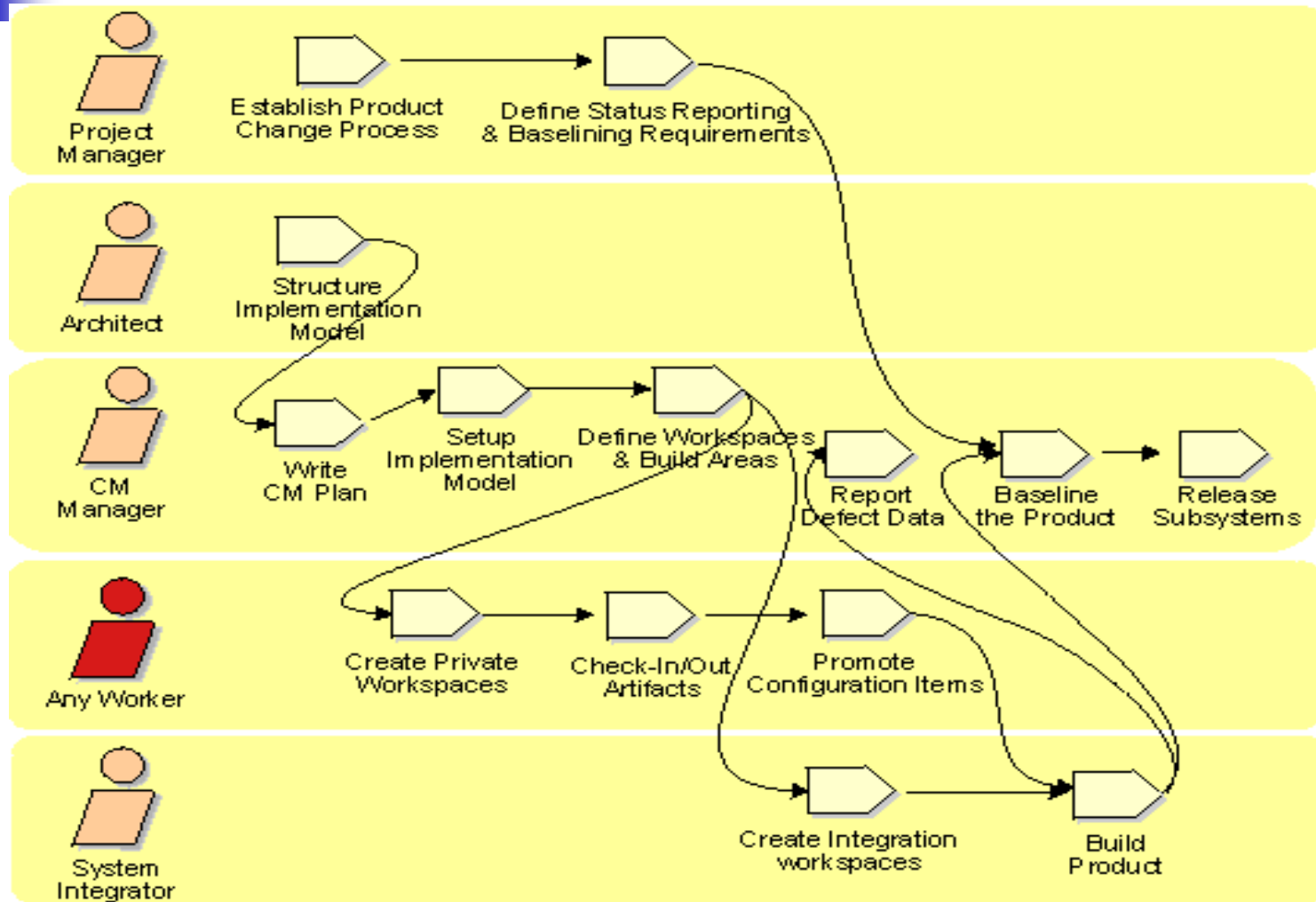
# Workflow de Teste



# Workflow de Gestão de Projetos



# Workflow Gestão de Configuração e Versão



# RUP

## Processo Iterativo

---

- Cada fase pode ser decomposta em iterações
- Iterações são definidas e planeadas em cada projeto concreto
- Cada iteração resulta num incremento ao produto
  - Tipicamente é analisado e implementado um grupo de casos de uso ou de variantes destes casos
- Cada iteração passa, em cascata, pelos *workflows* técnicos
  - Importância relativa dos *workflows* varia com as fases



