

# MELHORES PRÁTICAS EM DESENVOLVIMENTO DE SISTEMAS

# Roteiro

- Melhores Práticas
- Modelos de Processos de Desenvolvimento de Sistemas:
  - Processo Unificado
  - RUP – Rational Unified Process
  - Praxis

# Boa e Má Notícia

*"28% dos projetos de software foram bem-sucedidos."*

*Standish Group, CHAOS Report, 2001*

*"72% dos projetos de software falharam."*

# Problemas em desenvolvimento de sistemas

- Necessidades do usuário ou do negócio não satisfeitas.
- Requisitos que não se estabilizam
- Módulos que não se integram
- Difícil de manter
- Descoberta tardia de falhas
- Qualidade e experiência do usuário final ruim
- Esforço descoordenado da equipe
- Dificuldade em coordenar recursos terceirizados
- Questões de build-and-release

# Principais Causas

- Requisitos insuficientes
- Comunicação Ambigua
- Arquitetura Rústica
- Complexidade demasiada
- Inconsistências não detectadas
- Testes pobres
- Avaliação subjetiva
- Desenvolvimento em cascata
- Mudanças descontroladas
- Automação insuficiente

# Melhores Práticas em desenvolvimento de sistemas

**Desenvolver Iterativamente**

**Controlar  
Requisitos**

**Usar  
Arquitetura  
de  
Componentes**

**Modelar  
Visualmente**

**Verificar  
qualidade**

**Controlar Mudanças**

# Desenvolver Iterativamente

- Adote um processo flexível
- Gerencie requisitos
- Verifique continuamente a qualidade



Aplicativos de  
Negócio

Adaptam-se rapidamente para mudanças do negócio

Criam vantagem estratégica

São confiáveis, escaláveis e gerenciáveis

Desenvolver iterativamente para...

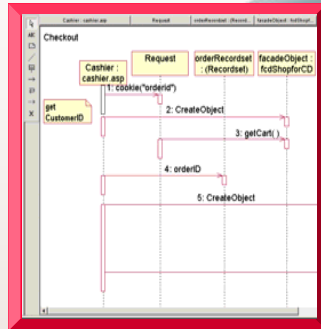
Reduzir risco de projetos e aumentar previsibilidade

Priorizar e definir escopo de projetos com mais eficiência

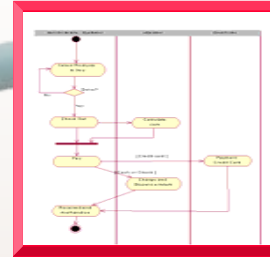
Descobrir problemas de Projeto mais cedo para aumentar a qualidade e os resultados

# Focar em Arquitetura

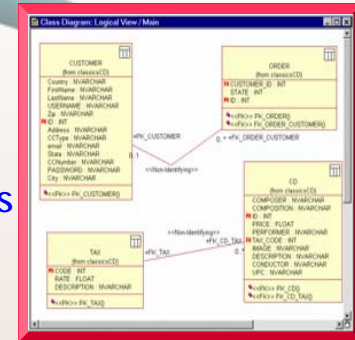
- Modele visualmente
- Use arquiteturas orientadas a componentes e serviços



Modelagem de Negócios



Modelagem de Aplicativos



Modelagem de Dados

Aplicativos de  
Negócio

Adaptam-se  
rapidamente  
para mudanças do  
negócio

Criam  
vantagem  
estratégica

São confiáveis,  
escaláveis e  
gerenciáveis

Focar em  
arquitetura para...

Projetar  
para mudança

Reduzir  
complexidade;  
trabalhar no nível  
correto de abstração

Garantir integridade  
na arquitetura e  
qualidade



# Verifique a Qualidade Continuamente

- Construa qualidade desde o início
- Teste automaticamente
- Una os times de desenvolvimento e teste



Aplicativos de  
Negócio

Adaptam-se  
rapidamente  
para mudanças do  
negócio

Criam  
vantagem  
estratégica

São confiáveis,  
escaláveis e  
gerenciáveis

Verifique  
qualidade  
para ...

**Garantir  
atendimento às  
necessidades do  
negócio**

**Validar as  
funcionalidades  
assim que ficam  
prontas**

**Verificar se a  
aplicação se  
comporta  
corretamente**

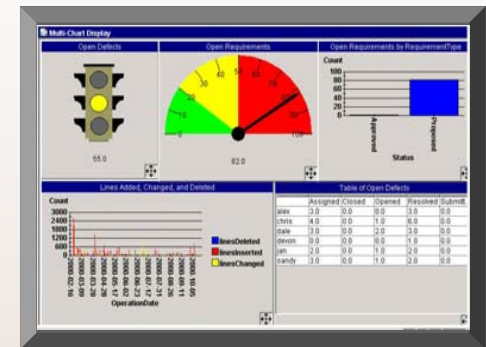
# Gerenciar mudanças e artefatos

- Gerencie mudanças
- Gerencie Artefatos
- Meça progresso e qualidade

## Gerência de Configuração



## Painel de Métricas



## Aplicativos de Negócio

Adaptam-se rapidamente para mudanças do negócio

Criam vantagem estratégica

São confiáveis, escaláveis e gerenciáveis

Habilitar equipes virtuais e desenvolvimento paralelo

Proteger artefatos estratégicos críticos

Implantar com confiança

Gerenciar mudanças e artefatos para...

# Processos de Desenvolvimento de Sistemas

## Modelos de Processos de Desenvolvimento de Sistemas:

- Processo Unificado
- Rational Unified Process – RUP
- Praxis

# Processo Unificado

- Características:
  - Agrega pontos positivos dos modelos anteriores:
    - Cascata: divisão em fases;
    - Prototipação: construção de protótipos executáveis em curtos espaços de tempo;
    - Espiral: desenvolvimento incremental e análise de riscos;

# Processo Unificado

## Fases:

Fase	Descrição
<b>Concepção</b>	Fase na qual se justifica a execução de um projeto de desenvolvimento de software, do ponto de vista do negócio do cliente.
<b>Elaboração</b>	Fase na qual o produto é detalhado o suficiente para permitir um planejamento acurado da fase de construção.
<b>Construção</b>	Fase na qual é produzida um versão completamente operacional do produto.
<b>Transição</b>	Fase na qual o produto é colocado à disposição de uma comunidade de usuários.

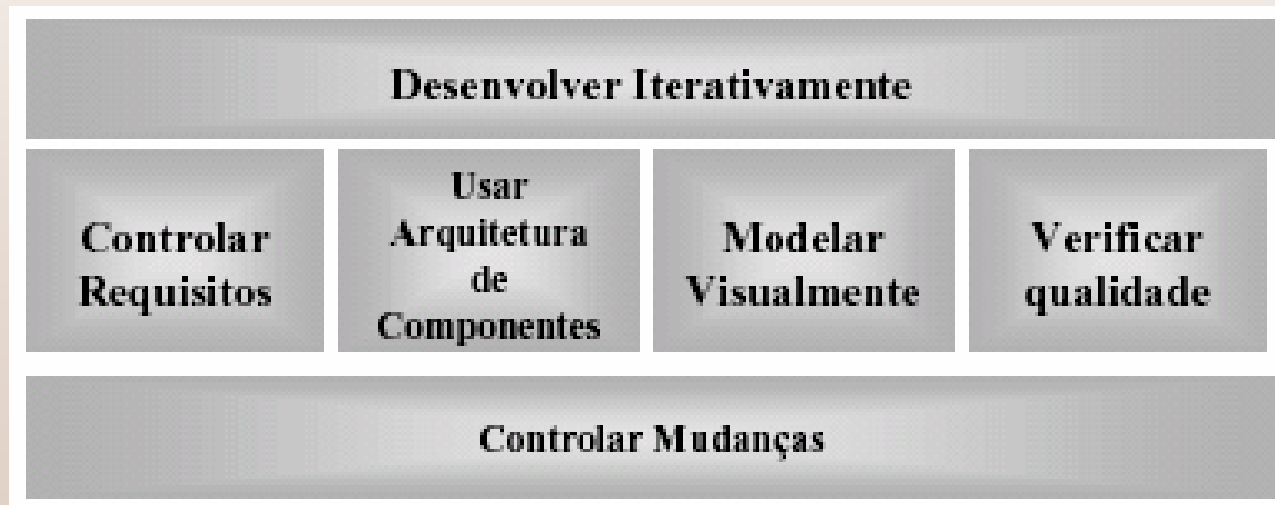
# Processo Unificado

## Fluxos:

Fluxo	Descrição
<b>Requisitos</b>	Fluxo que visa obter um conjunto de requisitos de um produto, acordado entre cliente e fornecedor.
<b>Análise</b>	Fluxo cujo objetivo é detalhar, estruturar e validar os requisitos, de forma que estes possam ser usados como base para o planejamento detalhado.
<b>Desenho</b>	Fluxo cujo objetivo é formular um modelo estrutural do produto, que sirva de base para a implementação.
<b>Implementação</b>	Fluxo cujo objetivo é realizar o desenho em termos de componentes de código.
<b>Testes</b>	Fluxo cujo objetivo é verificar os resultados da implementação.

# RUP - Rational Unified Process

Engloba as melhores práticas do mercado para desenvolvimento de software.



# RUP - Rational Unified Process

## Definição

- É um processo iterativo e incremental.
- É um processo controlado.
- Suporta as técnicas da orientação a objetos.
- Suporta o desenvolvimento baseado em componentes.
- É um processo configurável.
- Foi desenvolvido pela Rational Software Corporation.

