

Universidade Estadual de Santa Cruz - UESC

PROCESSOS DE SOFTWARE

Objetivos do estudo

- Introduzir modelos de processo de software
- Descrever a variedade de modelos de processo
- Descrever modelos de processo para engenharia de requisitos, desenvolvimento de software, teste e evolução
- Apresentar a tecnologias RUP e CASE para atividades de processo de software

Processos de Software

- Conjuntos de atividades técnicas para especificar, projetar, implementar e testar sistemas de software.

Processo de Software - conceito

- É representação abstrata num conjunto estruturado de atividades requeridas para desenvolver um sistema de software:

Especificação,
Análise,
Projeto,
Implementação,
Validação,
Evolução.

Modelos Gerais e Interação entre Processos

Modelo Sequencial

- Distingue as fases de especificação, desenvolvimento, validação e evolução

Desenvolvimento evolucionário

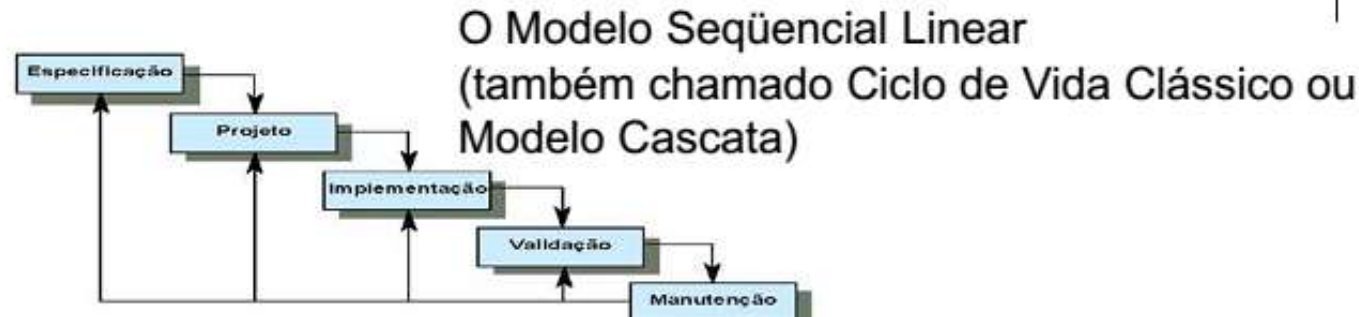
- Intercala atividades de especificação, desenvolvimento e validação

Desenvolvimento por reutilização

- O sistema é montado a partir de componentes existentes (CBSE)

Processos de Software

Modelo Seqüencial



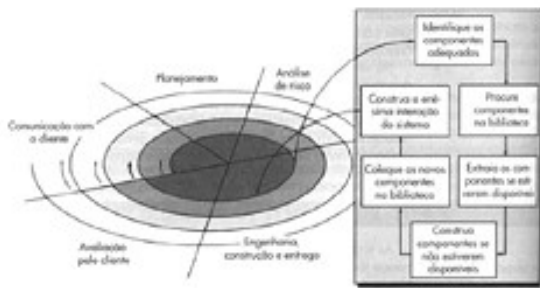
Desenvolvimento evolucionário



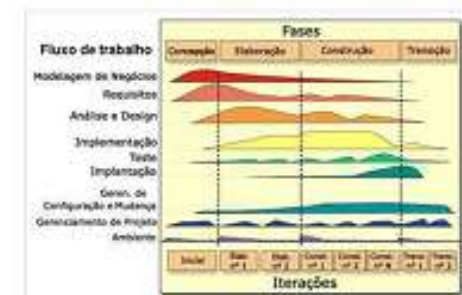
O Paradigma de Prototipação



Desenvolvimento por reutilização



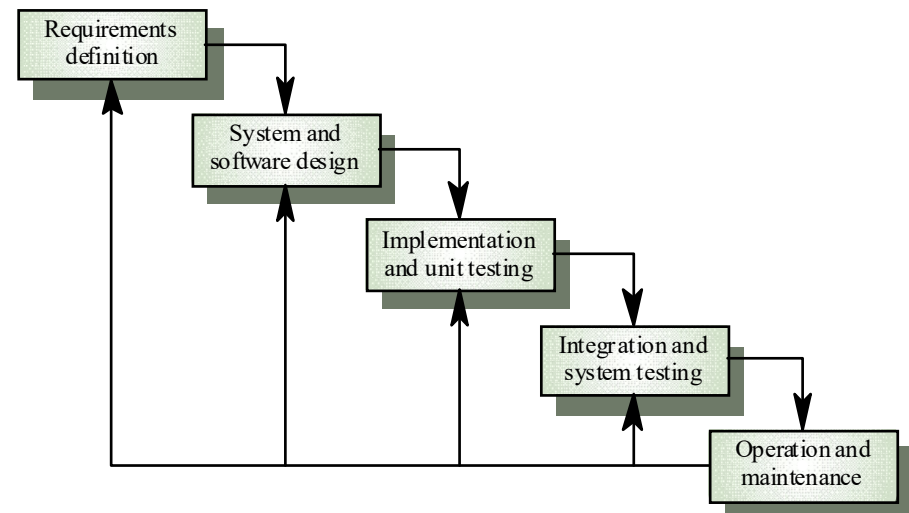
O Modelo Baseado em Componentes



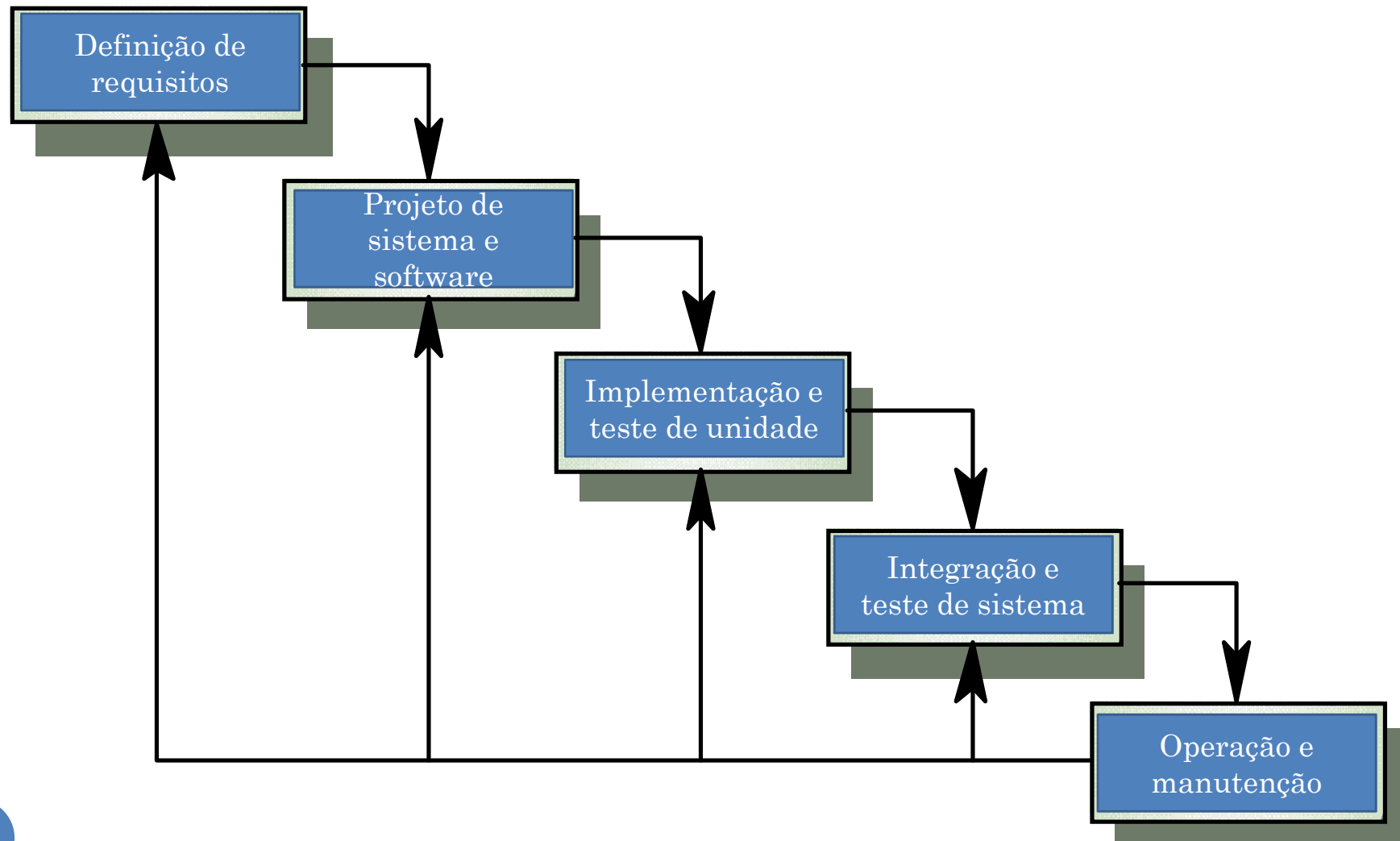
Processo Unificado

Modelo em Cascata - definição

Considera atividades fundamentais do processo pelo encadeamento de uma fase com outra do modelo num ciclo de vida do software.



