

Processos e Modelos de Processos de Software

- Natureza, Categorias, Atividades...

O Processo de Software

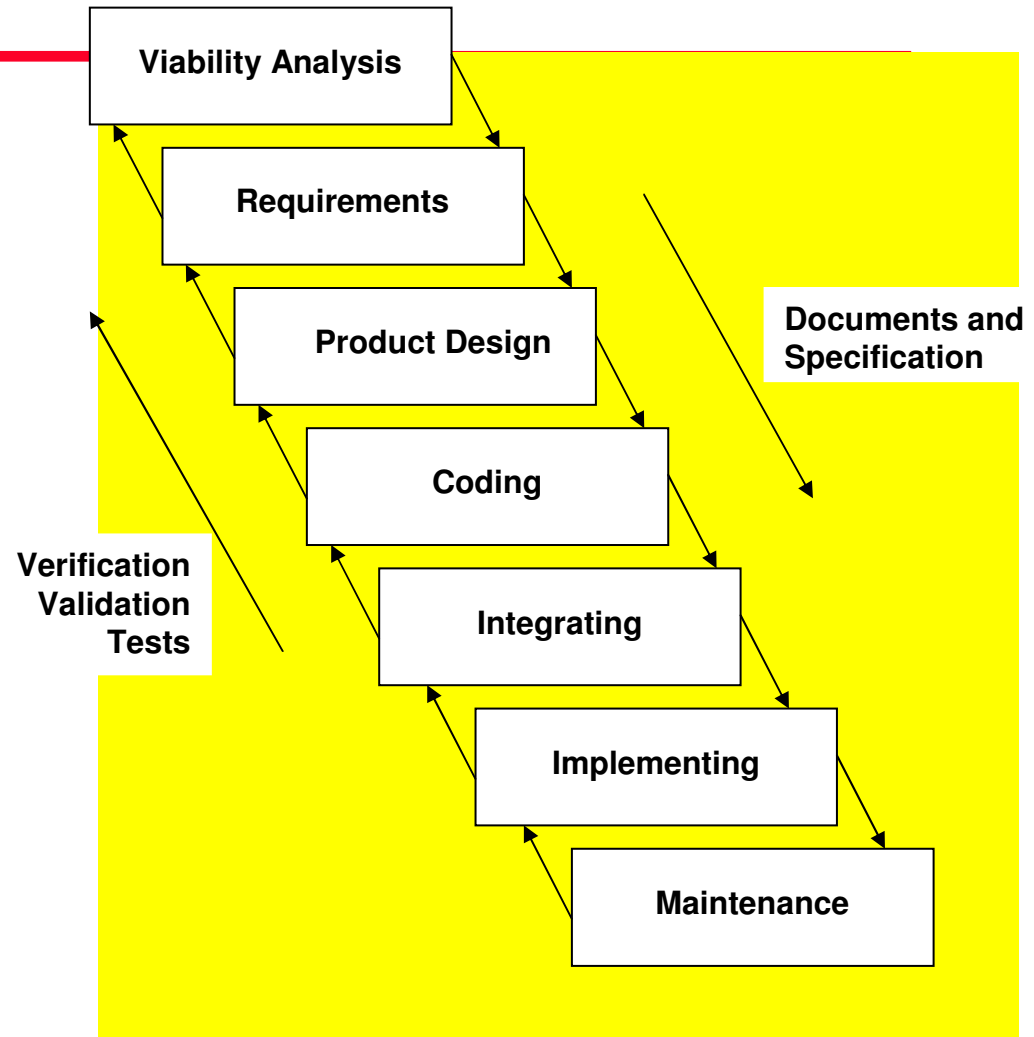
- Conjunto coerente de atividades requeridas para desenvolver um sistema de software
 - Especificação
 - Design/Projeto
 - Validação
 - Evolução
- O **Modelo de Processo de Software** é uma representação abstrata de um processo
 - descrição de um processo de uma perspectiva particular

Categorias de Desenvolvimento de Software

- Desenvolvimento clássico/genérico (ex. Modelo Cascata, o Modelo Espiral, outros)
 - Especificação e desenvolvimento são fases distintas e separadas
- Desenvolvimento Evolucionário
 - Especificação e desenvolvimento são interligadas
- Desenvolvimento Formal
 - Um modelo matemático de sistema é formalmente transformado até a implementação