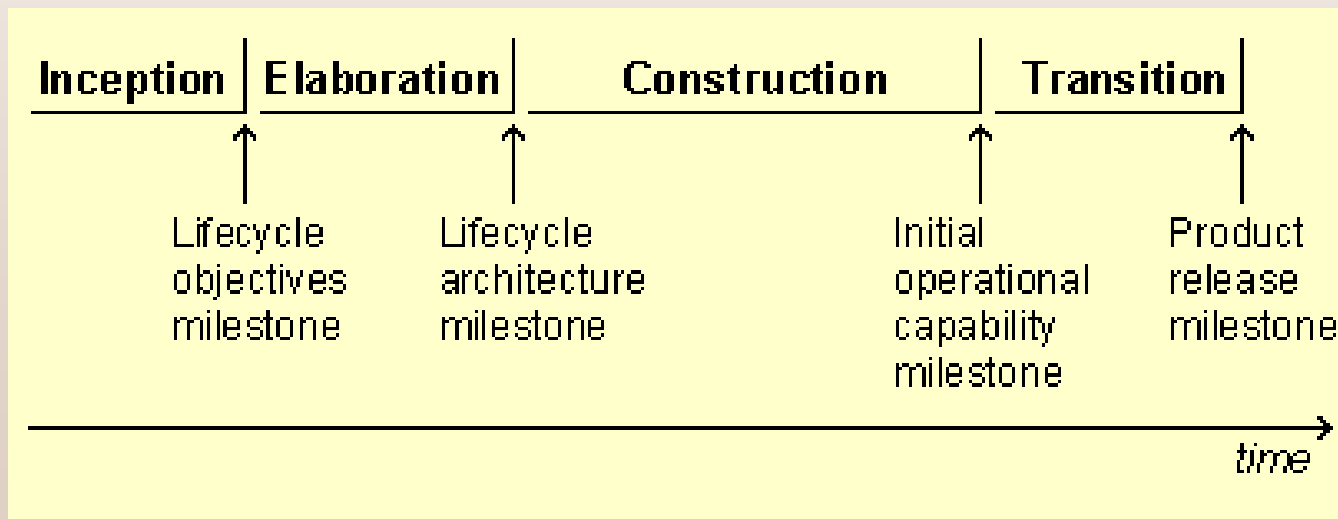


# Rational Unified Process



Características:

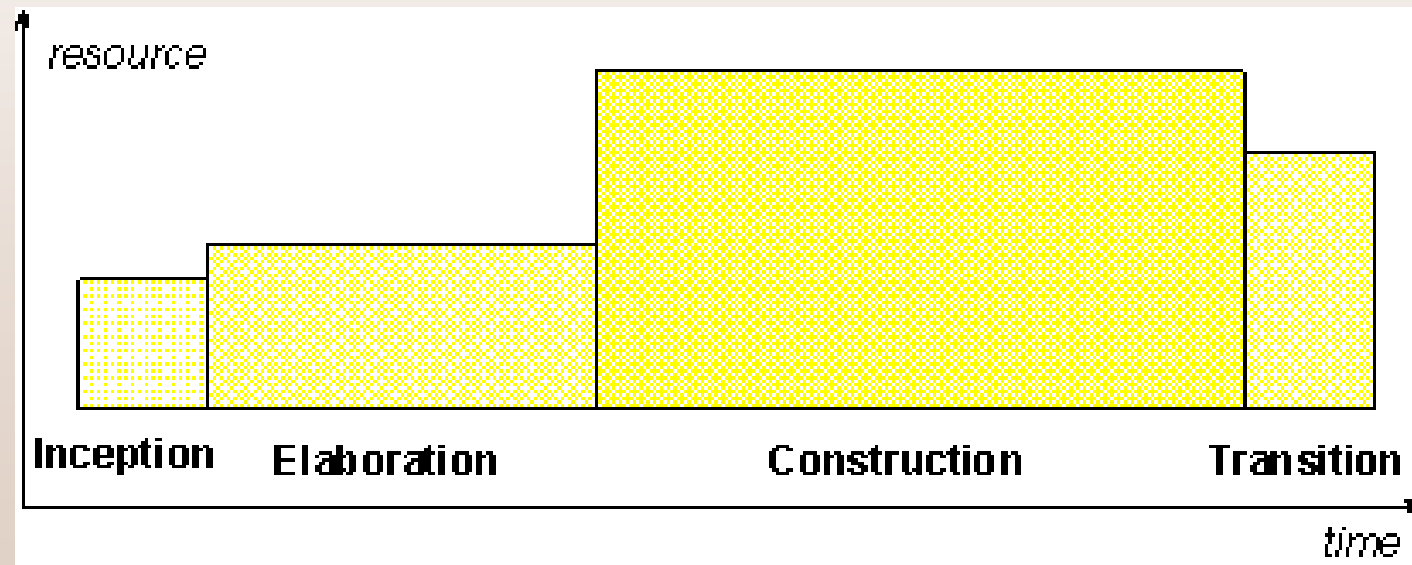
- O ciclo de desenvolvimento do software é dividido em fases onde o resultado de cada fase é a geração de um produto.



# Rational Unified Process



Distribuição do esforço ao longo do desenvolvimento



# Rational Unified Process



## Fase de Concepção

- Propósito:
  - Estabelecer qual é o negócio para um novo sistema ou para uma atualização importante de um sistema existente.
- Produtos necessários:
  - Os requisitos essenciais do projeto.
  - Uma avaliação inicial do risco - Identificar os elementos de maior risco do projeto.
  - Um modelo inicial de domínio (10% a 20% completo).
- Produtos opcionais:
  - Um protótipo conceitual.

# Rational Unified Process



## Fase de Elaboração

- Propósito
  - Analisar o domínio do problema.
  - Estabelecer uma base arquitetural apropriada.
  - Desenvolver um plano abrangente de como será completado o projeto.

# Rational Unified Process

## Fase de Elaboração

- Produtos:
  - Um modelo do comportamento do sistema (80% completo).
  - Uma arquitetura executável.
  - A visão de uma linha de produtos baseada no modelo de domínio.
  - Revisão da avaliação de riscos.

# Rational Unified Process



## Fase de Elaboração - Produtos:

- Um plano de desenvolvimento.
- Critérios de avaliação.
- Descrição das liberações de executáveis (*releases*).
- Versão preliminar do manual do usuário (opcional).
- Estratégia de testes.
- Plano de testes.

# Rational Unified Process



## Fase de Construção

- Propósito:
  - Desenvolver incrementalmente um produto de software completo que está pronto para a transição para a comunidade de usuários.
- Produtos:
  - Uma série de liberações de executáveis.
  - Resultados do controle de qualidade.
  - Documentação do sistema e do usuário.
  - Critério de avaliação para, pelo menos, a próxima iteração.

# Rational Unified Process

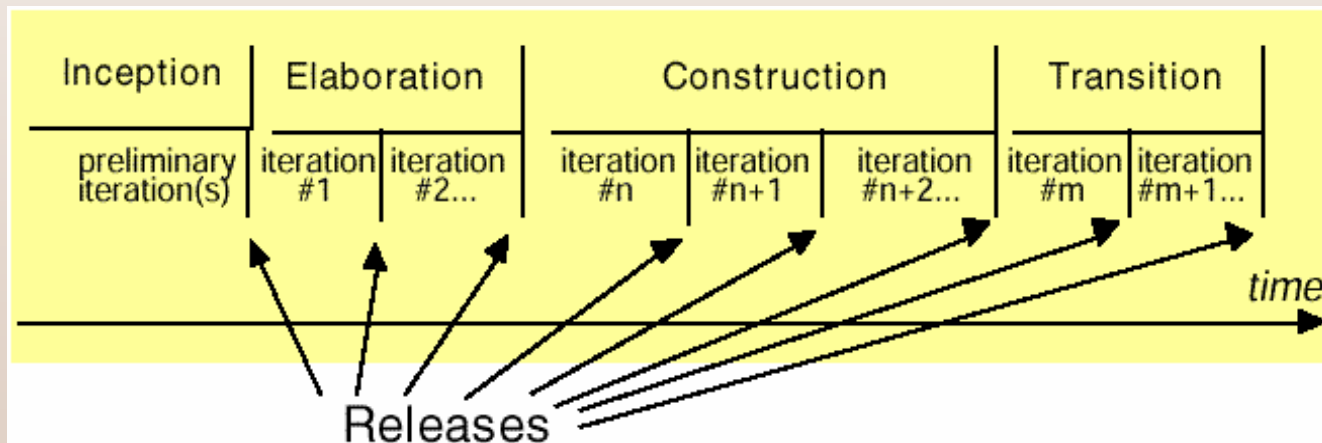
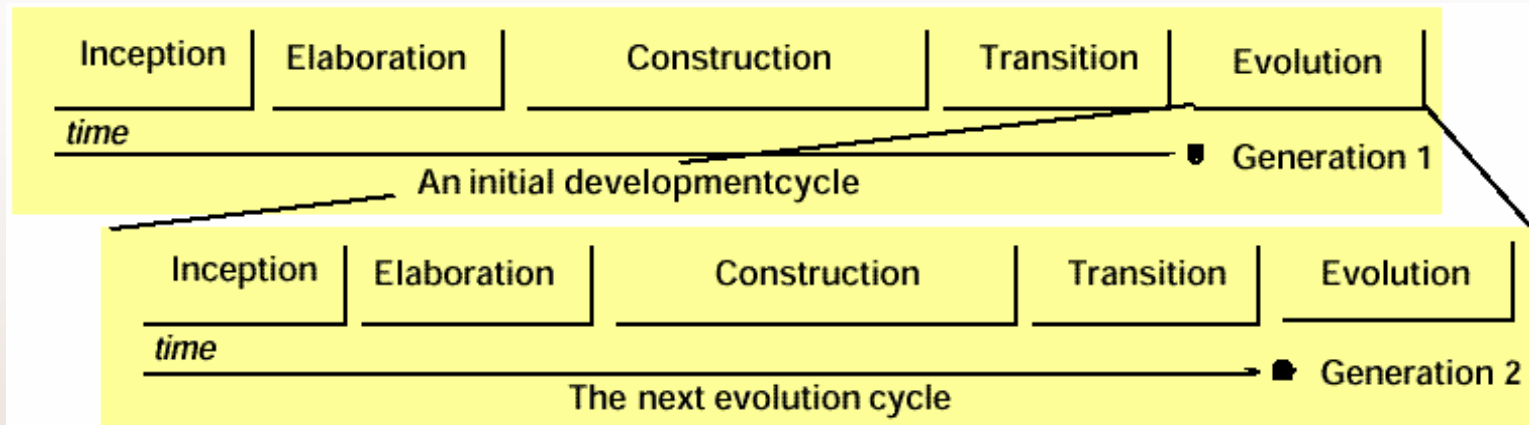