Modelo Cascata - fases



Modelo em Cascata - fases

Análise e definição de requisitos

Serviços, restrições e objetivos do sistema



Projeto do sistema e do software

Identificação e descrição do sistema; arquitetura geral



Implementação e teste da unidade

Realização de programas ou unidades programadas



Integração e teste do sistema

Programas são integrados e testados como sistema completo



Operação e manutenção

O sistema é instalado e colocado em operação

Desvantagens

- Divisão inflexível do projeto em diferentes estágios
- Difícil responder aos requisitos mutáveis dos clientes
- Modelo apropriado somente quando os requisitos são bem entendidos
- Dificuldade de manutenção da fase anterior

Modelo Cascata



Contribuição

- O Modelo de processo em Cascata trouxe contribuições importantes para o processo de desenvolvimento de software:
 - Imposição de **disciplina**, **planejamento** e **gerenciamento**

Modelo Evolucionário

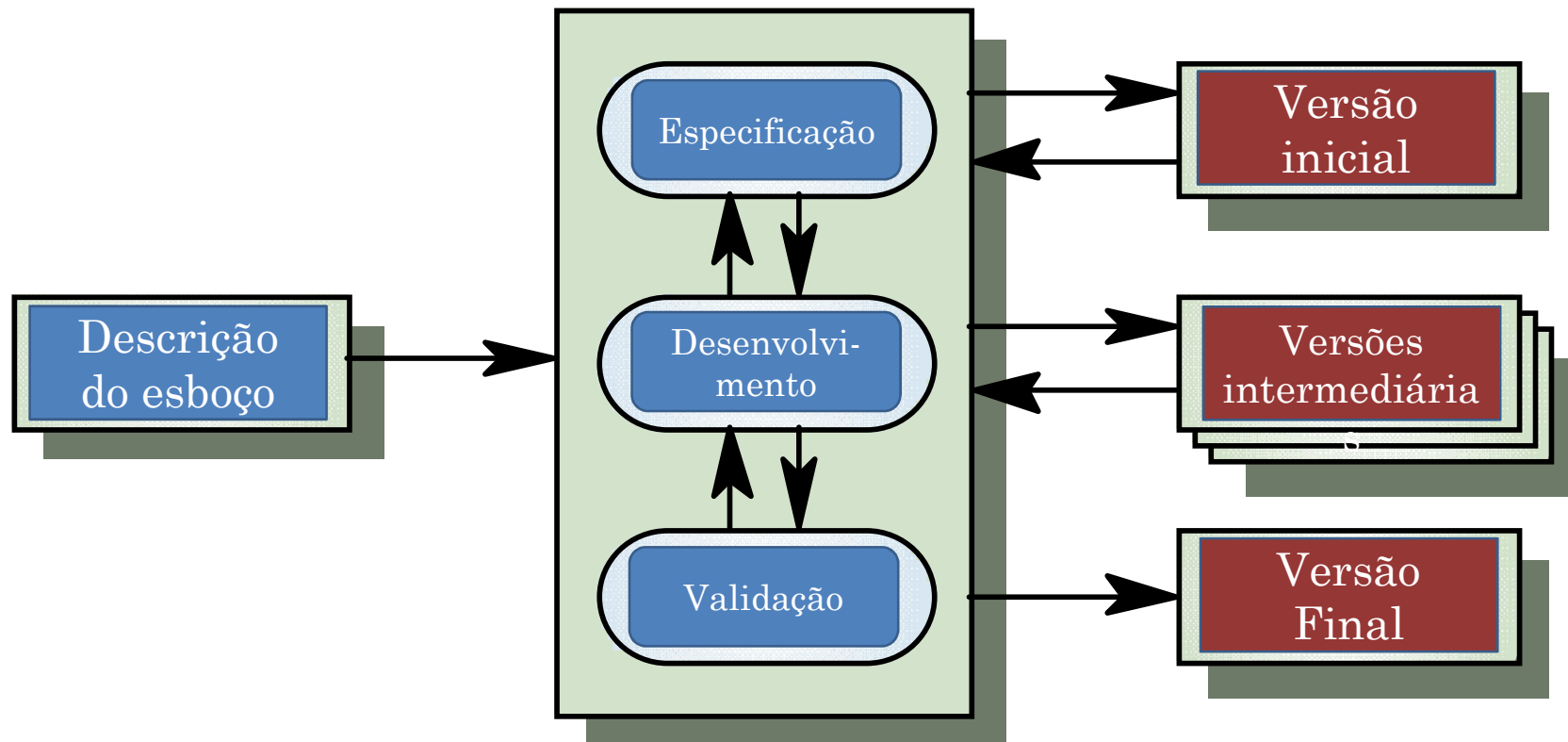
(prototipação)

Modelo Evolucionário (prototipação)

Desenvolvimento de implementação inicial para levantar requisitos do usuário, sendo refinado por meio de várias versões até que o desenvolvimento seja adequado.

Desenvolvimento Evolucionário

Atividades
simultâneas



Tipos de Desenvolvimento Evolucionários

Desenvolvimento Exploratório

- Sistema é desenvolvido pelas partes **compreendidas** e evolui com novas características até o final.

Prototipação *Throwaway*

- Desenvolvimento evolucionário a partir de protótipo e se concentra nos requisitos **mal compreendidos** para evolui.

Desenvolvimento Incremental

Processo Incremental

A funcionalidade do sistema estará **disponível mais cedo**;

Incrementos na forma de **protótipos** para elicitación de requisitos para incrementos posteriores;

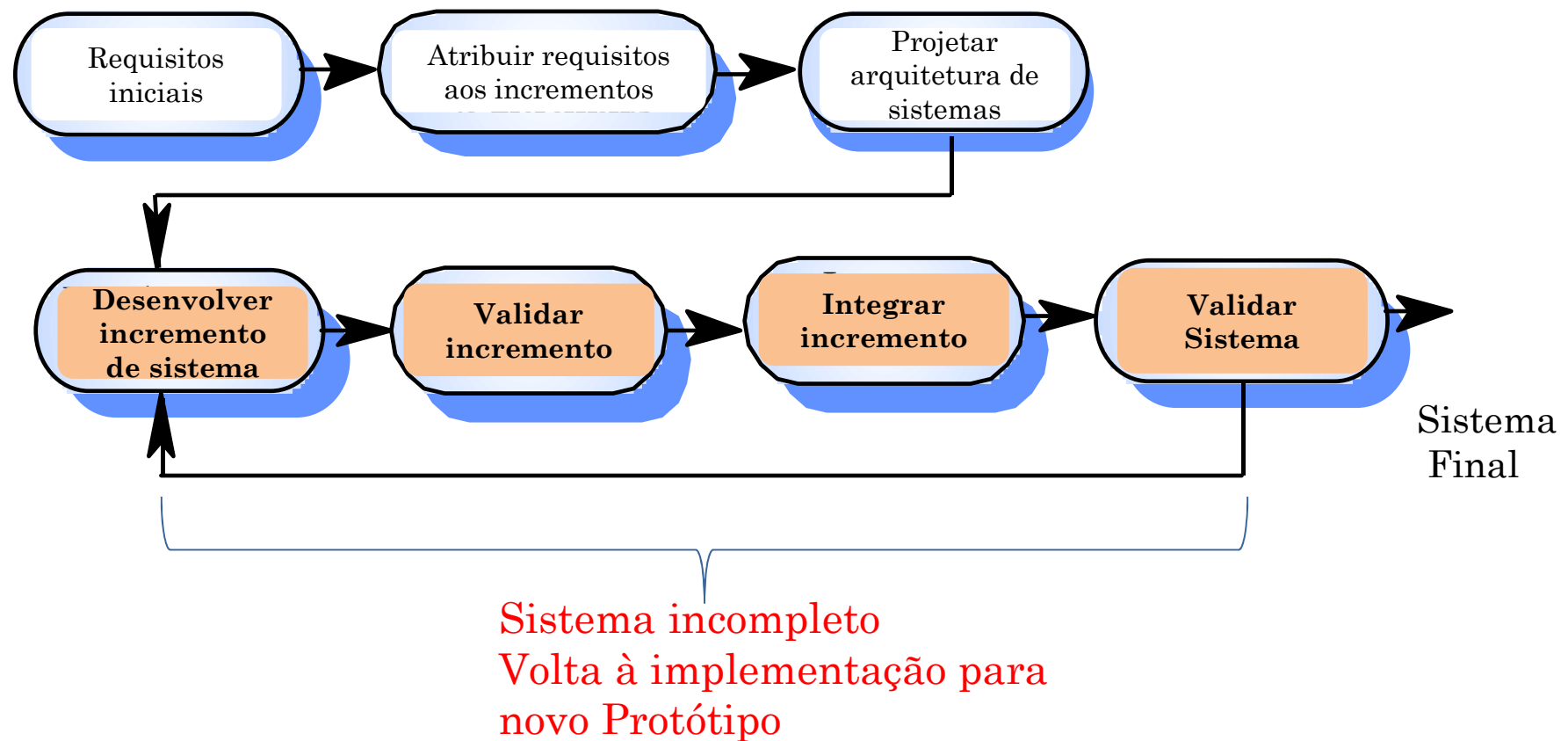
Menores riscos de falha no projeto em geral;

Os serviços do sistema de alta prioridade **recebem a maioria dos testes**.