O Modelo Waterfall (Cascata)

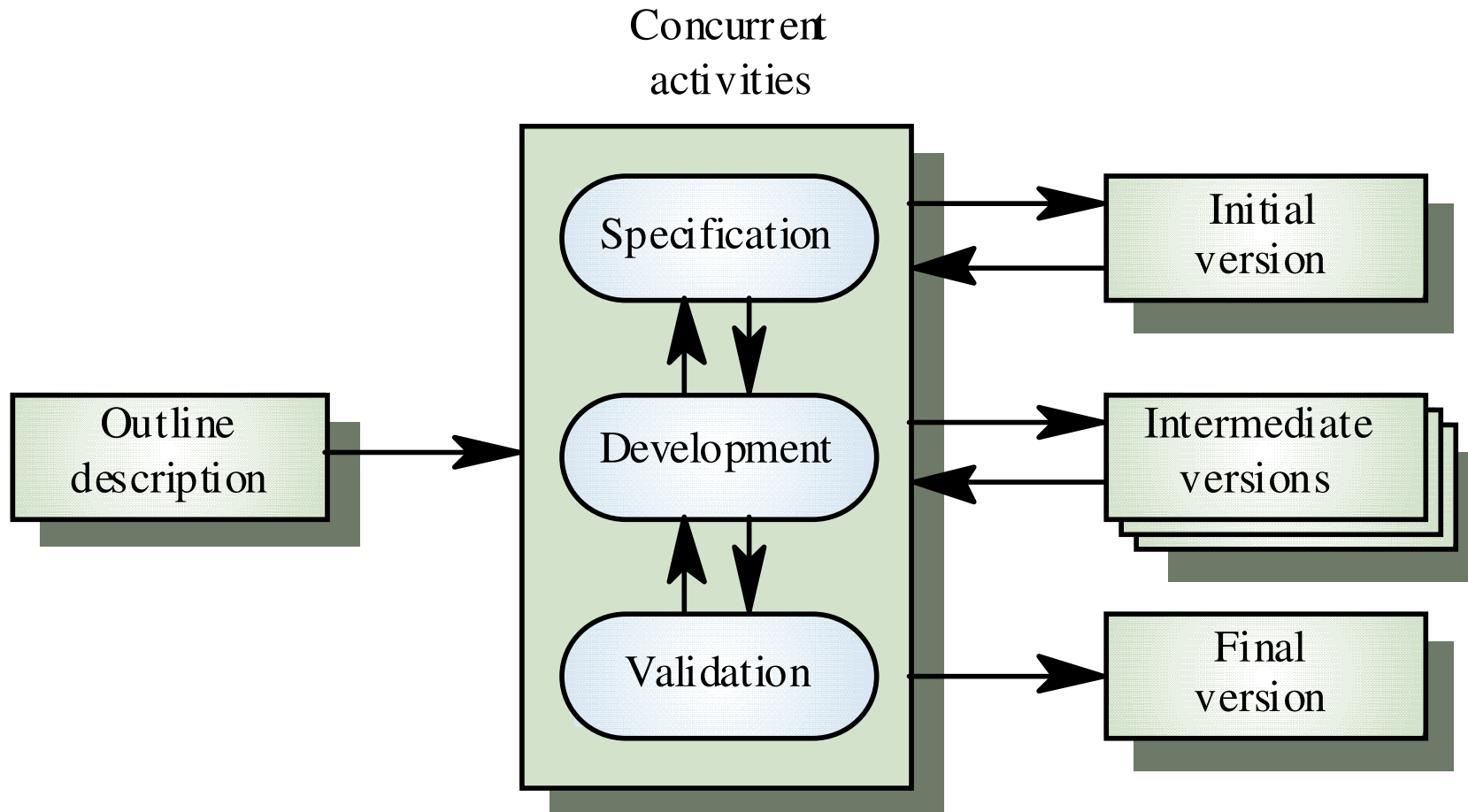
- Relaciona a forma e função do sistema à estrutura do processo
- É difícil entender e expressar os requisitos antecipadamente