## Fase de Transição

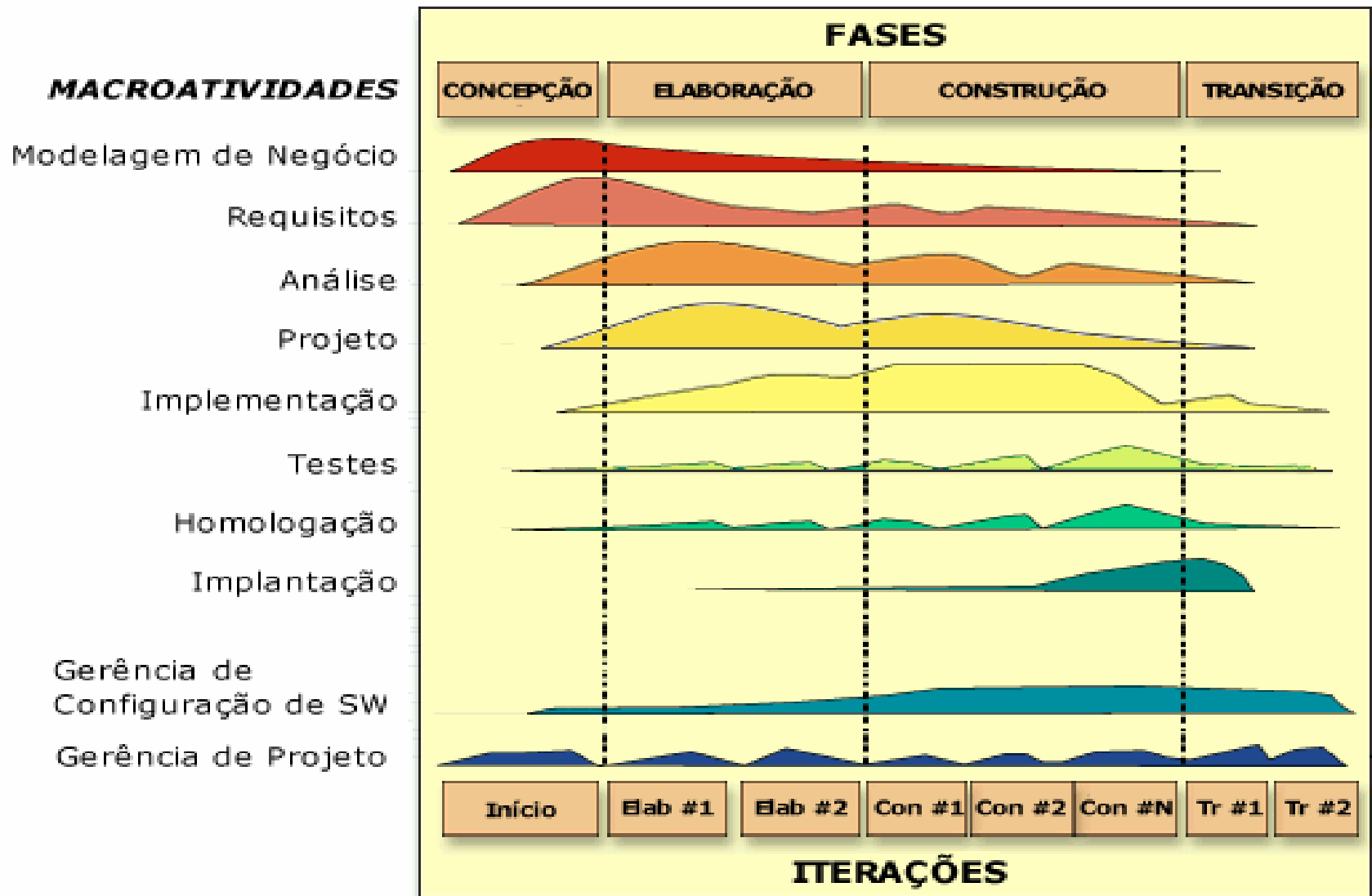
- Propósito:
  - Promover a transição do produto para a comunidade usuária.
- Produtos:
  - Uma série de liberações de executáveis.
  - Resultados do controle de qualidade.
  - Documentação atualizada do sistema e do usuário.
  - Análise do desempenho do projeto.



# Rational Unified Process



# Rational Unified Process



# Rational Unified Process



## Iterações

- Cada iteração passa por todos os aspectos do desenvolvimento de software:
  - Modelagem de Negócio
  - Análise e Desenho
  - Implementação
  - Teste
  - Disciplinas gerenciais são executadas em paralelo
- Cada liberação é uma “parte” completa do sistema final.

# Rational Unified Process

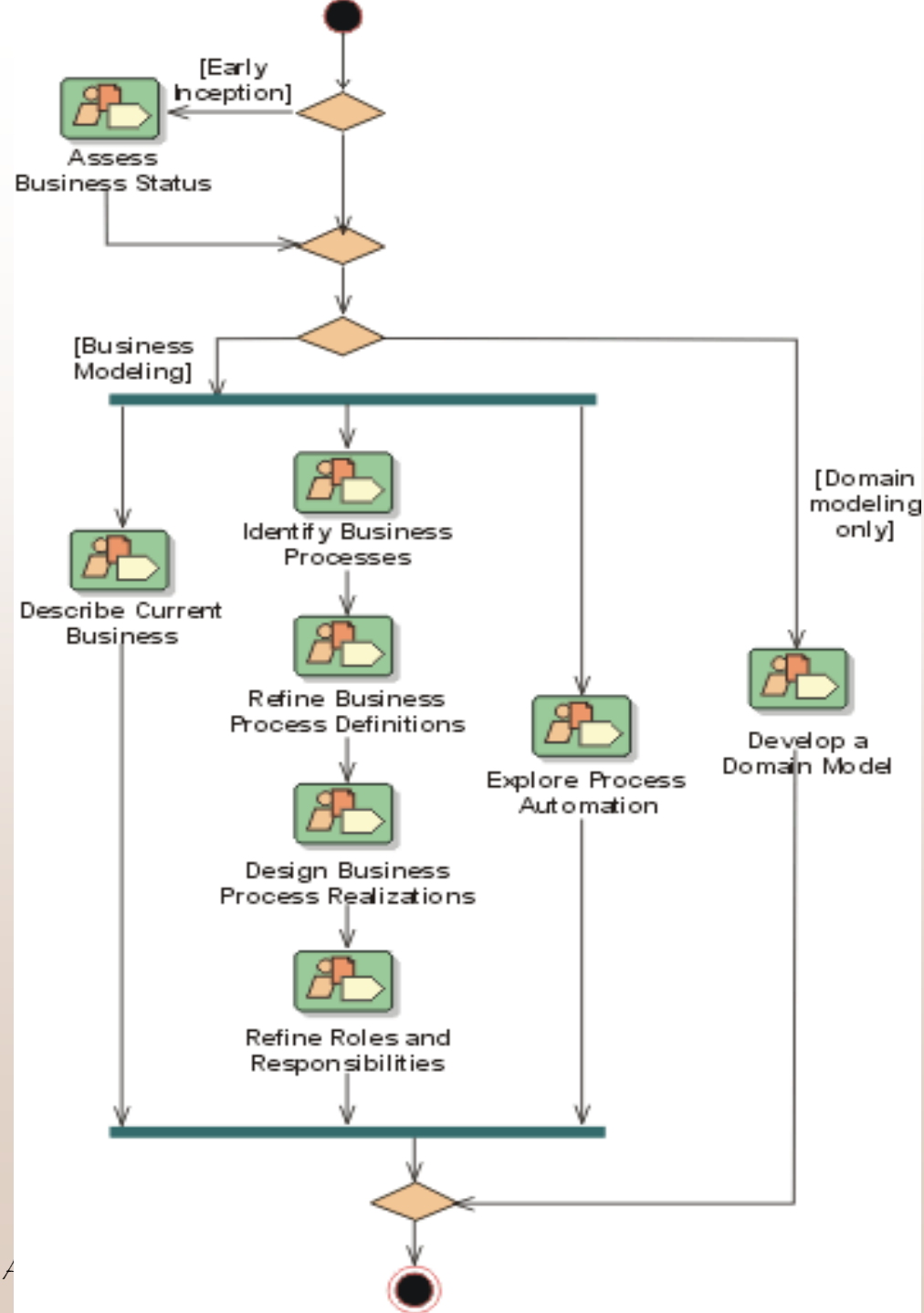


## Planejamento de Iterações

- Identificar e priorizar os riscos significativos do projeto.
- Selecionar um número pequeno de casos de uso que endereçam diretamente os riscos mais prioritários.
- Os casos de uso escolhidos são utilizados para:
  - Os desenvolvedores identificarem o que deve ser implementado pela iteração.
  - Os testadores desenvolverem os planos e procedimentos de teste para a iteração.
- Ao final da iteração:
  - Determinar que riscos foram reduzidos ou eliminados.
  - Determinar se novos riscos foram descobertos
  - Atualizar o plano para as iterações restantes.

# Rational Unified Process

## Disciplina de Modelagem de Negócio



# Rational Unified Process

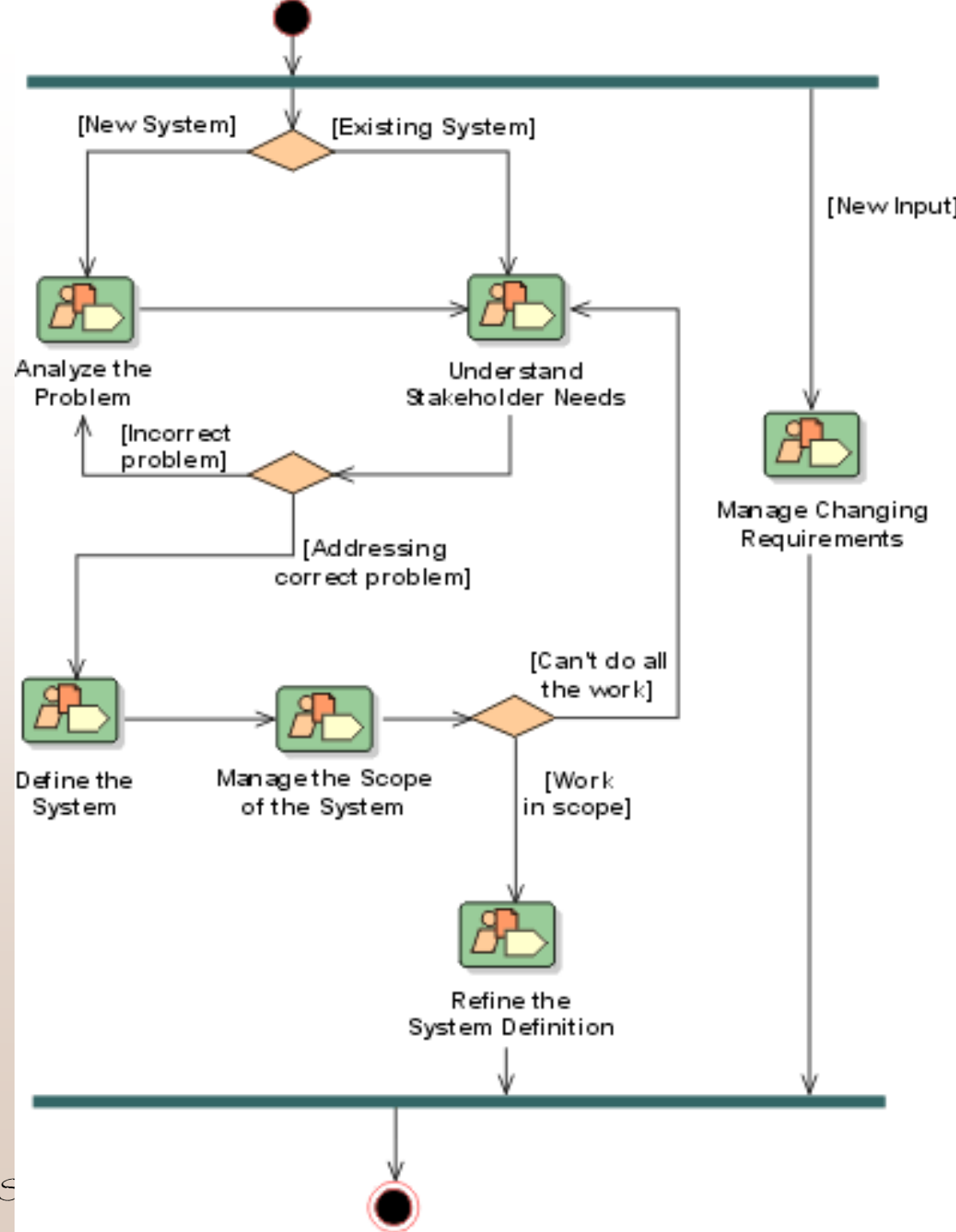


## Disciplina de Modelagem de Negócio

- Objetivos
  - Entender a estrutura e dinâmica da organização onde o sistema será implantado
  - Entender os problemas existentes atualmente nos processos da organização e identificar pontos candidatos a melhorias
  - Equalizar o entendimento dos processos da organização entre os clientes, usuários e desenvolvedores
  - Identificar, em linhas gerais, os requisitos que o sistema deve ter para atender à organização