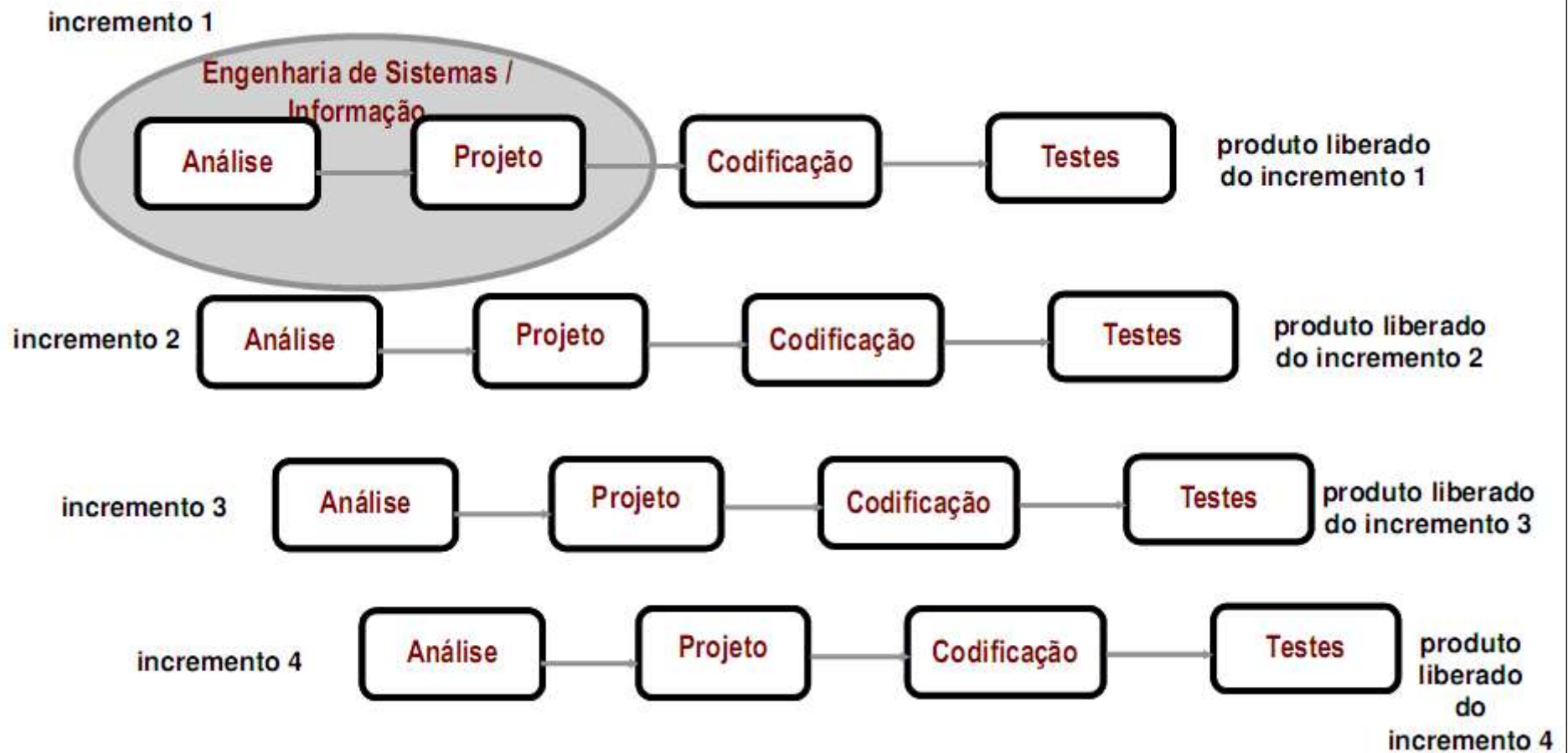
Modelo Incremental

- Combina elementos do Modelo Linear com a filosofia da Prototipação.
- Aplica sequências lineares numa abordagem de “saltos” à medida do progresso.
- Cada sequência linear produz um incremento do software.
- O processo se repete até que um produto completo seja produzido.
- Difere da Prototipação, pois a cada incremento produz uma versão do software.

Desenvolvimento Incremental



Modelo Incremental



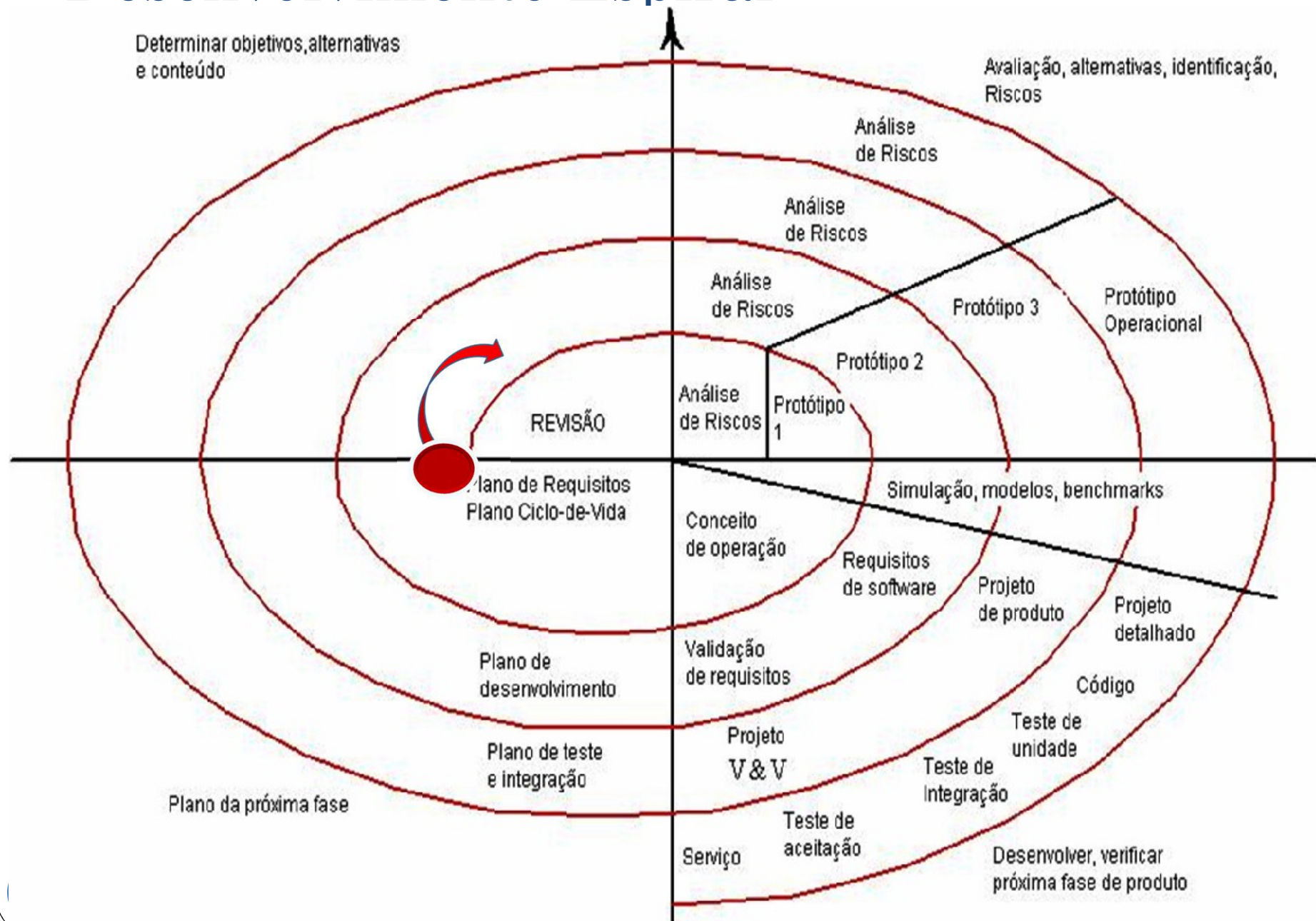
Desenvolvimento espiral



Desenvolvimento espiral

- ❑ Processo é representado como uma espiral ao invés de uma sequência de atividades com retorno;
- ❑ Cada volta na espiral representa uma fase no processo;
- ❑ Não existem fases fixas, as voltas na espiral são escolhidas de acordo com o que é requerido;
- ❑ Os riscos são explicitamente cotados e resolvidos durante todo o processo.

Desenvolvimento Espiral



Setores do modelo espiral

Estabelecimento de objetivos

- Identificados os objetivos, restrições e plano do projeto.

Avaliação e redução de riscos

- Os riscos são avaliados e atividades para reduzir os riscos principias

Desenvolvimento e validação

- Um modelo de desenvolvimento para o sistema é escolhido, podendo ser qualquer um dos modelos genéricos

Planejamento

- O projeto é revisado e a fase seguinte da espiral é planejada

Evolucionário: vantagens x desvantagens

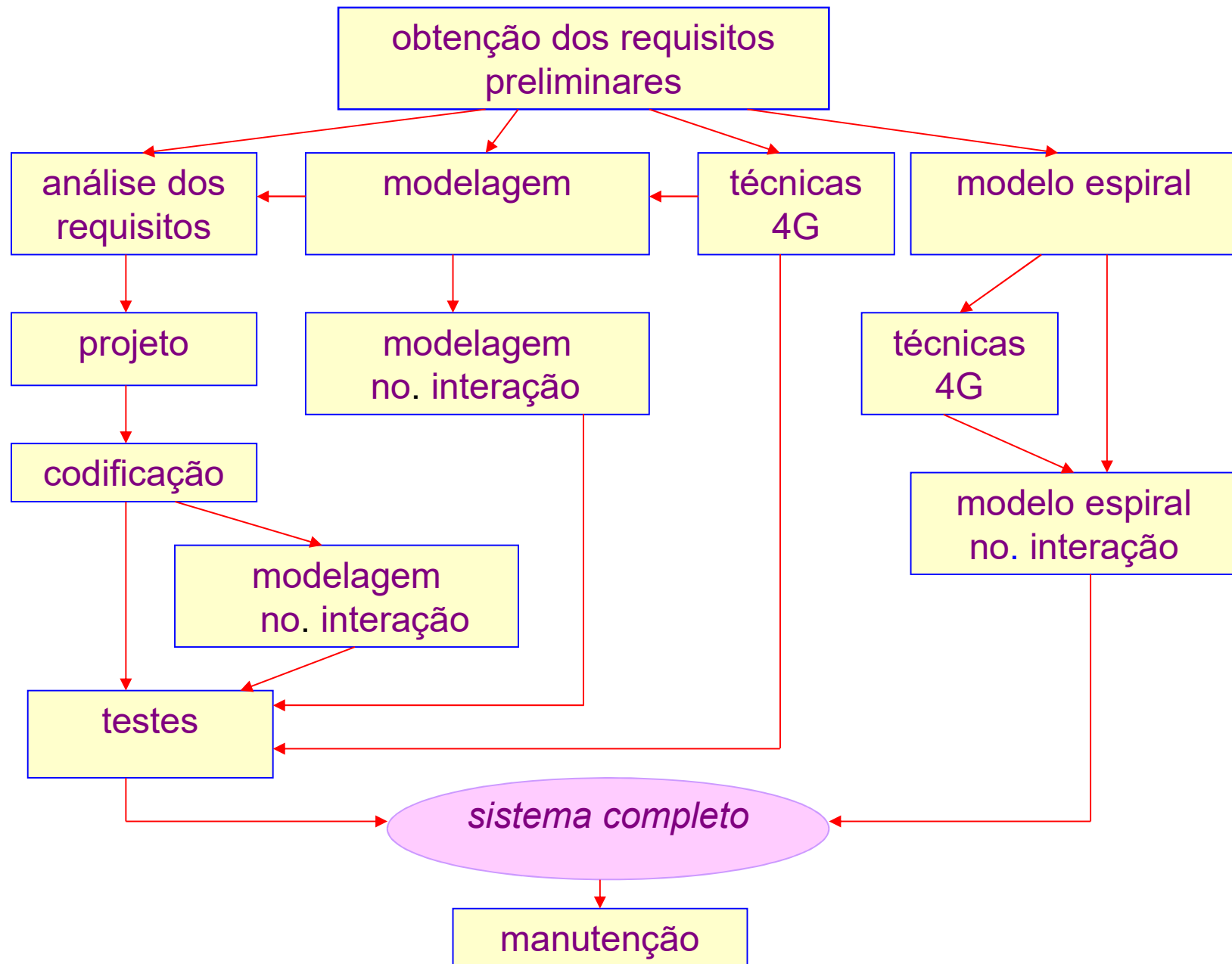
- **VANTAGENS:**

- É mais eficaz pelo o desenvolvimento em Cascata no atendimento imediato de clientes;
- A especificação pode ser desenvolvida de forma incremental.