Desenvolvimento Evolucionário

- Desenvolvimento Exploratório
 - Objetivo é trabalhar com clientes e evoluir um sistema final a partir de uma especificação geral inicial.
 - Deve começar com **requisitos bem entendidos**
- Prototipação descartável
 - Objetivo é entender os requisitos do sistema
 - Deve começar com **requisitos ainda pouco entendidos**

Desenvolvimento Evolucionário



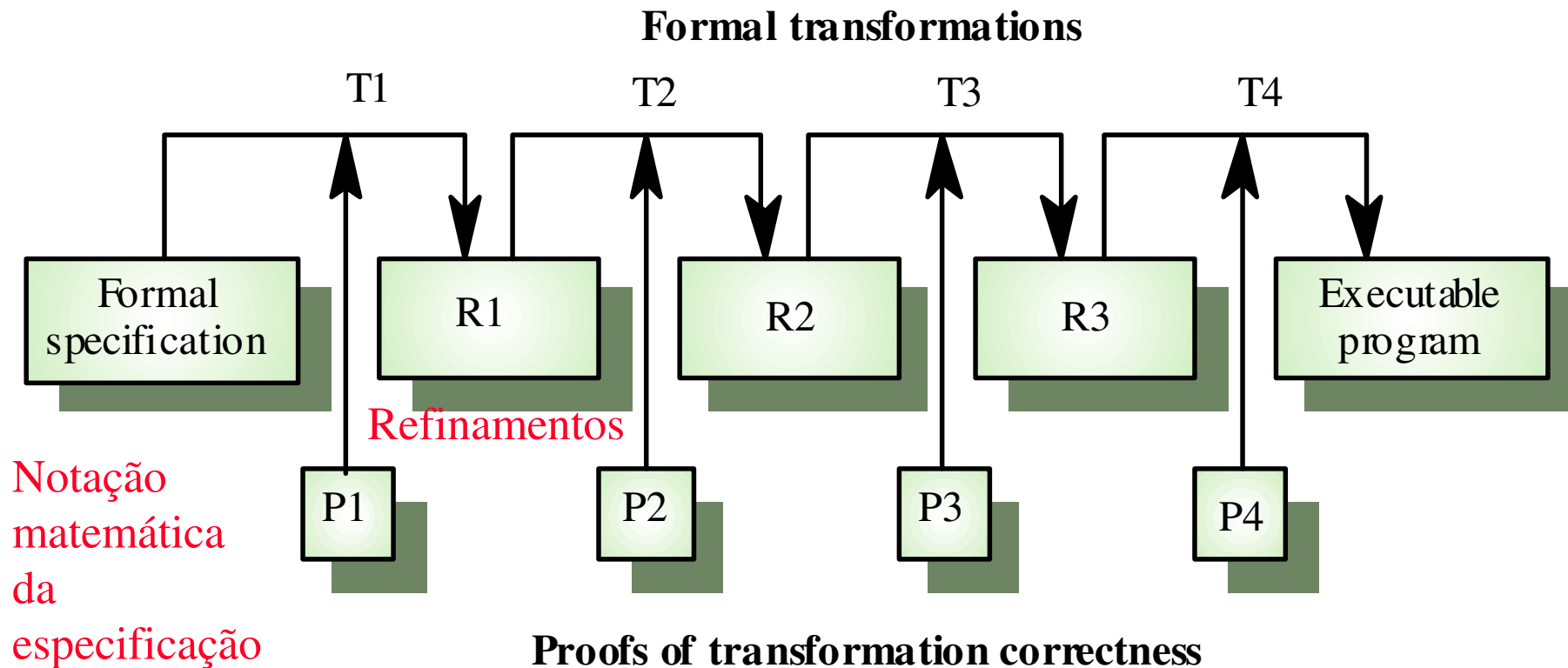
Desenvolvimento Evolucionário

- Problemas
 - Falta de visibilidade do processo
 - Sistemas pouco estruturados
 - Habilidades especiais (ex. Linguagens para prototipação rápida) podem ser requeridas
- Aplicabilidade
 - Sistemas interativos de pequeno a médio porte
 - Partes de grandes sistemas (ex. a interface de usuário)
 - Sistemas de vida curta

Desenvolvimento Formal de Sistemas

- Baseado na transformação de uma especificação matemática através de representações diferentes até chegar ao programa executável
- Transformações preservam a corretude ('correctness-preserving')
 - é direto mostrar que o programa está em conformidade com sua especificação
- Embutido na abordagem 'Cleanroom' do desenvolvimento de software