# Rational Unified Process

## Disciplina de Requisitos



# Rational Unified Process



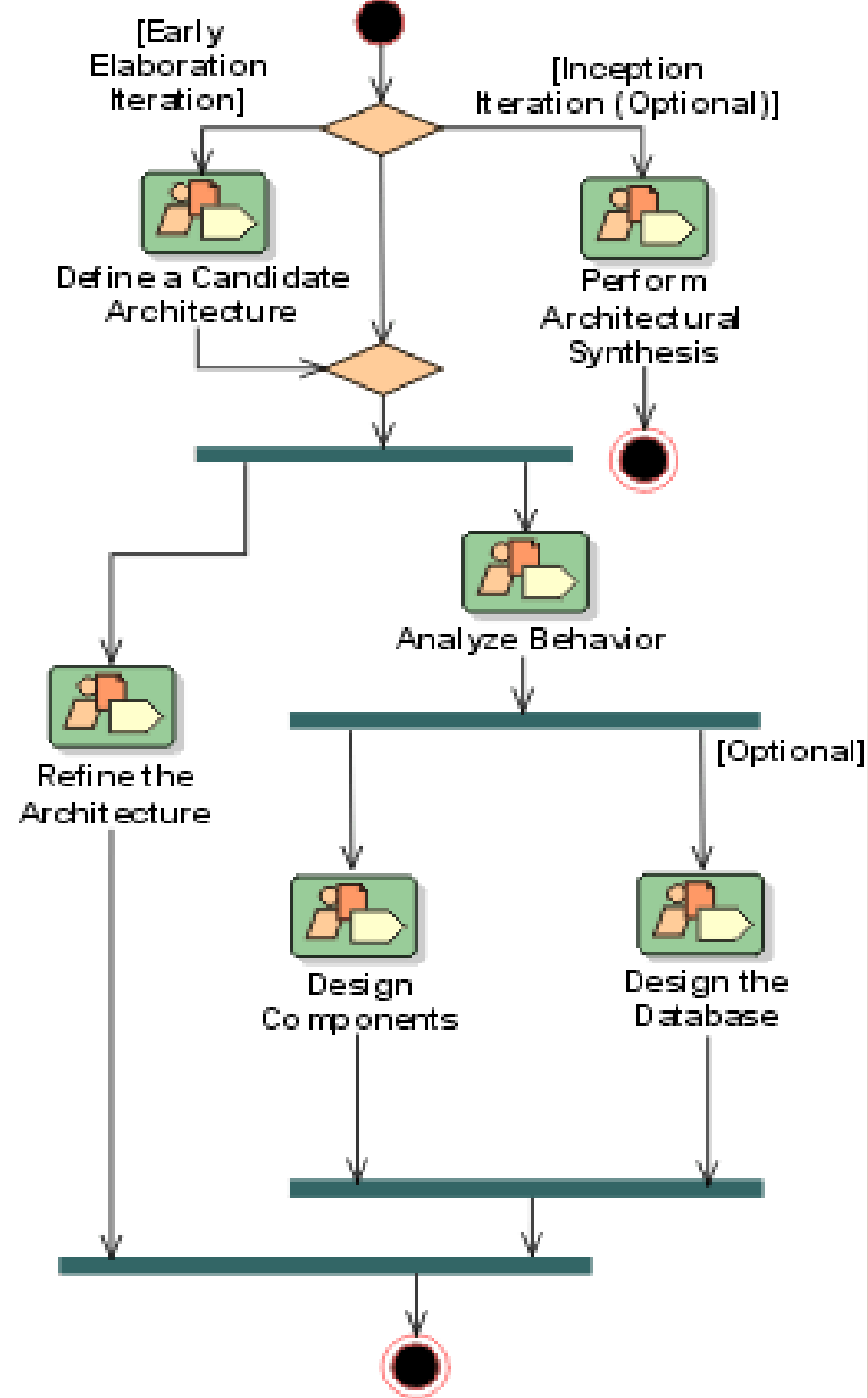
## Disciplina de Requisitos

- Objetivos
  - Estabelecer e manter um acordo entre clientes e outros stakeholders do que o sistema deve fazer
  - Fornecer aos desenvolvedores um melhor entendimento dos requisitos do sistema
  - Definir o escopo do sistema e seus limites
  - Fornecer bases para o planejamento do conteúdo técnico de cada iteração
  - Fornecer uma base para estimativas de custo e prazo para o desenvolvimento do sistema



# Rational Unified Process

## Disciplina de Análise e Projeto



# Rational Unified Process

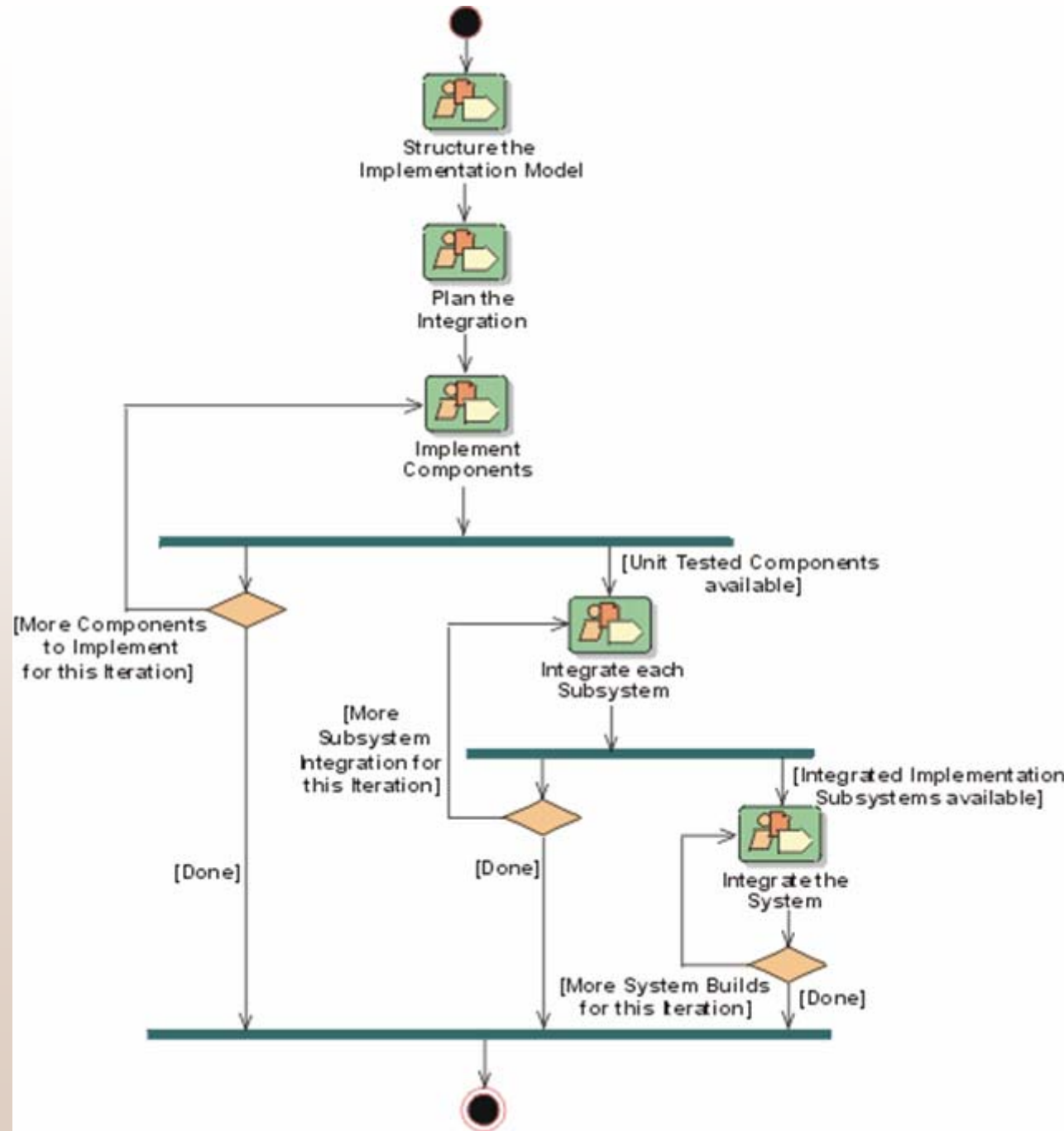


## Disciplina de Análise e Desenho

- Objetivos
  - Transformar os requisitos em um projeto do sistema a ser desenvolvido
  - Definir uma arquitetura robusta para o sistema
  - Adaptar o projeto (desenho) para o ambiente em que o sistema será implementado, preocupando-se com performance