- **PROBLEMAS:**

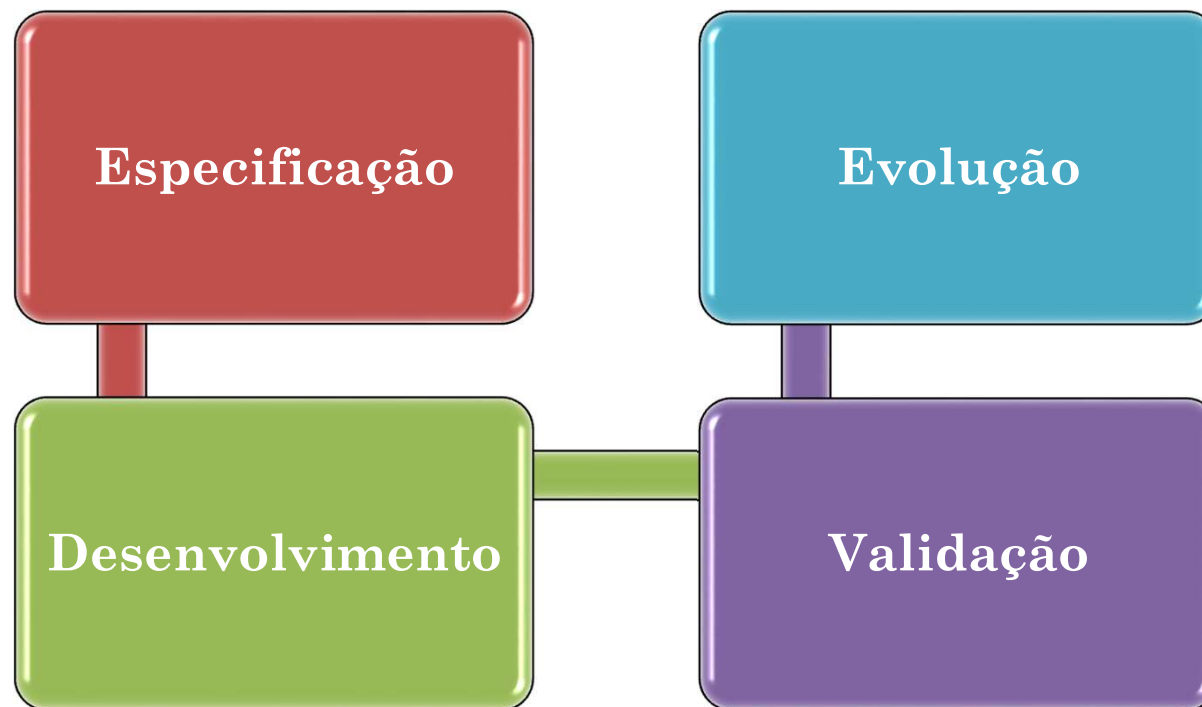
- O processo não é visível;
- Os sistemas são frequentemente mal estruturados;
- Tornam-se problemas graves para sistemas complexos e de longo ciclo de vida.

Combinação dos Métodos de Ciclo de Vida



Atividades Genérica de Processo

Atividades Genéricas de Processo



Especificação de Software (I)

ou Engenharia de Requisitos

É o processo para compreender requisitos, definir os serviços e identificar as restrições de operação e de desenvolvimento do sistema.

Conduz a:

- requisitos de alto nível para os clientes;
- especificação de sistema para projetistas;
- elaboração de documentos de requisitos.

Fases no Processo de Engenharia de Requisitos

Estudo da Viabilidade

Avaliação das necessidades dos usuários, os custos e decisões do projeto.



Elicitação e Análise de Requisitos

Observação e coletas de informação sobre os serviços do sistema.



Especificação de Requisitos

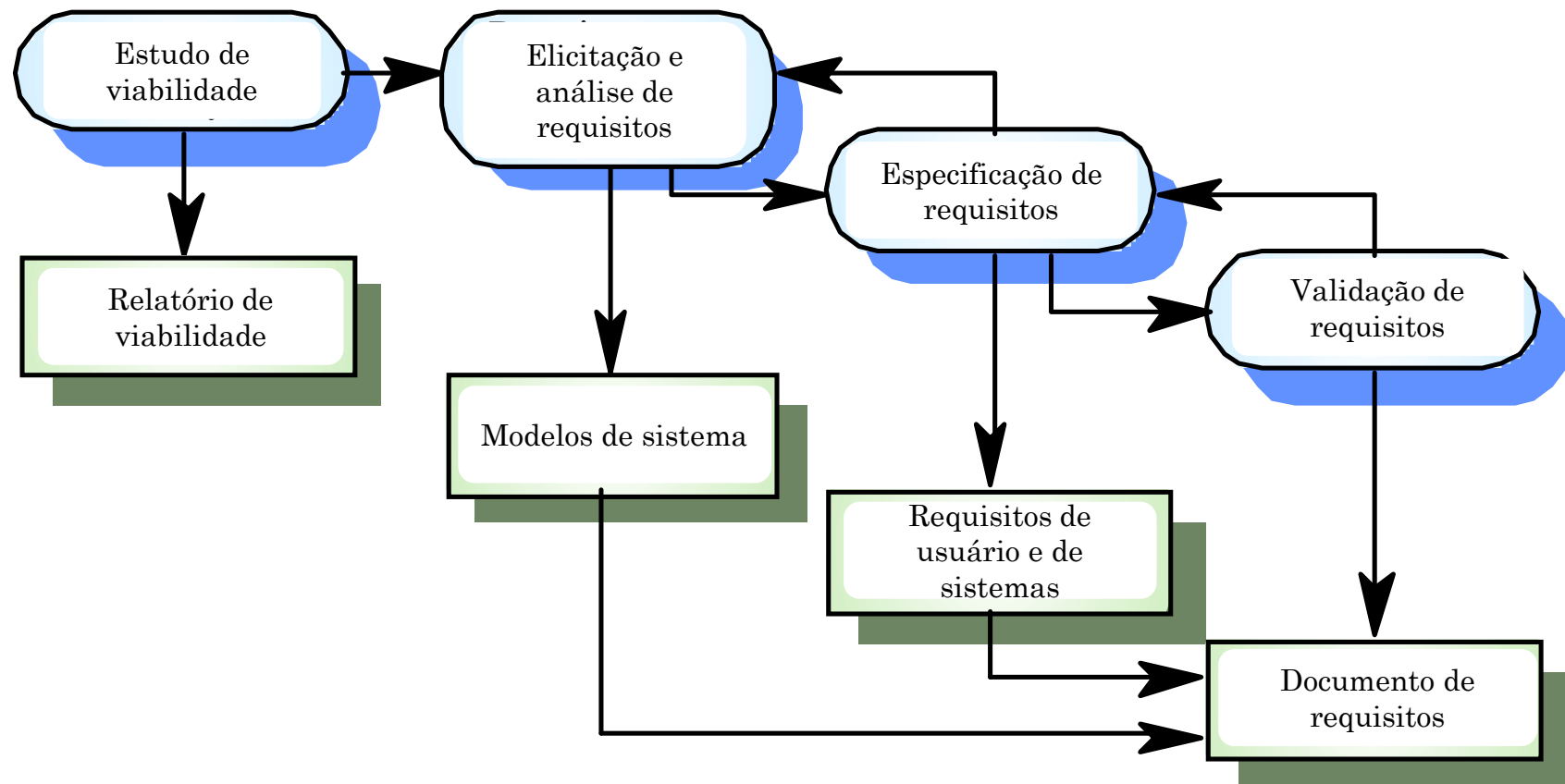
Descrever pormenor as informações coletadas e definir requisitos.



Validação de Requisitos

Verificar realismo, consistência e abrangência dos requisitos.

Processo de Engenharia de Requisitos



Implementação de Software (2)

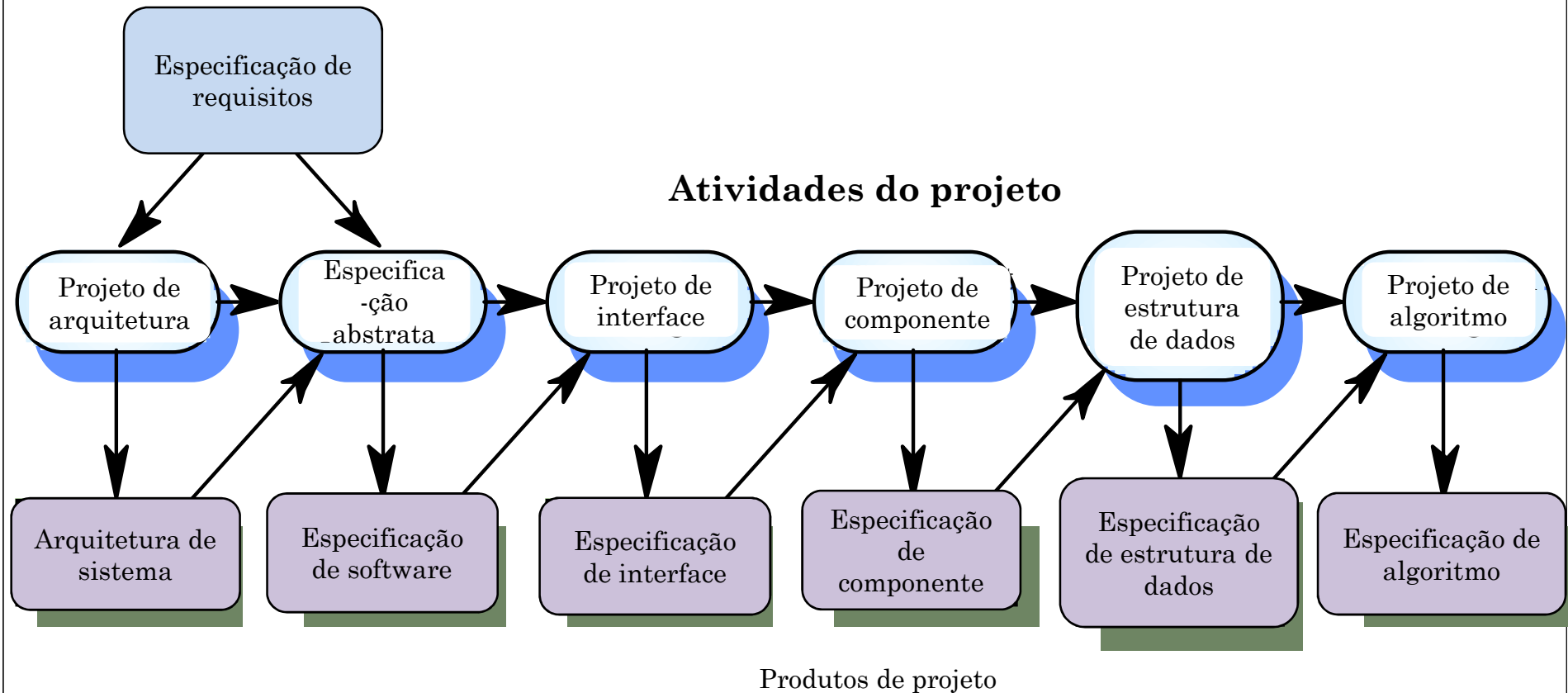
É o processo que transforma as especificações numa linguagem formal, segundo uma semântica e sintaxe, de entendimento dos computadores. (Presman, 2006).