Transformações Formais



Desenvolvimento Formal de sistemas

- Problemas
 - Habilidade e treino para aplicar a técnica
 - Dificuldade para especificar formalmente alguns aspectos do sistema como por ex. a interface de usuário
- Aplicabilidade
 - Sistemas Críticos
 - » aqueles onde a segurança deve ser feita antes do sistema ser colocado em operação

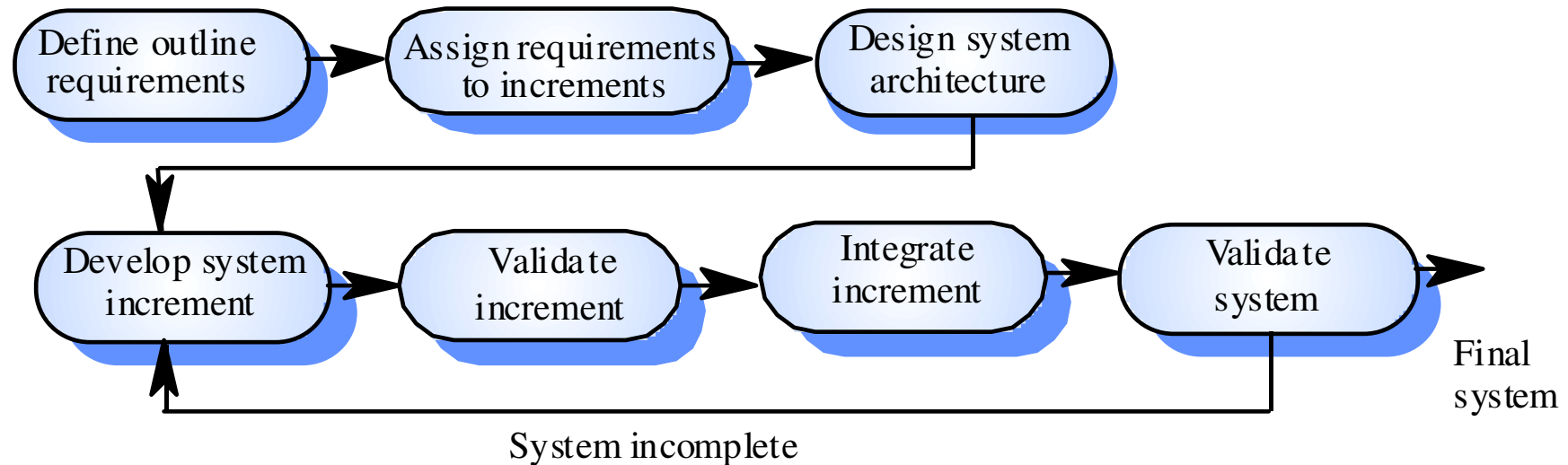
Iteração de Processo

- Requisitos do sistema SEMPRE evoluem no curso do projeto
 - Iteração do processo onde fases anteriores são retrabalhadas é sempre parte do processo para grandes sistemas
- Iteração pode ser aplicada para qualquer modelo genérico de processo
- Duas abordagens (relacionadas)
 - Desenvolvimento incremental
 - Desenvolvimento espiral

Desenvolvimento Incremental

- Em vez de entregar o sistema “de uma só vez”, o desenvolvimento e a entrega são quebrados em incrementos
 - cada incremento entrega parte da funcionalidade requerida
- Requisitos do usuário são priorizados
 - requisitos de mais alta prioridade são incluídos nos incrementos iniciais
- Uma vez iniciado o desenvolvimento de um incremento, os requisitos são congelados
 - embora requisitos para incrementos posteriores possam continuar a evoluir

Desenvolvimento Incremental



Ex. Extreme Programming

- Nova abordagem ao desenvolvimento baseada no desenvolvimento e entrega de incrementos pequenos da funcionalidade
- Depende de melhorias constantes no código, envolvimento do usuário no grupo de desenvolvimento e programação por pares (codifica/testa)

Vantagens do Desenvolvimento Incremental

- Funcionalidade do sistema fica disponível mais cedo ao cliente
- Incrementos iniciais funcionam como protótipos para ajudar a elicitare requisitos para incrementos subsequentes
- Baixo risco de falha geral do projeto
- Os serviços de mais alta prioridade tendem a passar por mais testes

Desenvolvimento espiral

- O processo é representado como uma espiral ao invés de uma seqüência de atividades com realimentação.
- Cada *loop* na espiral representa uma fase no processo.
- Sem fases definidas, tais como especificação ou projeto – os *loops* na espiral são escolhidos dependendo do que é requisitado.
- Os riscos são explicitamente avaliados e resolvidos ao longo do processo.