# Rational Unified Process

## Disciplina de Implementação



# Rational Unified Process

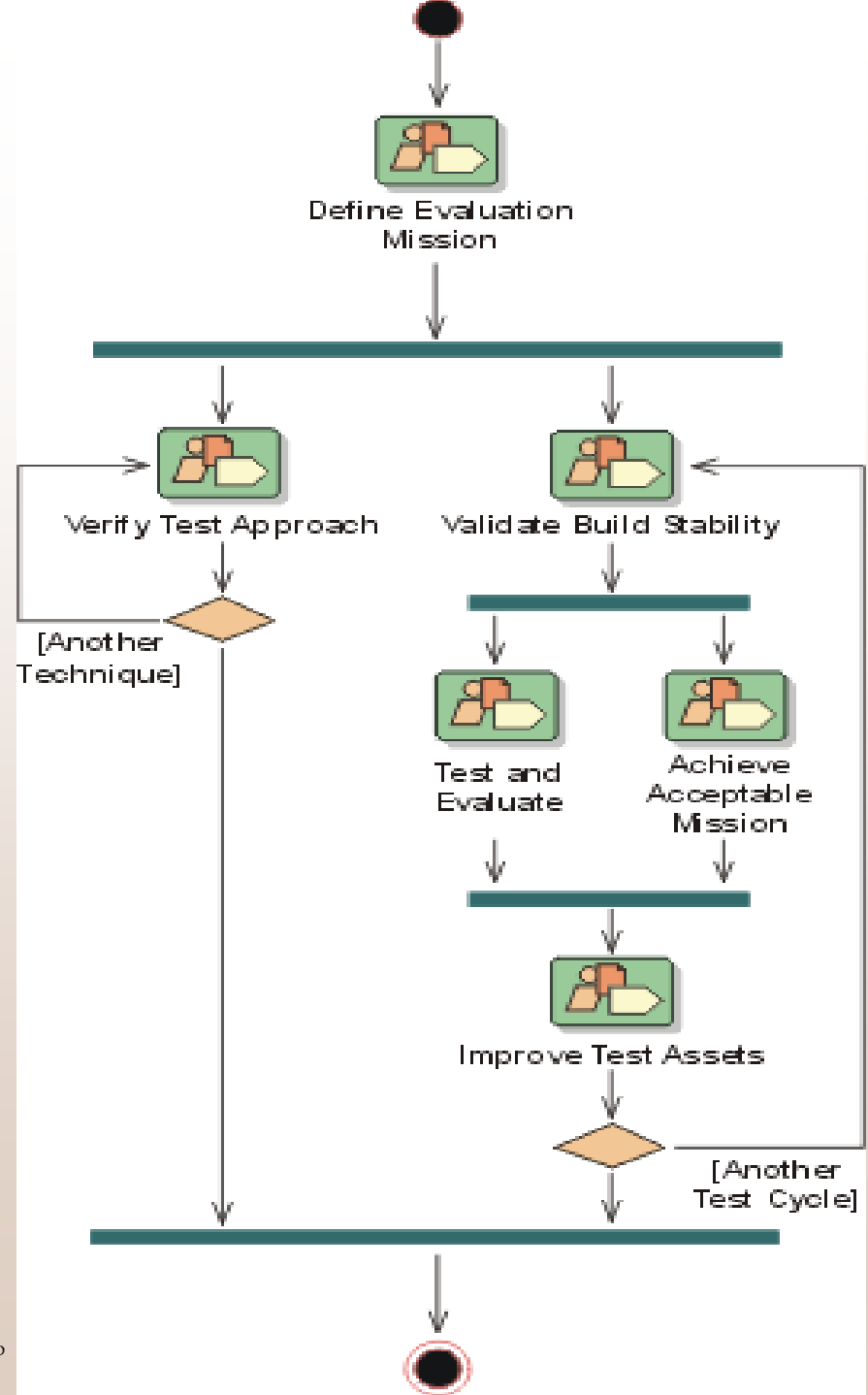


## Disciplina de Implementação

- Objetivos
  - Definir a organização do código em termos de subsistemas e camadas
  - Implementar classes e objetos em termos de componentes
  - Realizar os testes de unidade
  - Integrar os resultados produzidos por cada desenvolvedor, formando assim um produto executável, o sistema

# Rational Unified Process

## Disciplina de Teste



# Rational Unified Process

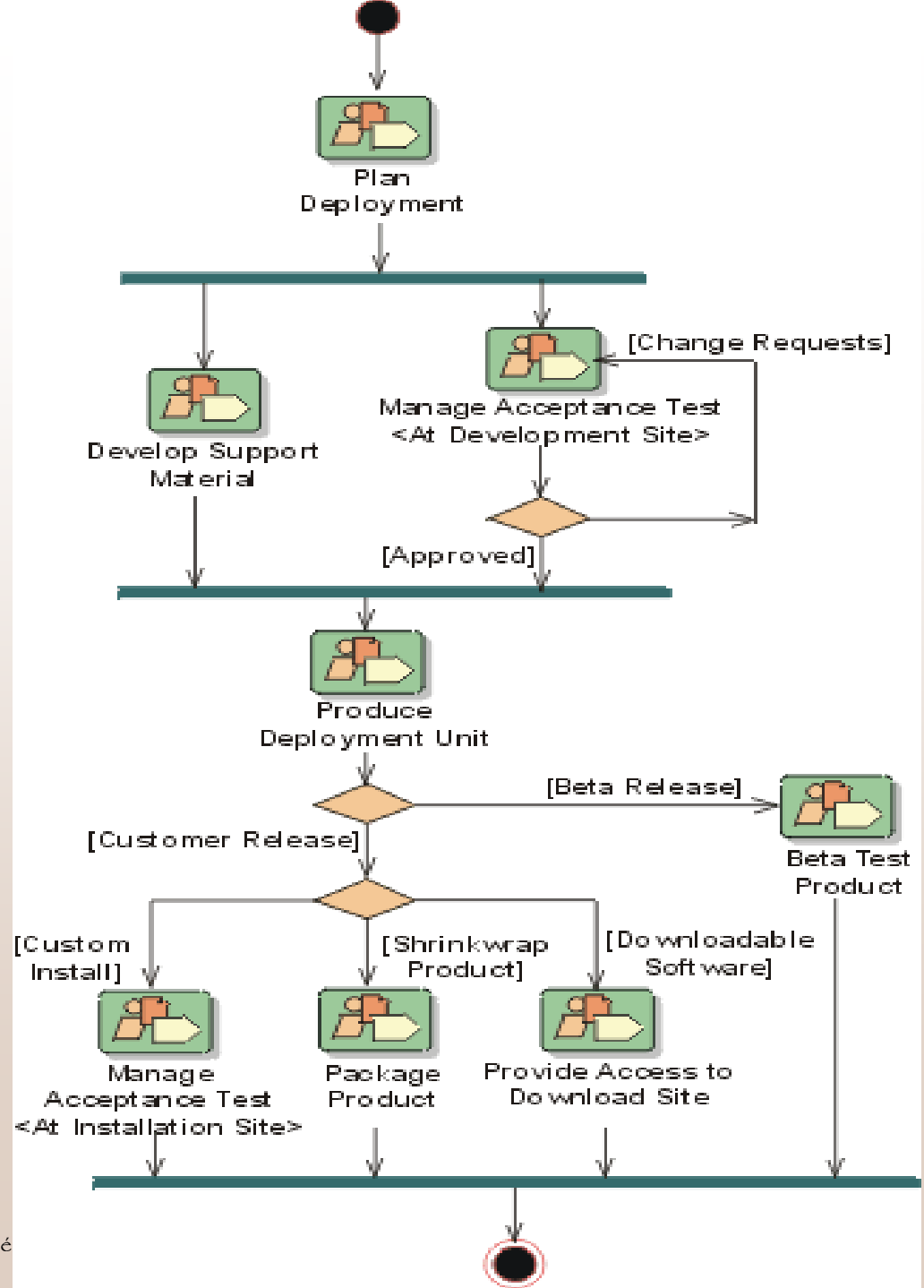


## Disciplina de Teste

- Objetivos
  - Encontrar e documentar defeitos
  - Validar as hipóteses levantadas no desenho e especificações de requisitos através de demonstrações concretas
  - Verificar se as funcionalidades do sistemas foram implementadas de acordo com o desenho
  - Verificar que os requisitos foram implementados de forma adequada

# Rational Unified Process

## Disciplina de Implantação



# Rational Unified Process



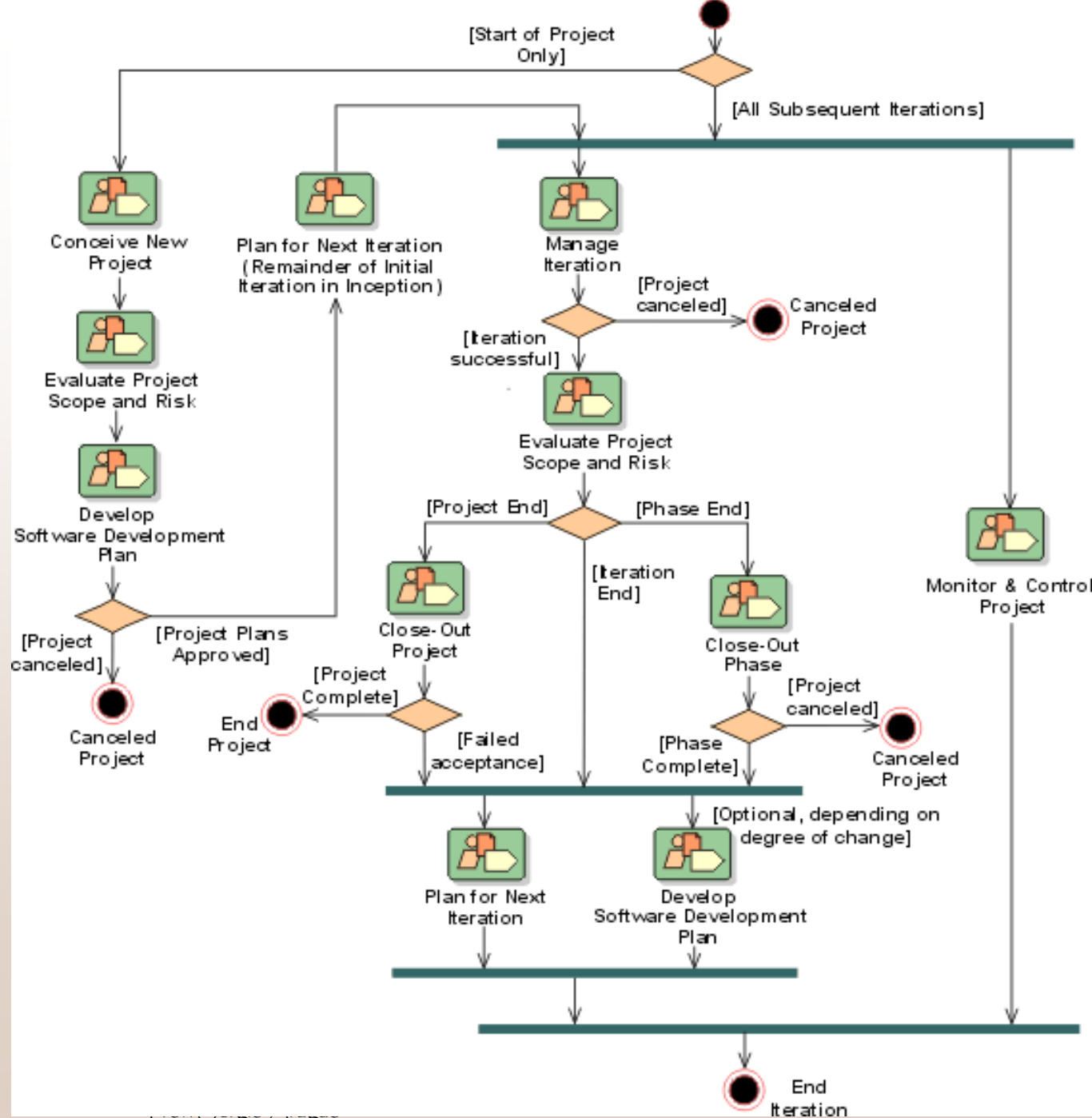
## Disciplina de Implantação

- Objetivos
  - Assegurar que o produto seja disponibilizado de forma correta para seus usuários finais, conforme especificado pelo cliente