Conversão de uma especificação de sistema em um sistema executável. (Sommerville, 2006)

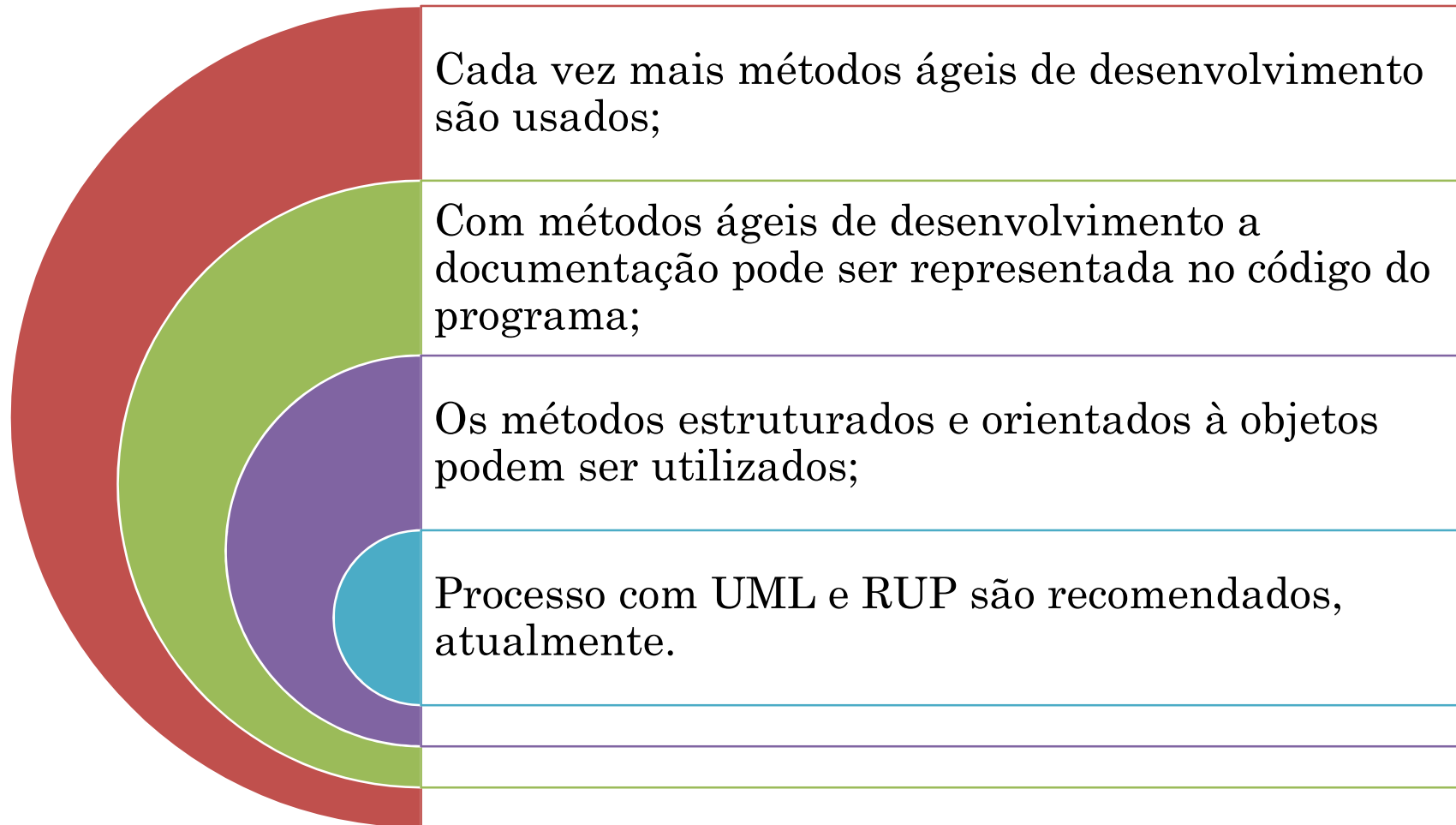
Fases do Desenvolvimento



Modelo do Processo



Considerações sobre Desenvolvimento



Exemplos de Modelos de Processo

Modelo de Objeto

Modelo de Sequência

Modelo de Transição

Modelo Estruturado

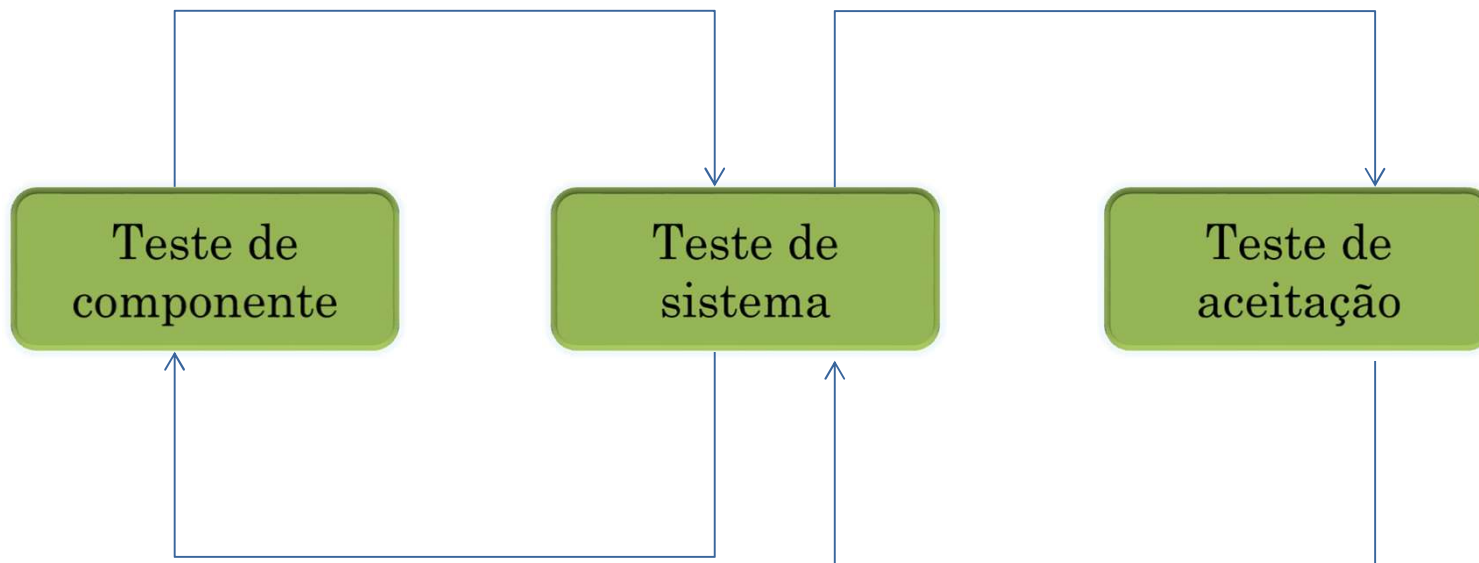
Modelo de Fluxo de Dados

Validação de Software (3)

Destina a mostrar que um sistema está em conformidade com sua especificação e que atende às expectativas do cliente.

Envolve verificações, inspeções e revisões a cada estágio do processo. (Sommerville, 2006)

Processo de Teste



Teste de Defeitos

A meta é descobrir defeitos em programas.

Um teste bem sucedido faz um programa se comportar de uma maneira anômala.

Os testes mostram a presença e não a ausência de defeitos.

Processo Básico de Teste

Teste de sistema

Teste de grupos de componentes integrados para criar um sistema ou um subsistema;

A responsabilidade é da equipe independente de teste;

Os testes são baseados em uma especificação de sistema.