# RUP

## Processo Iterativo

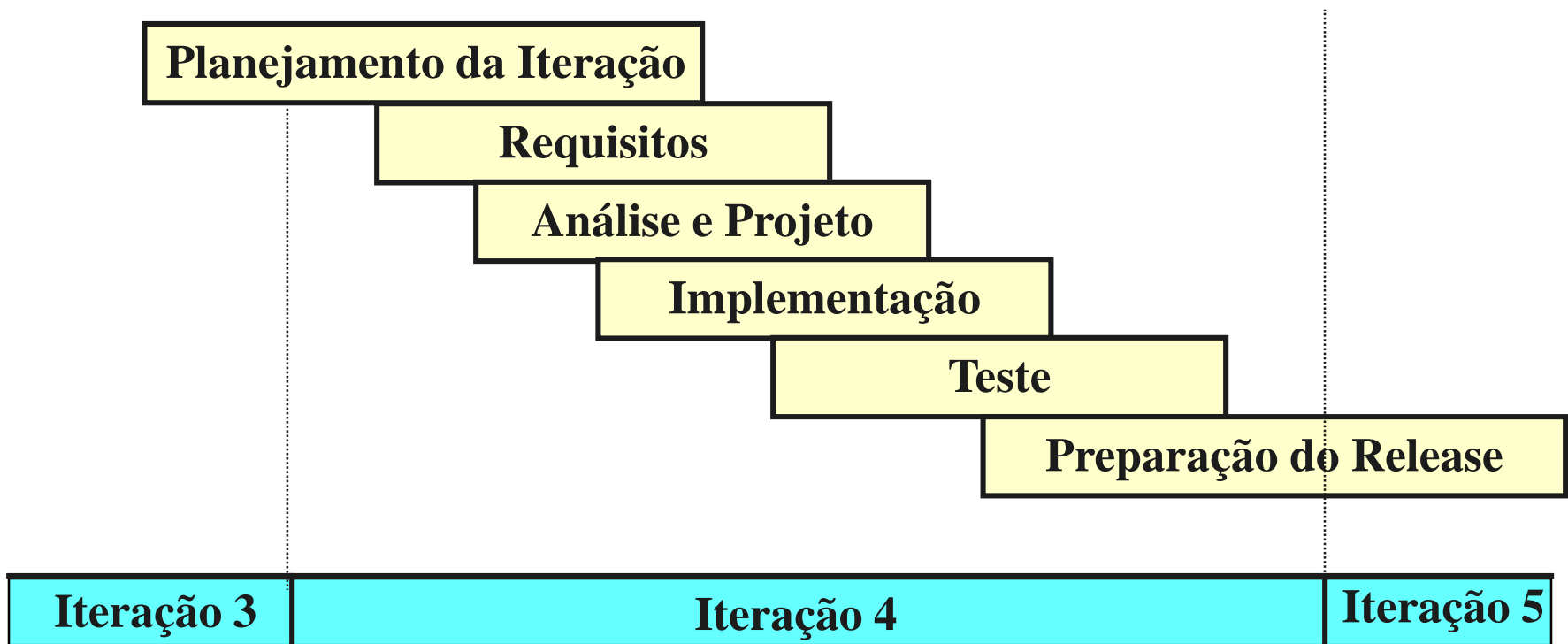


Uma iteração é uma seqüência distinta de atividades com um plano de linha base (*baseline*) e critério de avaliação resultando em um *release* (interno ou externo) e controlado por um marco (*milestone*).