# Rational Unified Process

## Disciplina de Gestão de Projetos



# Rational Unified Process

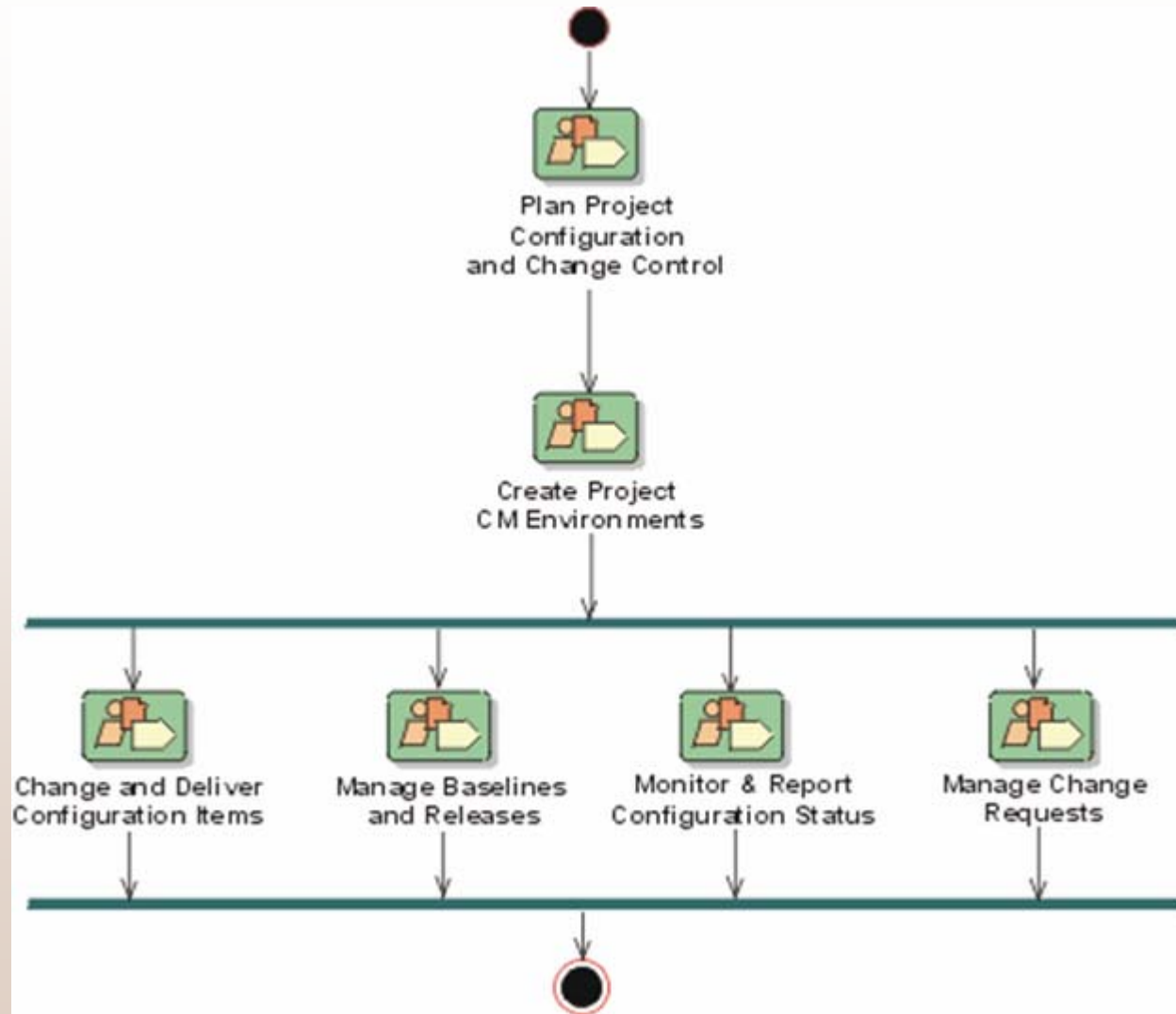


## Disciplina de Gestão de Projetos

- Objetivos
  - Gerir riscos
  - Planejar projetos iterativos, para o desenvolvimento como um todo e em cada iteração
  - Acompanhar o progresso dos projetos iterativos, coletar métricas para experiências futuras