Teste de componentes

Teste de componentes individuais de programa;

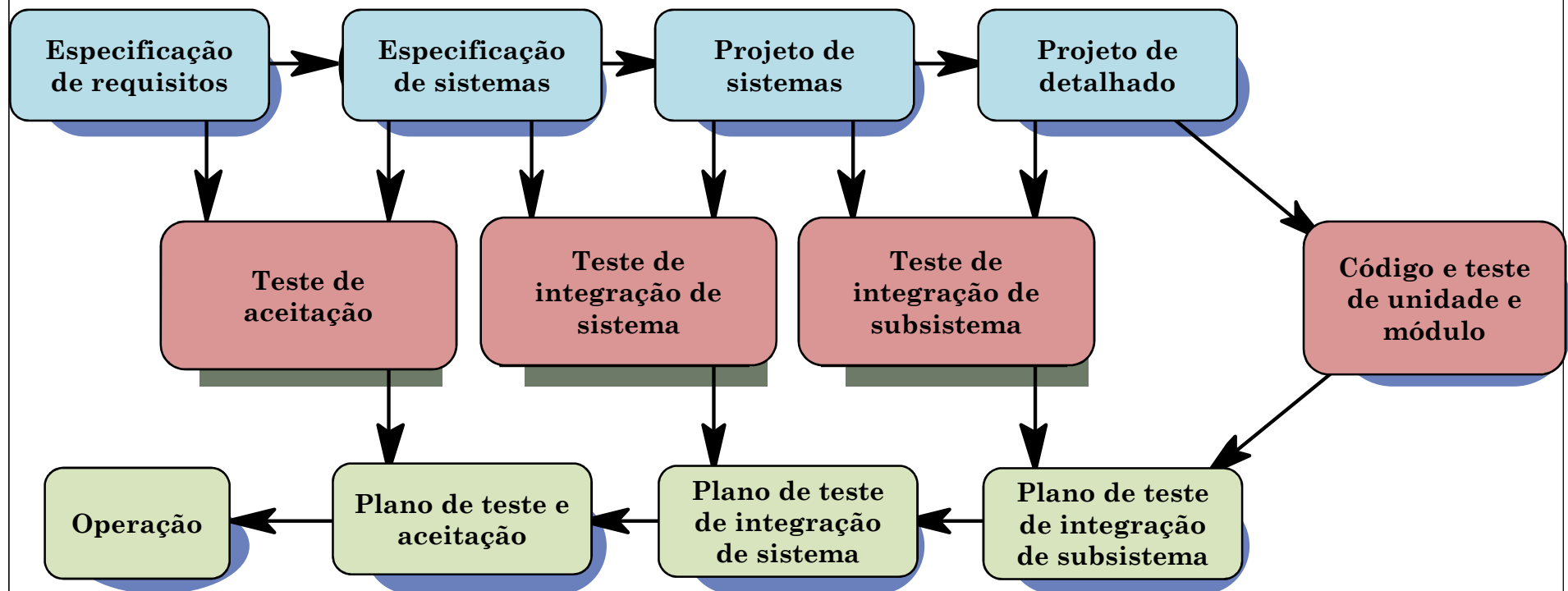
Responsabilidade do desenvolvedor do componente (exceto para sistemas críticos);

Os testes são derivados da experiência do desenvolvedor.

Teste de Aceitação

Teste final do processo, antes que o sistema seja aceito. Testado com dados fornecidos pelos clientes.

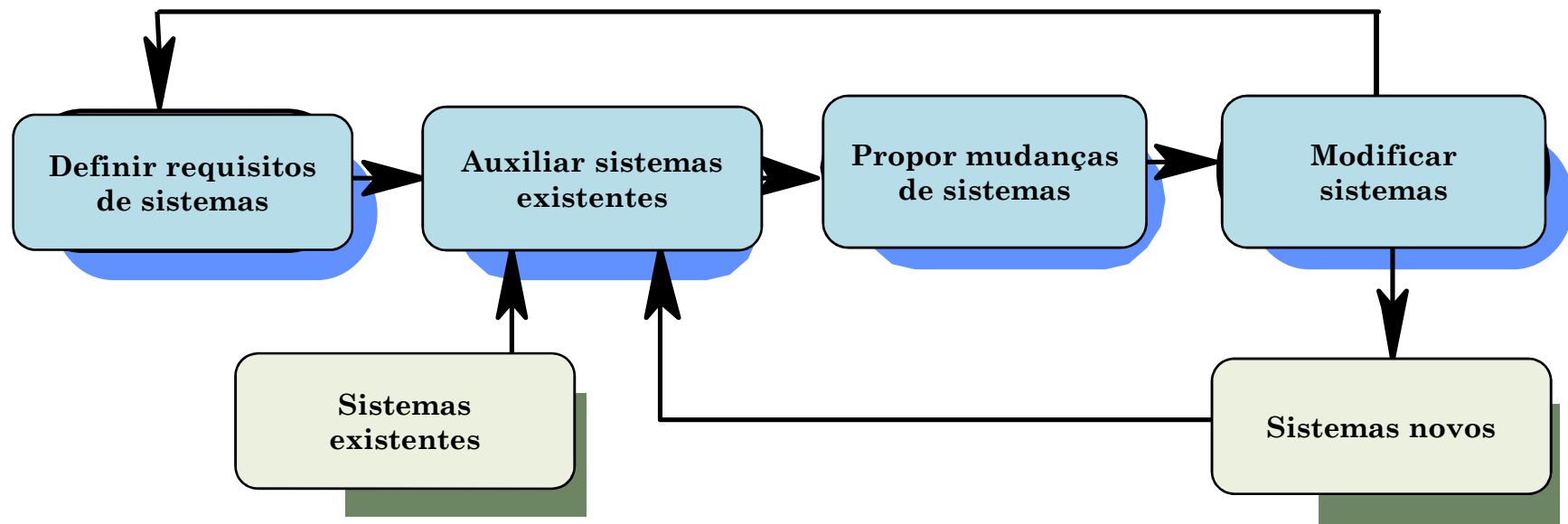
Fases do Processo de Teste



Validação de Software (4)

Consiste nas mudanças no software após implementação e entrega, como suporte às manutenções ou atendimentos as novas características tecnológicas e exigências do cliente.