Modelo espiral do processo de software

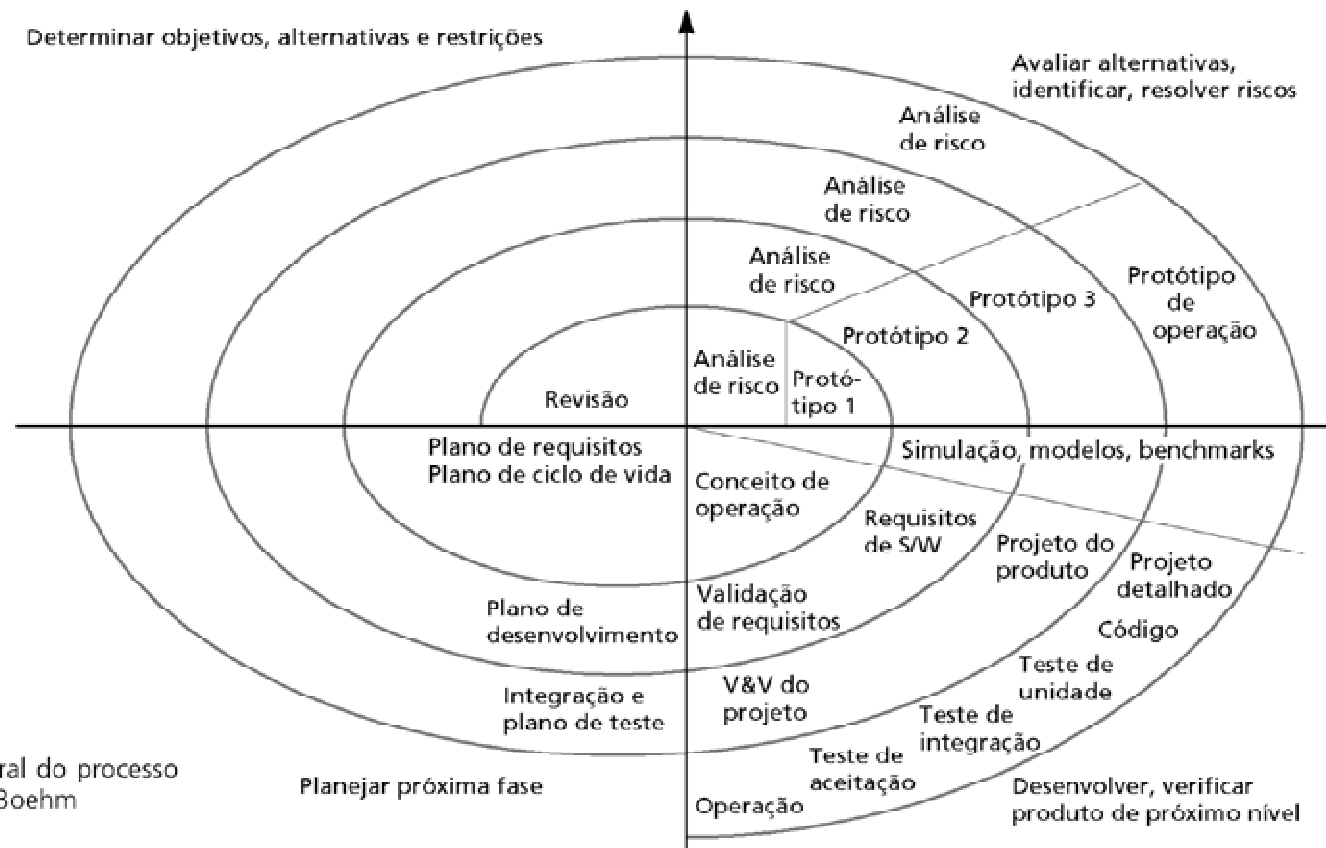


Figura 4.5

Modelo em espiral do processo de software de Boehm (©IEEE, 1988).