# Rational Unified Process

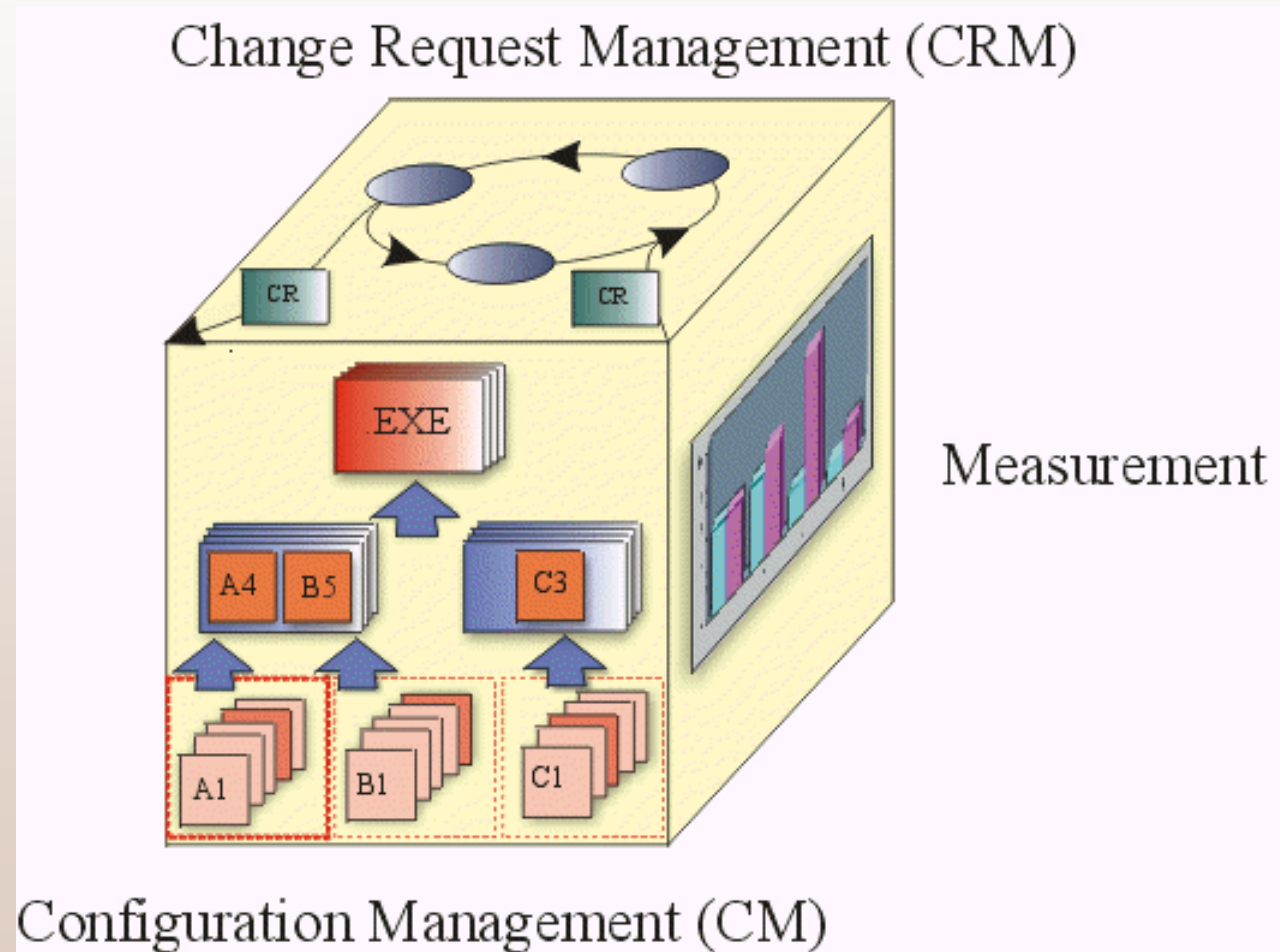
## Gestão de Configurações e Mudanças



# Rational Unified Process



## Gestão de Configurações e Mudanças



# Rational Unified Process

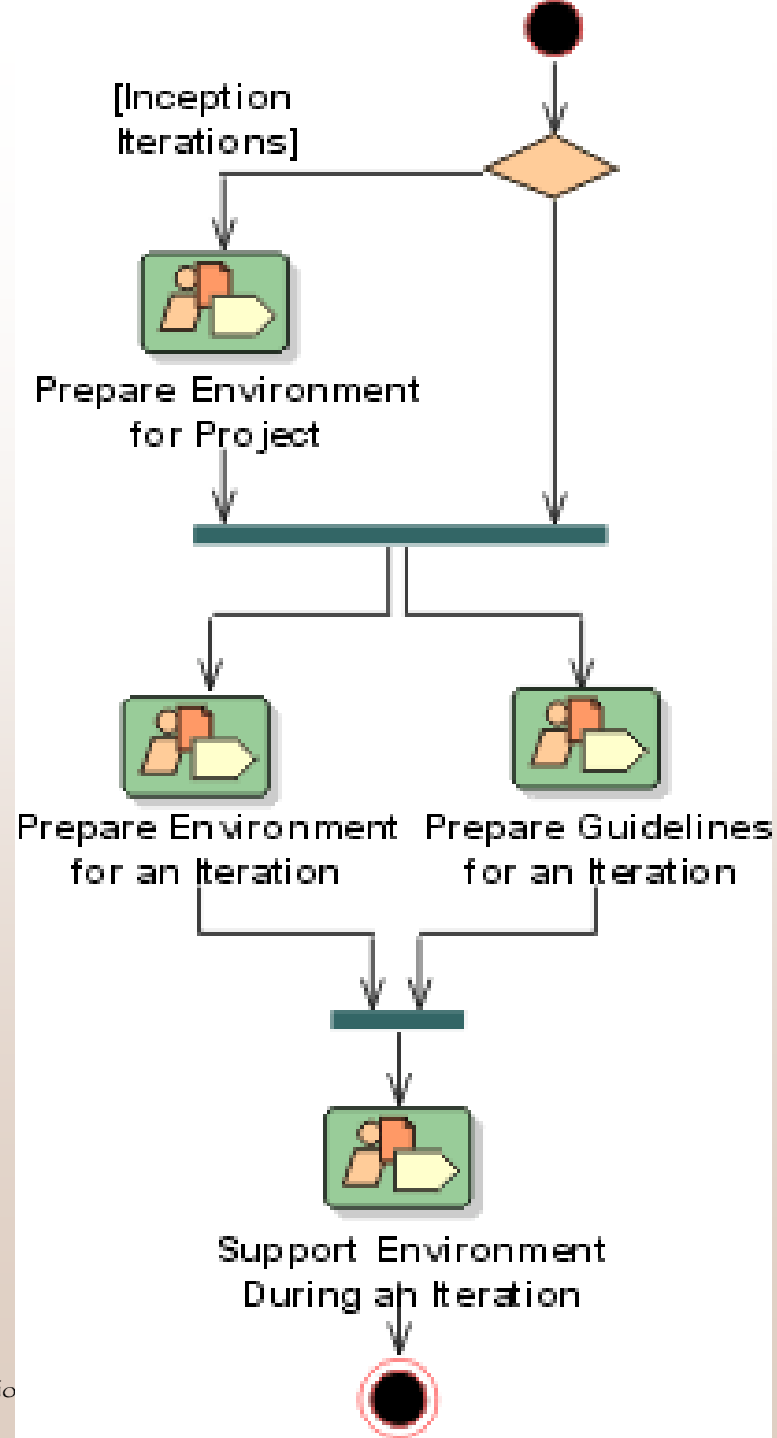


## Gestão de Configurações e Mudanças

- Objetivos
  - Garantir a integridade do produto e demais artefatos,
  - Garantir a completeza e correção do produto e suas versões,
  - dar suporte ao desenvolvimento,
  - Restringir mudanças nos artefatos baseando-se em políticas estabelecidas para o projeto,
  - Garantir a não ocorrência dos seguintes problemas: atualização simultânea, notificação limitada, múltiplas versões

# Rational Unified Process

## Ambiente



# Rational Unified Process



## Ambiente

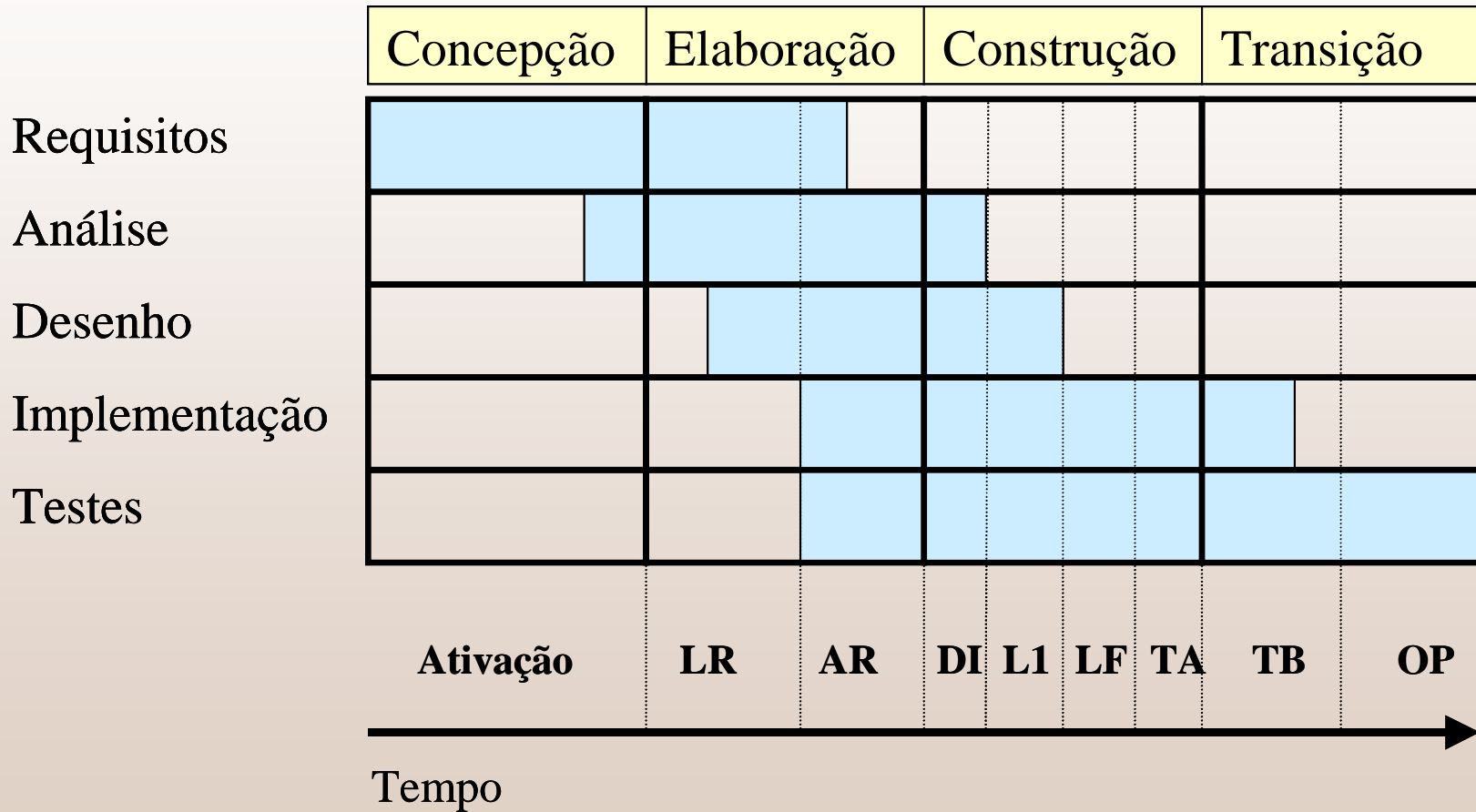
- Objetivos
  - Configurar o processo para um determinado projeto, criando modelos, guias para execução das atividades.
  - Disponibilizar o ambiente necessário ao desenvolvimento, tanto com relação ao processo quanto com relação às ferramentas a serem utilizadas

# O processo Praxis

- Praxis - Processo para Aplicativos Extensíveis Iterativos.
- Ênfase na Tecnologia Orientada a Objetos.
  - Notação principal - UML.
- Ênfase na extensibilidade:
  - entrega evolutiva por projeto (versão);
  - espiral por produto.



# O processo Praxis



# O processo Praxis

- Fases do Praxis:

Fase	Descrição
<b>Concepção</b>	Fase na qual necessidades dos usuários e conceitos da aplicação são analisados o suficiente para justificar a especificação de um produto de software, resultando em uma proposta de especificação.
<b>Elaboração</b>	Fase na qual a especificação do produto é detalhada o suficiente para modelar conceitualmente o domínio do problema, validar os requisitos em termos deste modelo conceitual e permitir um planejamento acurado da fase de construção.
<b>Construção</b>	Fase na qual é desenvolvida (desenhada, implementada e testada) uma liberação completamente operacional do produto, que atende aos requisitos especificados.
<b>Transição</b>	Fase na qual o produto é colocado à disposição de uma comunidade de usuários para testes finais, treinamento e uso inicial.

# O processo Praxis

- Fluxos do Praxis:

Fluxo	Descrição
<b>Requisitos</b>	Fluxo que visa obter um conjunto de requisitos de um produto, acordado entre cliente e fornecedor.
<b>Análise</b>	Fluxo que visa detalhar, estruturar e validar os requisitos, em termos de um modelo conceitual do problema, de forma que estes possam ser usados como base para o planeamento e acompanhamento detalhados da construção do produto.
<b>Desenho</b>	Fluxo que visa formular um modelo estrutural do produto que sirva de base para a implementação, definindo os componentes a desenvolver e a reutilizar, assim como as interfaces entre si e com o contexto do produto.
<b>Implementação</b>	Fluxo que visa detalhar e implementar o desenho através de componentes de código e de documentação associada.
<b>Testes</b>	Fluxo que visa verificar os resultados da implementação através do planeamento, desenho e realização de baterias de testes.

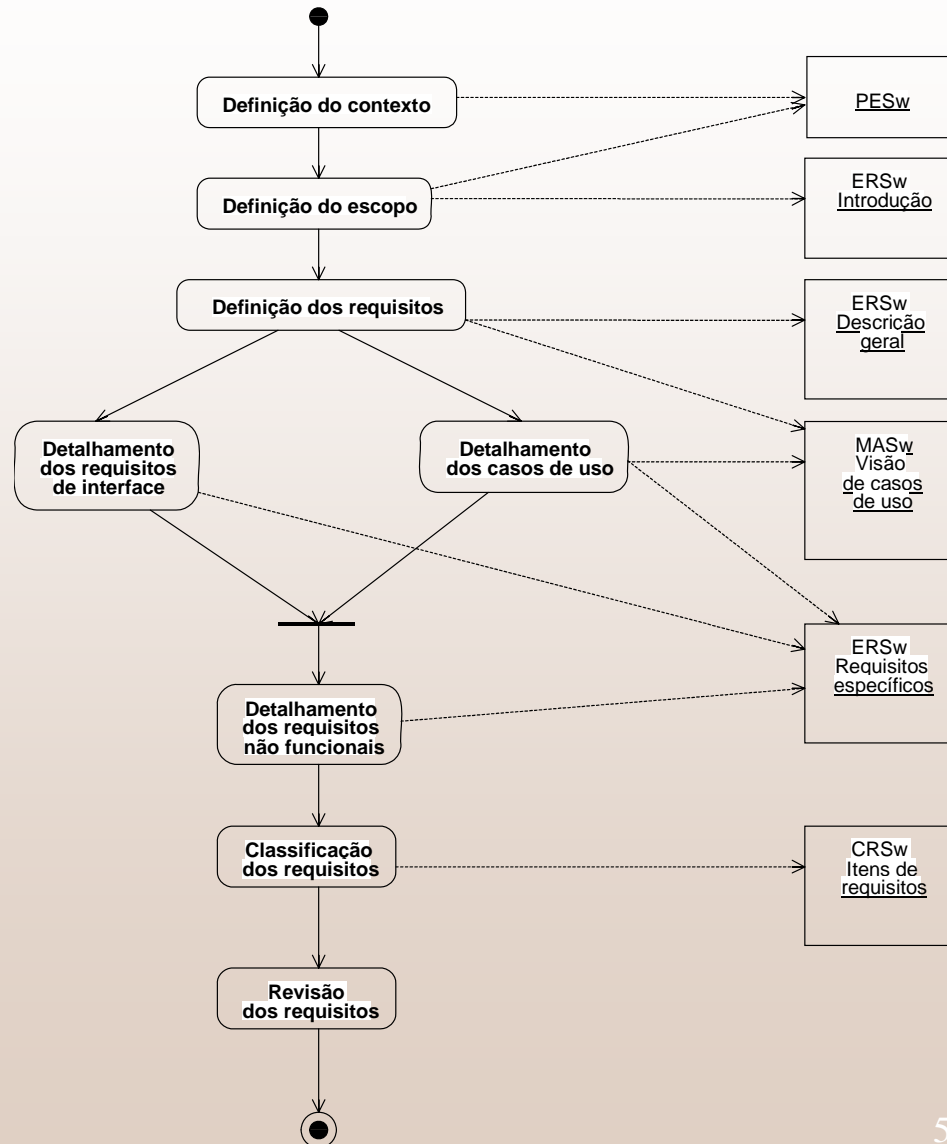
# O processo Praxis

- Iterações do Praxis:

Fase	Iteração
Concepção	Ativação
Elaboração	Levantamento dos Requisitos
	Análise dos Requisitos
Construção	Desenho Inicial
	Liberação 1
	Liberação ...
	Liberação Final
	Testes Alfa
Transição	Testes Beta
	Operação Piloto

# O processo Praxis

- Exemplo de diagrama de atividades de fluxo:



# O processo Praxis

- Atividades da iteração - Ativação:

Fluxo	Tarefas
<b>Requisitos</b>	Definição do escopo do produto. Definição dos requisitos (preliminar).
<b>Análise</b>	Estudos de viabilidade (opcional).
<b>Desenho</b>	Esboço da arquitetura do produto (opcional). Estudos de viabilidade (opcional).
<b>Implementação</b>	Prototipagem dos requisitos (opcional).
<b>Testes</b>	Testes dos protótipos dos requisitos (opcional).
<b>Gestão</b>	Levantamento das metas gerenciais. Estimativas da fase de Elaboração. Elaboração de proposta de especificação.

# O processo Praxis

- Atividades da iteração - Levantamento dos Requisitos:

Fluxo	Tarefas
<b>Requisitos</b>	Levantamento completo dos requisitos. Detalhamento das interfaces. Detalhamento dos casos de uso. Detalhamento dos requisitos não funcionais.
<b>Análise</b>	Estudos de viabilidade (opcional).
<b>Desenho</b>	Estudos de viabilidade (opcional).
<b>Implementação</b>	Prototipagem dos requisitos (opcional).
<b>Testes</b>	Testes dos protótipos dos requisitos (opcional).
<b>Gestão</b>	Cadastramento dos requisitos.

# O processo Praxis

- Atividades da iteração - Análise dos Requisitos:

Fluxo	Tarefas
<b>Requisitos</b>	Revisão e modificação dos requisitos (se necessário).
<b>Análise</b>	Identificação das classes. Identificação dos atributos e relacionamentos. Realização dos casos de uso. Revisão e iteração.
<b>Desenho</b>	Estudos de viabilidade (opcional). Desenho arquitetônico.
<b>Implementação</b>	Prototipagem dos requisitos (opcional).
<b>Testes</b>	Testes dos protótipos dos requisitos (opcional).
<b>Gestão</b>	Cadastramento dos itens de análise no cadastro de requisitos. Planejamento do desenvolvimento. Planejamento da qualidade.



# O processo Praxis

- Atividades da iteração - Desenho Inicial:

Fluxo	Tarefas
<b>Requisitos</b>	Revisão e modificação dos requisitos (se necessário).
<b>Análise</b>	Revisão e modificação do modelo de análise (se necessário).
<b>Desenho</b>	Desenho dos subsistemas. Desenho do acesso a dados persistentes. Desenho das interfaces de usuário. Elaboração dos casos de uso de desenho. Desenho das liberações.
<b>Implementação</b>	Prototipagem de problemas de desenho (opcional).
<b>Testes</b>	Planejamento e desenho dos testes de aceitação.
<b>Gestão</b>	Cadastramento dos itens de teste no cadastro de requisitos. Cadastramento dos itens de desenho no cadastro de requisitos. Planejamento detalhado das liberações. Atualização dos planos de desenvolvimento e da qualidade.

# O processo Praxis

- Atividades da iteração - Liberação:

Fluxo	Tarefas
<b>Requisitos</b>	Revisão e modificação dos requisitos (se necessário).
<b>Análise</b>	Revisão e modificação do modelo de análise (se necessário).
<b>Desenho</b>	Revisão e modificação do modelo de desenho (se necessário).
<b>Implementação</b>	Desenho detalhado dos componentes desta liberação. Codificação dos componentes desta liberação. Compilação dos componentes desta liberação.
<b>Testes</b>	Planejamento e desenho dos testes de integração da liberação. Planejamento e desenho dos testes de unidade da liberação. Realização dos testes de unidade da liberação. Realização dos testes de integração da liberação.
<b>Gestão</b>	Revisão e modificação dos planos de desenvolvimento e da qualidade (se necessário).

# O processo Praxis

- Atividades da iteração - Testes Alfa:

Fluxo	Tarefas
<b>Requisitos</b>	Revisão e modificação dos requisitos (se necessário).
<b>Análise</b>	Revisão e modificação do modelo de análise (se necessário).
<b>Desenho</b>	Revisão e modificação do desenho de alto nível (se necessário).
<b>Implementação</b>	Revisão e modificação do desenho detalhado e código (se necessário). Produção da documentação de usuário.
<b>Testes</b>	Realização dos testes alfa (aceitação no ambiente dos desenvolvedores).
<b>Gestão</b>	Planejamento detalhado da Transição. Atualização dos planos de desenvolvimento e da qualidade.

# O processo Praxis

- Atividades da iteração - Testes Beta:

Fluxo	Tarefas
<b>Requisitos</b>	Revisão e modificação dos requisitos (se necessário).
<b>Análise</b>	Revisão e modificação do modelo de análise (se necessário).
<b>Desenho</b>	Revisão e modificação do desenho de alto nível (se necessário).
<b>Implementação</b>	Revisão e modificação do desenho detalhado e código (se necessário). Revisão e modificação da documentação de usuário (se necessário).
<b>Testes</b>	Realização dos testes beta (aceitação no ambiente dos usuários).
<b>Gestão</b>	Revisão e modificação dos planos de desenvolvimento e da qualidade (se necessário).

# O processo Praxis

- Atividades da iteração - Operação Piloto:

Fluxo	Tarefas
<b>Requisitos</b>	Revisão e modificação dos requisitos (se necessário).
<b>Análise</b>	Revisão e modificação do modelo de análise (se necessário).
<b>Desenho</b>	Revisão e modificação do desenho de alto nível (se necessário).
<b>Implementação</b>	Revisão e modificação do desenho detalhado e código (se necessário). Revisão e modificação da documentação de usuário (se necessário).
<b>Testes</b>	Revisão e modificação da documentação de testes (se necessário).
<b>Gestão</b>	Balanco final do projeto. Produção do Relatório Final do Projeto.

# O processo Praxis

- Documentos permanentes do Praxis:

Nome	Sigla	Descrição
Proposta de Especificação do Software	PESw	Documento que delimita preliminarmente o escopo de um projeto, contendo um plano da fase de Elaboração.
Especificação dos Requisitos do Software	ERSw	Documento que descreve, de forma detalhada, o conjunto de requisitos especificados para um produto de software.
Plano de Desenvolvimento do Software	PDSw	Documento que descreve, de forma detalhada, os compromissos que o fornecedor assume em relação ao projeto, quanto a recursos, custos, prazos, riscos e outros aspectos gerenciais.
Plano da Qualidade do Software	<u>PQSw</u>	Documento que descreve, de forma detalhada, os procedimentos de garantia da qualidade que serão adotados no projeto.
Descrição do Desenho do Software	<u>DDSw</u>	Documento que descreve, de forma detalhada, os aspectos mais importantes do desenho do software.
Descrição dos Testes do Software	DTSw	Documento que descreve, de forma detalhada, os planos e especificações dos testes que serão executados.
Manual do Usuário do Software	MUSw	Documento que serve de referência para uso do produto.

# O processo Praxis

- Modelos permanentes do Praxis:

Nome	Sigla	Descrição	Ferramentas aplicáveis
Cadastro dos Requisitos do Software	CRS <sub>sw</sub>	Modelo que contém os requisitos levantados, assim como referências aos itens correspondentes dos modelos seguintes.	Planilha, banco de dados
Modelo de Análise do Software	MAS <sub>sw</sub>	Modelo que detalha os conceitos do domínio do problema resolver que sejam relevantes para a validação dos requisitos.	Ferramenta de modelagem orientada a objetos
Memória de Planejamento do Projeto do Software	MPPS <sub>sw</sub>	Modelo que contém a informação necessária para o planejamento e acompanhamento de tamanhos, esforços, custos, prazos e riscos do projeto.	Planilha, ferramenta de gestão de projetos
Modelo de Desenho do Software	MDS <sub>sw</sub>	Modelo que detalha a estrutura lógica e física do produto, em termos de seus componentes.	Ferramenta de modelagem orientada a objetos
Bateria de Testes de Regressão do Software	BTRS <sub>sw</sub>	Conjunto dos scripts dos testes de regressão.	Ferramenta de desenvolvimento, ferramenta de testes
Códigos-fonte do Software	CFS <sub>sw</sub>	Conjunto dos códigos-fonte produzidos.	Ferramenta de desenvolvimento
Códigos Executáveis do Software	CES <sub>sw</sub>	Conjunto dos códigos executáveis produzidos.	Ferramenta de desenvolvimento

# O processo Praxis

- Relatórios do Praxis:

Nome	Sigla	Descrição
Relatórios dos Testes do Software	RTSw	Conjunto dos relatórios que descrevem os resultados dos testes realizados.
Relatórios de Revisão do Software	RRSw	Conjunto dos relatórios que descrevem as conclusões das revisões realizadas.
Relatórios das Auditorias da Qualidade do Software	RAQSw	Conjunto dos relatórios que descrevem as conclusões das auditorias da qualidade realizadas.
Relatórios de Acompanhamento do Projeto do Software	RAPSw	Conjunto dos relatórios de acompanhamento do projeto, que relatam esforços, custos, prazos e riscos do período relatado, comparados com o que foi planejado.
Relatório Final do Projeto do Software	RFPSw	Relatório de balanço final do projeto.



# O processo Praxis

- Procedimentos de controle:
  - revisões técnicas e inspeções;
  - avaliações dos usuários;
  - auditorias da qualidade;
  - revisões gerenciais;
  - aceitação pelo cliente.

# Exercício

- Comparação entre o RUP e Praxis
  - Aponte pelo menos três aspectos semelhantes nos dois processos e explique estes aspectos;
  - Aponte pelo menos três diferenças e explique.