Evolução de Sistema



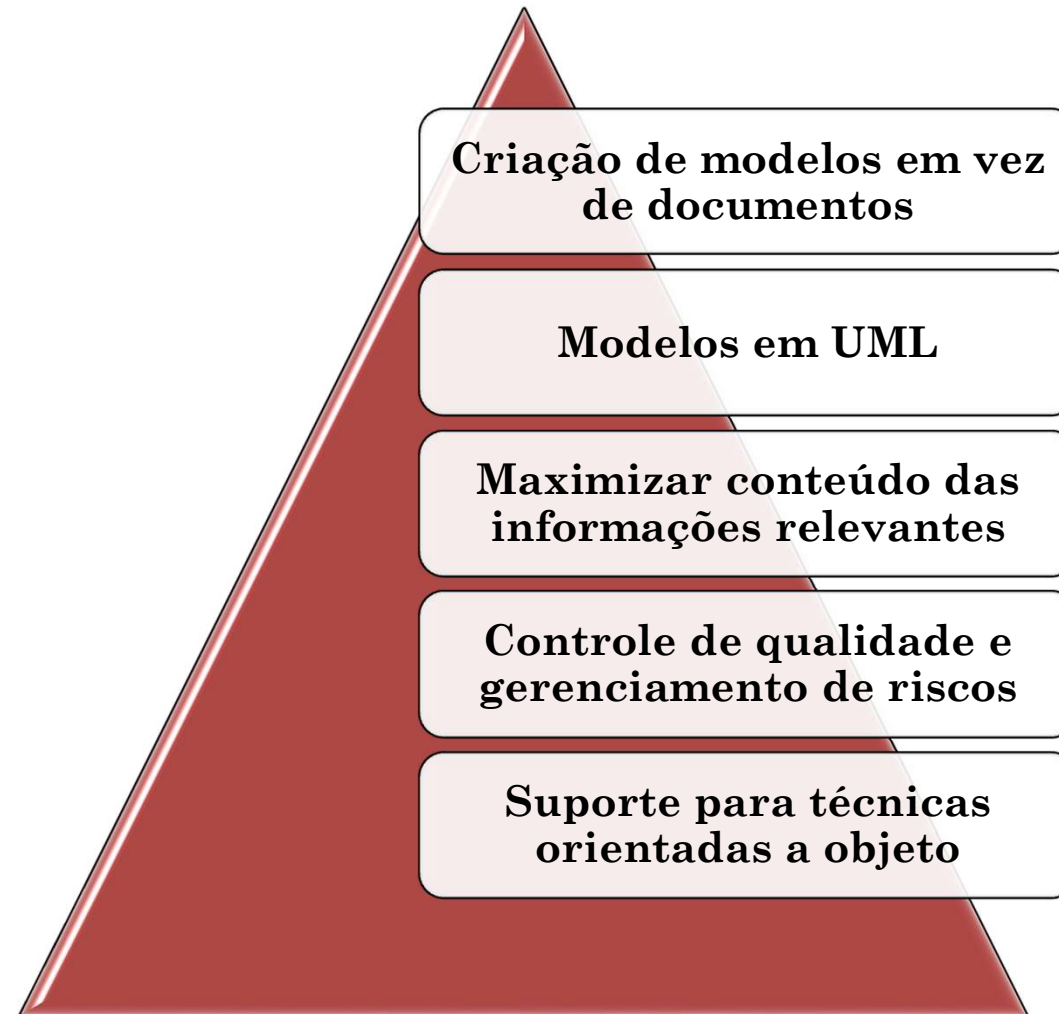
Desenvolvimento ágil - componentização

Rational Unified Process - RUP

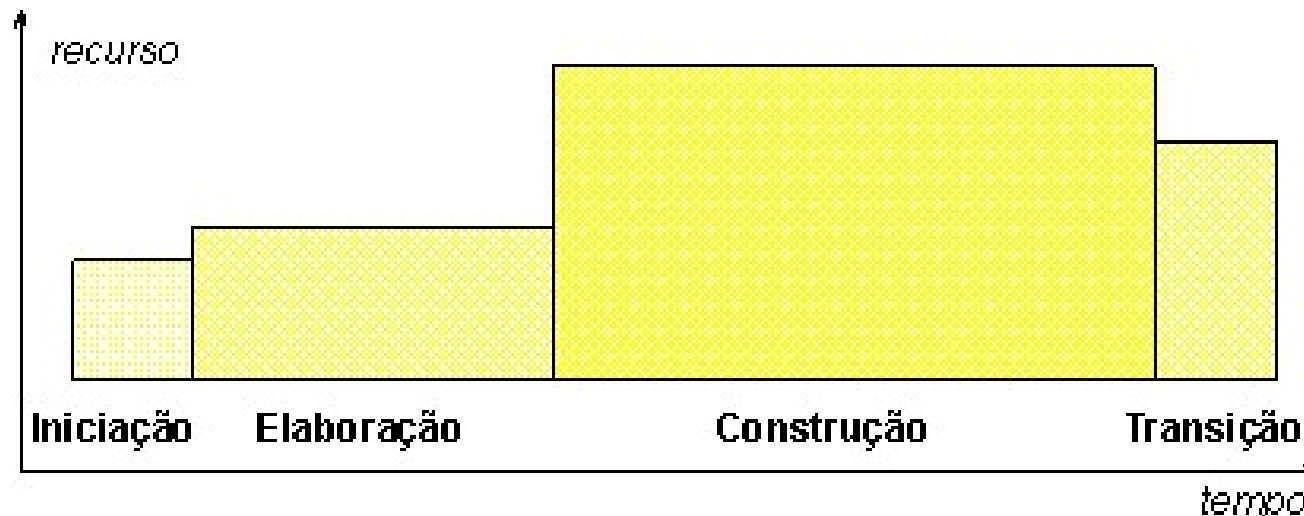
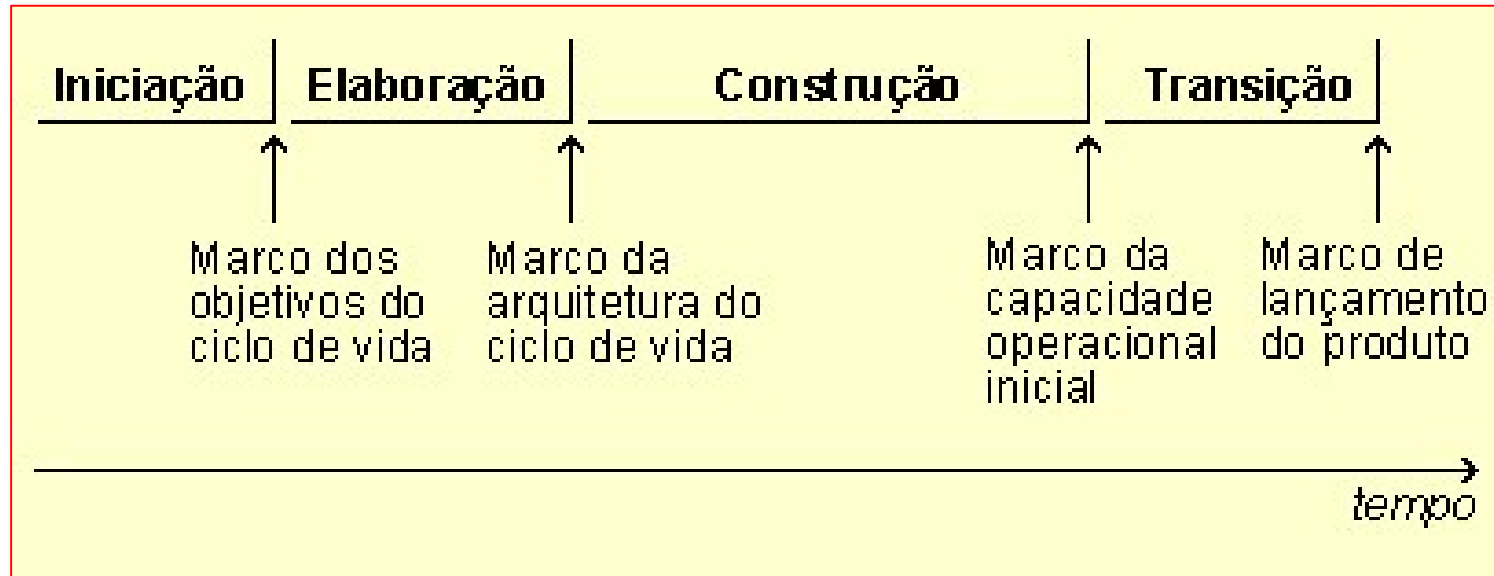
Rational Unified Process - RUP

É um conjunto de processo
que contempla várias
práticas atuais de
desenvolvimento de software,
buscando qualidade e pode
ser adaptada a uma ampla
diversidade de projetos.

Características do RUP



Fases Dinâmicas do RUP



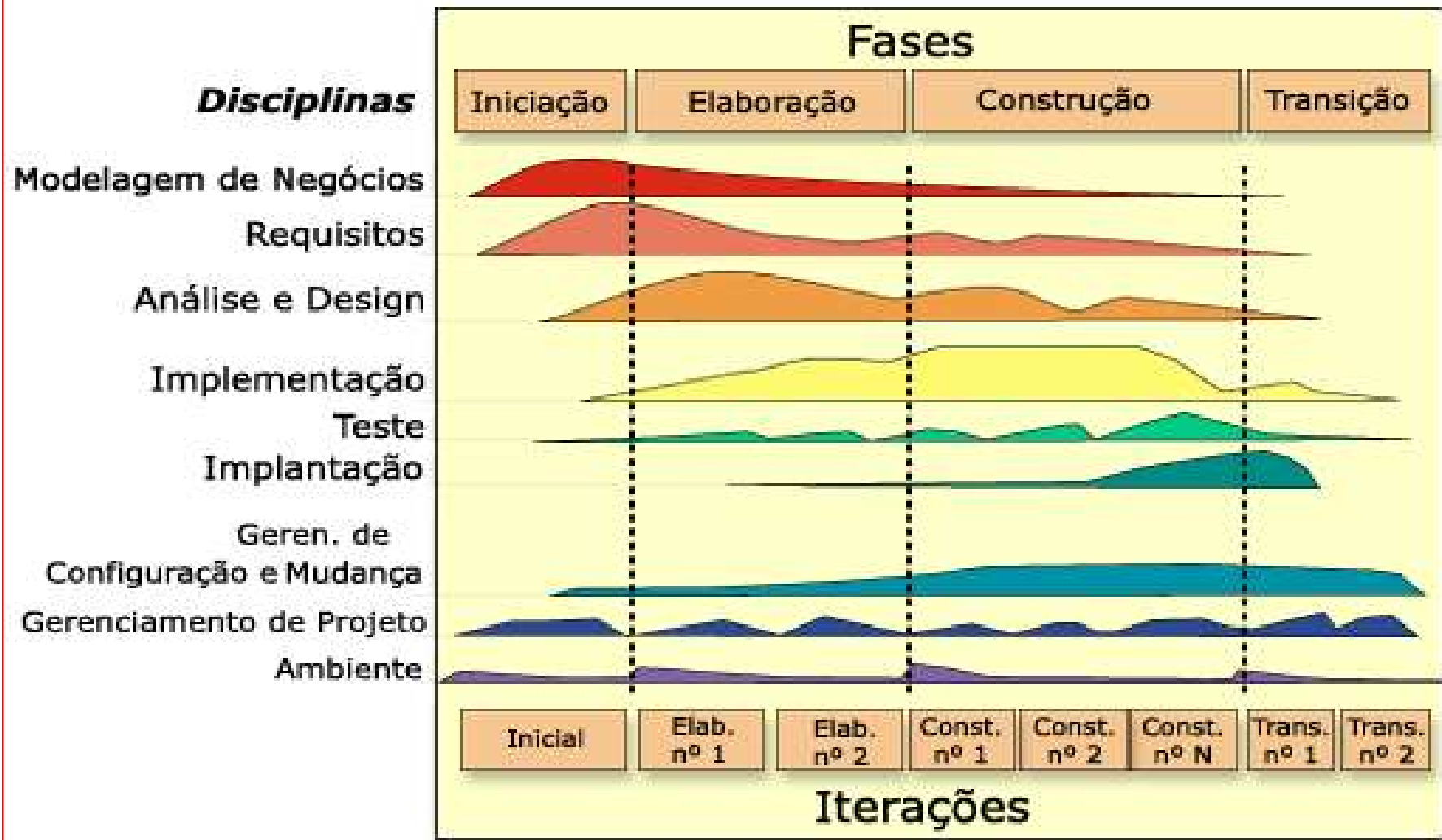
Workflows Estáticos do RUP

Workflow (atividades)	Descrição
Modelagem de negócios	Processos de negócios modelados com casos de uso
Requisitos	Identificação dos agentes e serviços para requisitos de sistemas
Análise de Projeto	Um projeto é criado e documentado usando modelos de arquitetura, componente, objeto, sequência
Implementação	Componentes são implementados. Pode-se usar geração automática de código.
Teste	Processo iterativo realizado em conjunto com a implementação.
Implantação	Uma versão do produto é criada e distribuída.
Gerenciamento de configuração e mudanças	Gerencia as mudanças do sistema.
Gerenciamento de projetos	Apoia gestão de projetos de processo.
Ambiente	Disponibilização de ferramentas e apoia equipe de desenvolvimento.

Perspectivas Práticas

- Desenvolver o software interativamente
- Gerenciar requisitos
- Usar arquitetura baseadas em componentes
- Modelar o software visualmente
- Verificar a qualidade do software
- Controlar as mudanças do software