Setores do modelo espiral

- Definição de objetivos
 - Objetivos específicos para a fase são identificados.
- Avaliação e redução de riscos
 - Riscos são avaliados e atividades são realizadas para reduzir os riscos-chave.
- Desenvolvimento e validação
 - Um modelo de desenvolvimento para o sistema, que pode ser qualquer um dos modelos genéricos, é escolhido.
- Planejamento
 - O projeto é revisado e a próxima fase da espiral é planejada.

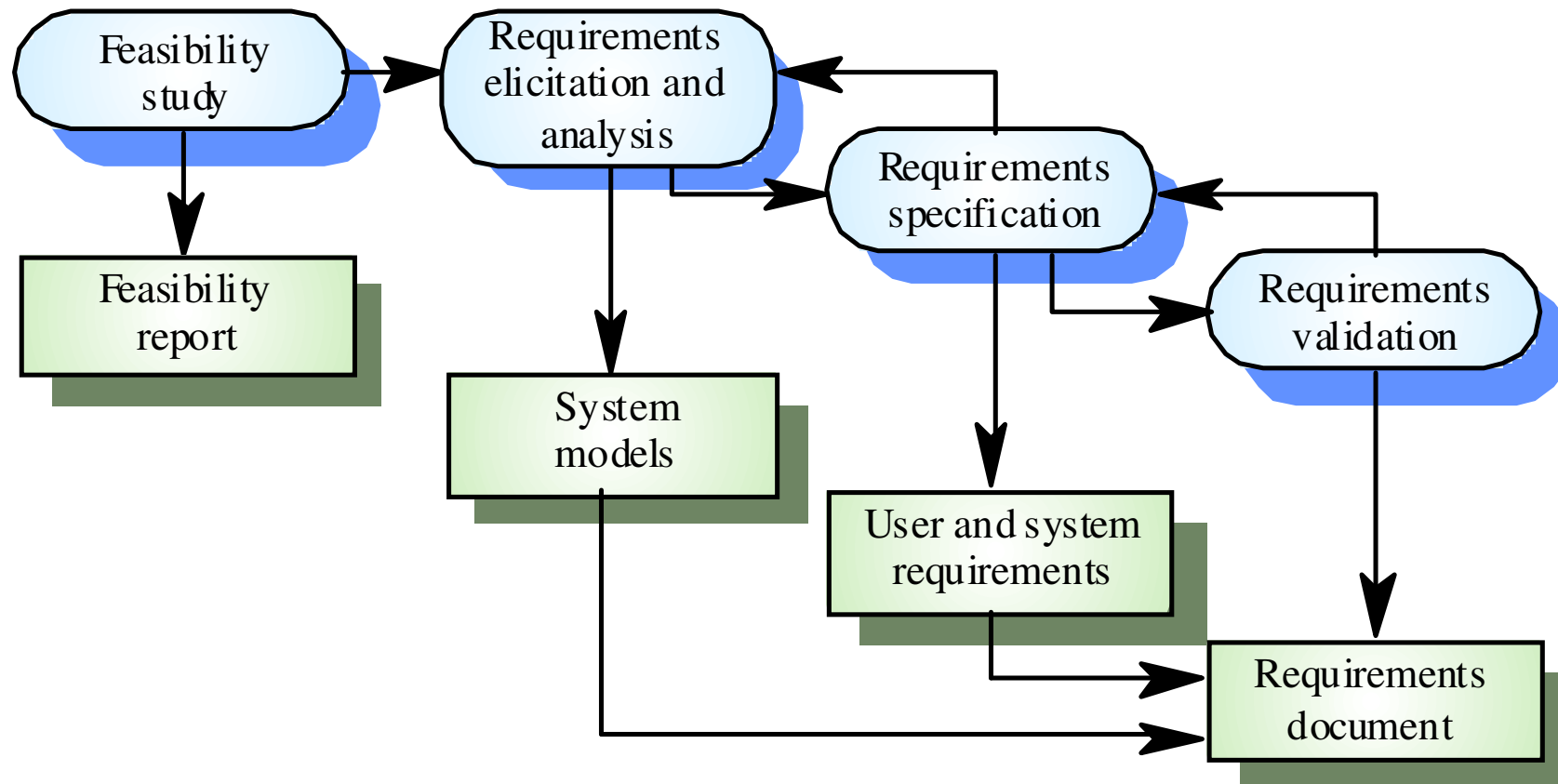
Atividades de processo

- Especificação de software
- Projeto e implementação de software
- Validação de software
- Evolução de software