# RUP

## Processo Iterativo

**Uma Interação como uma Mini Cascata:**



# RUP

## Processo Iterativo

---

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

# RUP

## Processo Iterativo

---

- **Iteração: Duração e Número**
  - Duração dirigida por
    - + tamanho da organização
    - + tamanho do projeto
    - familiaridade com o projeto, maturidade
    - simplicidade técnica
  - 6 mais ou menos 3
    - Concepção: 0..1
    - Elaboração: 1..3
    - Construção: 1..3
    - Transição: 1..2

# RUP

## Processo Iterativo

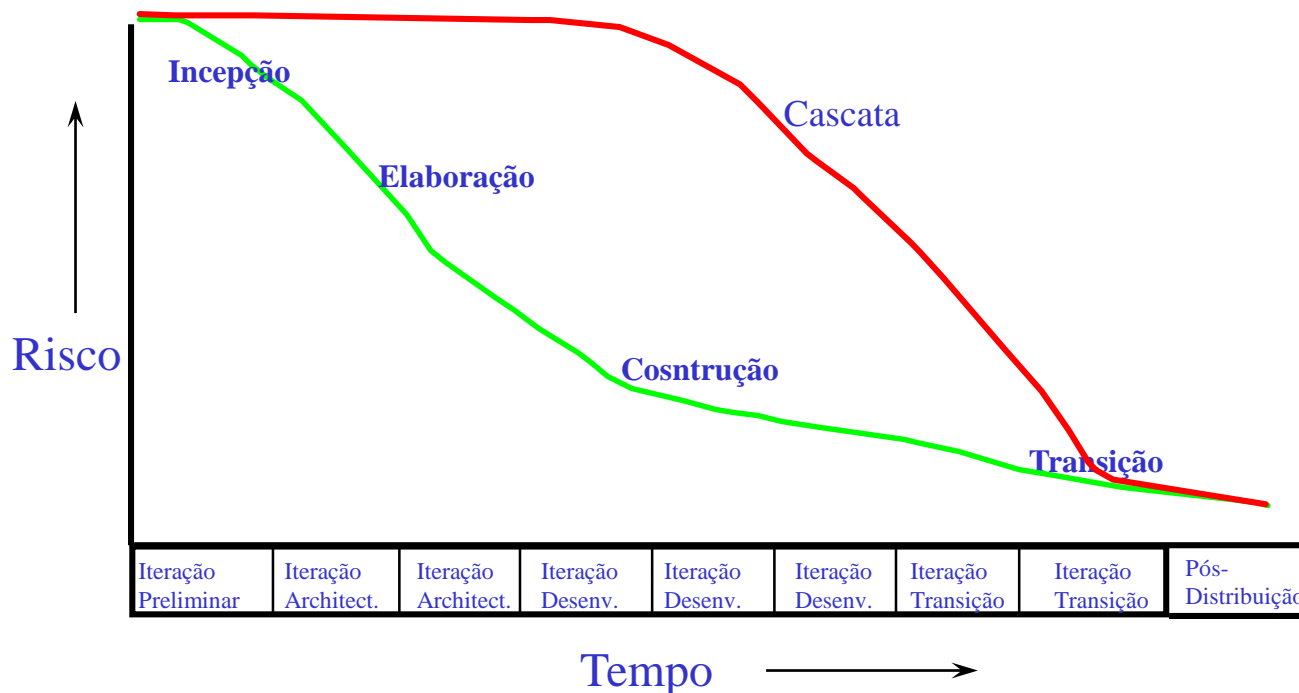
### ■ Redução dos Riscos do Projeto



# RUP

## Processo Iterativo

### Perfil de Risco de um Desenvolvimento Iterativo





# Vantagens do RUP

---

- Vantagens:
  - O RUP tem como base os princípios de engenharia de software refletidos na sua abordagem de desenvolvimento iterativa, incremental, orientada a requisitos e baseada em arquitetura
  - Sério investimento em ferramenta de suporte



# Limitações do RUP

---

- Limitações:
  - Nas áreas de manutenção, gestão de métricas, gestão de pessoal, gestão de reutilização e testes
  - A ferramenta é basicamente um sistema de hipertexto. Por exemplo, não se encontra integrado a outras ferramentas.