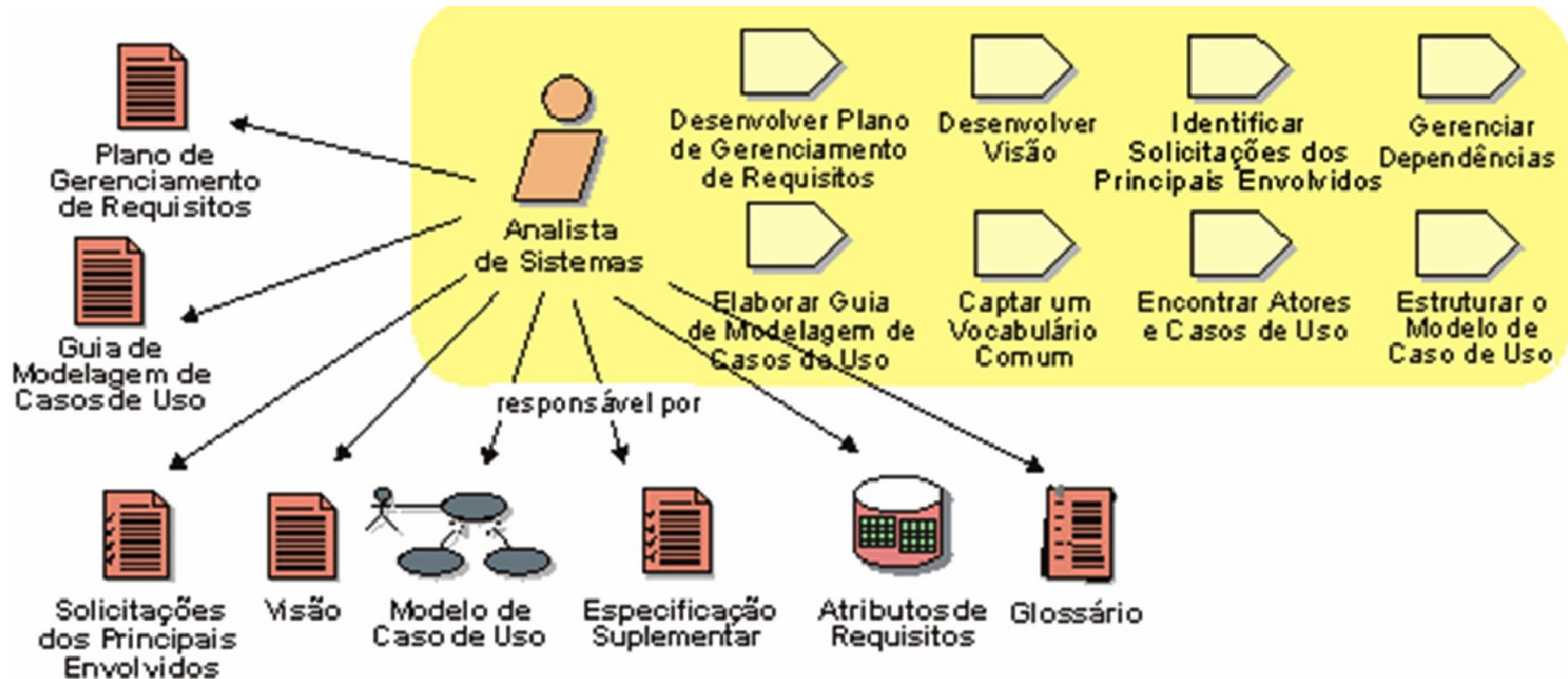
RUP



O **eixo horizontal** representa o tempo e os aspectos do ciclo de vida do processo

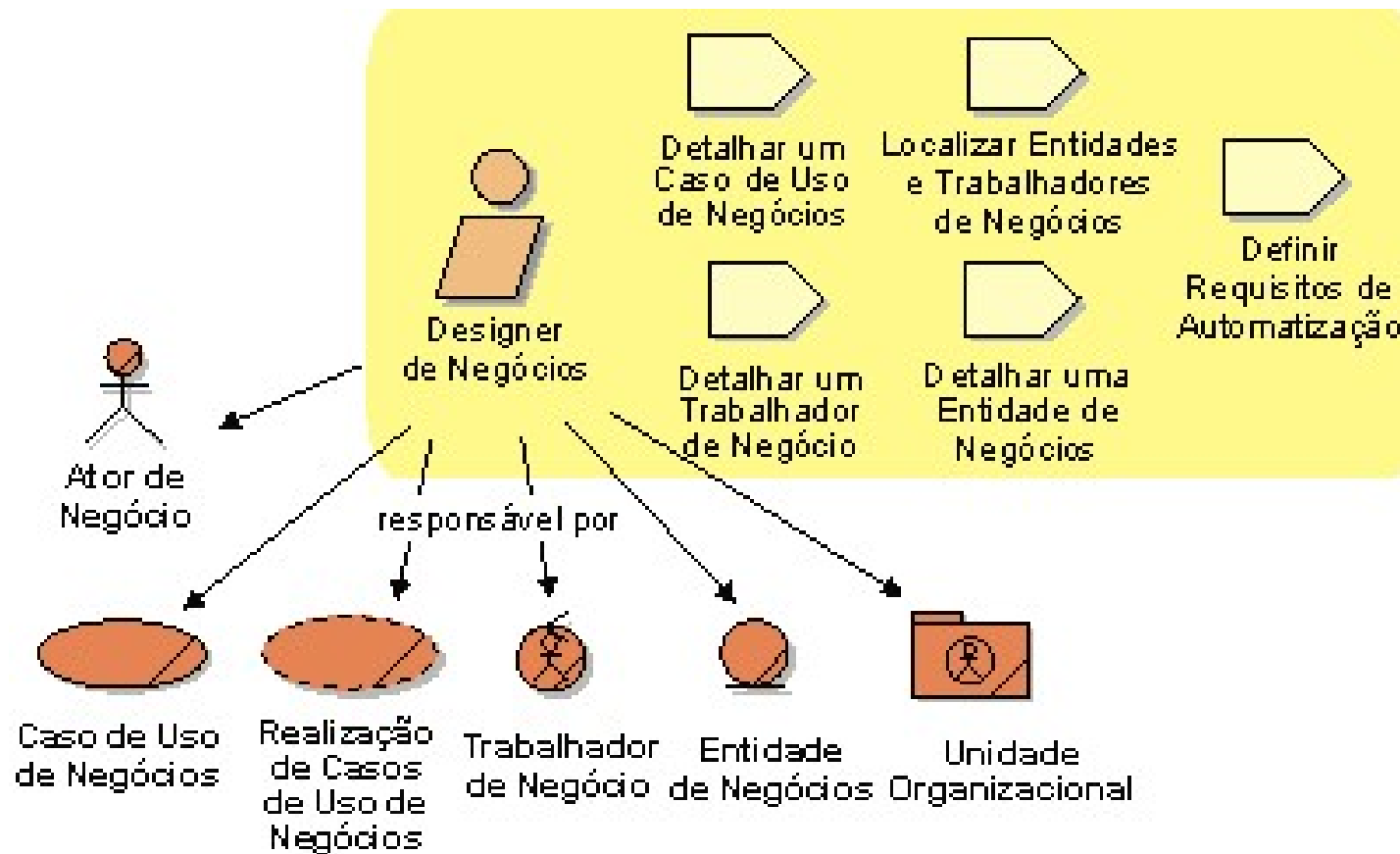
O **eixo vertical** representa as disciplinas com as atividades de maneira lógica

Papel do Analista de Sistemas



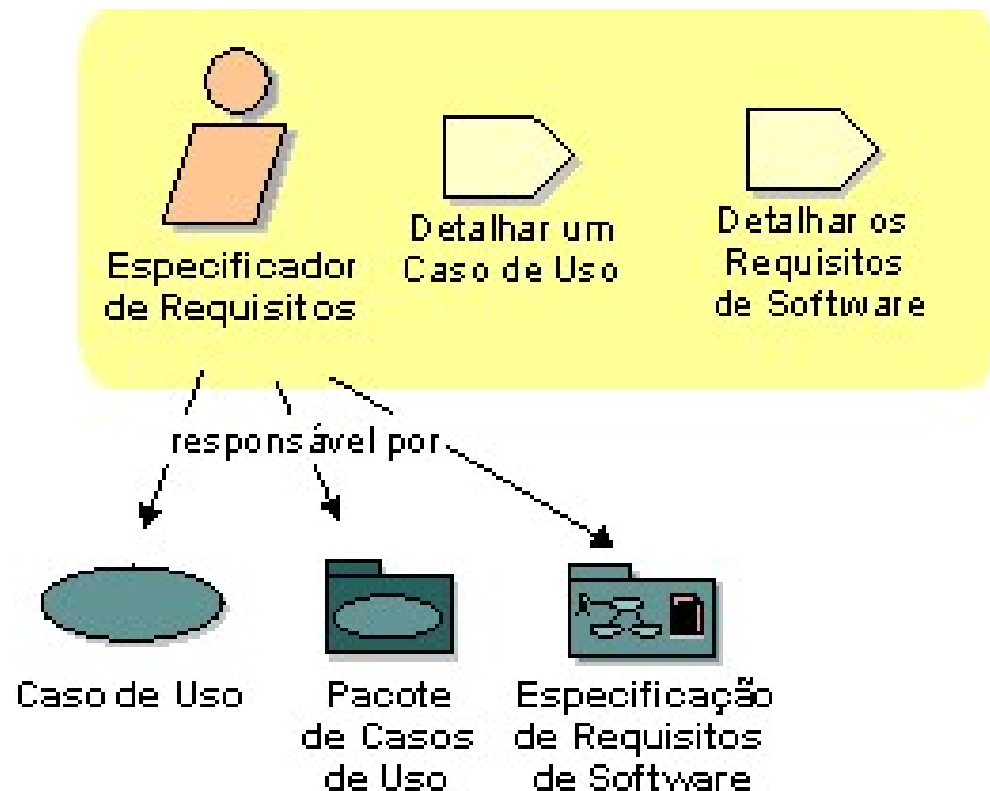
coordena a identificação de requisitos e a modelagem de casos de uso, delimitando o sistema e definindo sua funcionalidade

Designer de Negócios



Entender as necessidades do cliente e do usuário, suas estratégias e metas.
Facilitar a modelagem da organização-alvo.

Especificador de Requisitos



detalha a especificação de uma parte da funcionalidade do sistema, descrevendo o aspecto Requisitos de um ou de vários casos de uso e outros requisitos de software de apoio.

Computer Aided Software Engineering - CASE

CASE

Processo para auxiliar na engenharia de software, projeto, desenvolvimento de programas e teste, através de ferramentas para automação de algumas atividades.

Algumas Atividades Suportadas

Desenvolvimento de modelos gráficos

Dicionário de dados com informações sobre entidades e relacionamentos

Geração de interfaces com o usuário

Debuggin de programas

Processo de teste, documentação, reengenharia

Ferramentas CASE

Tipo de Ferramenta
Ferramentas de planejamento
Ferramentas de edição
Ferramentas de gerenciamento de mudanças
Ferramentas de gerenciamento de configuração
Ferramentas de prototipação
Ferramentas de apoio a métodos
Ferramentas de processamento de linguagens
Ferramentas de análise de programa
Ferramentas de teste
Ferramentas de depuração
Ferramentas de documentação
Ferramentas de reengenharia

Ferramentas, Workbenches, Ambientes



Referências

- PAULA, Wilson Pádua. Engenharia de Software. 2a. Ed. São Paulo, LTC, 2005.
- PRESSMAN, Roger S. Engenharia de Software. 6a. Ed. São Paulo: Makron Books, 2006.
- SOMMERVILLE, Ian. Engenharia de Software. 8a. Ed., São Paulo: Addison Wesley, 2006.