Especificação de Software

- Processo de estabelecer que serviços são requeridos e as limitações na operação e desenvolvimento do sistema
- Processo de Engenharia de Requisitos
 - Estudo de viabilidade
 - Elicitação e Análise de Requisitos
 - Especificação de Requisitos
 - Validação de Requisitos

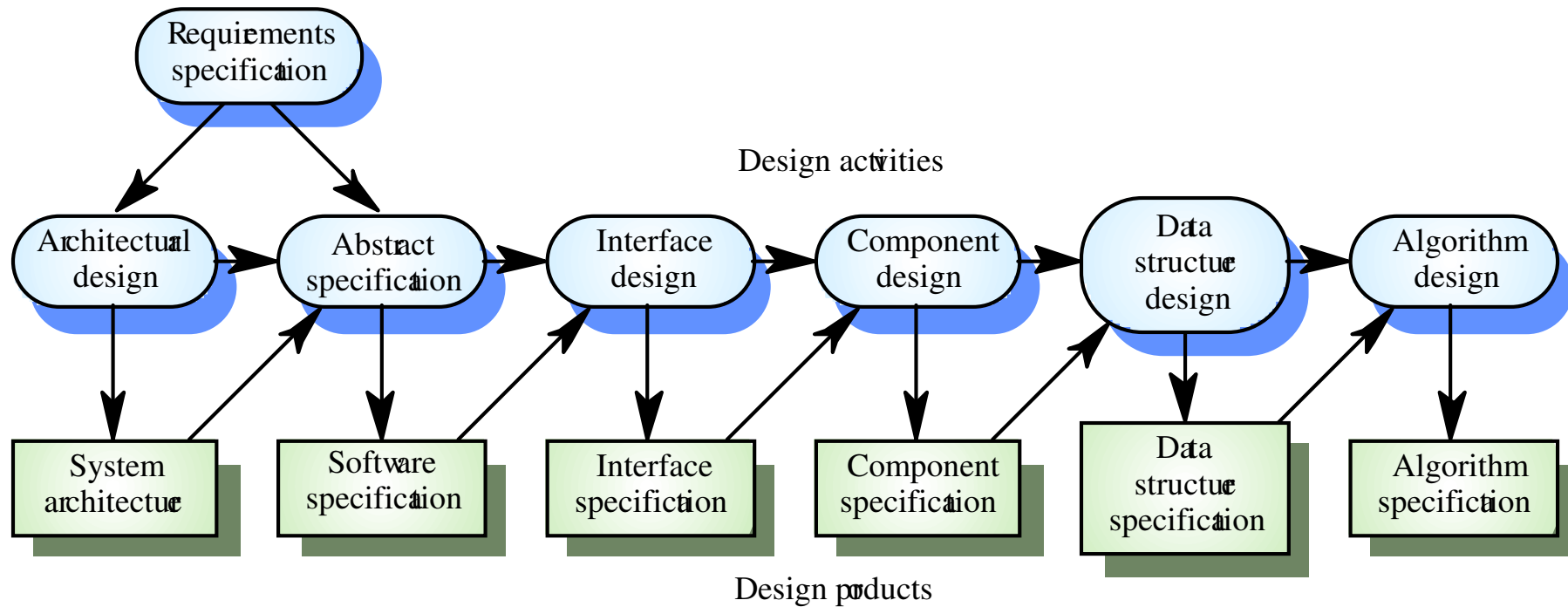
Um Processo (geral) de Engenharia de Requisitos



Design e implementação de Software

- Processo de converter a especificação do sistema em um sistema executável
- Design/Projeto do Software
 - Estrutura que representa a especificação
- Implementação
 - Traduz essa estrutura em um programa executável
- Atividades de design e implementação são inter-relacionadas

Atividades e produtos de um processo de design de software



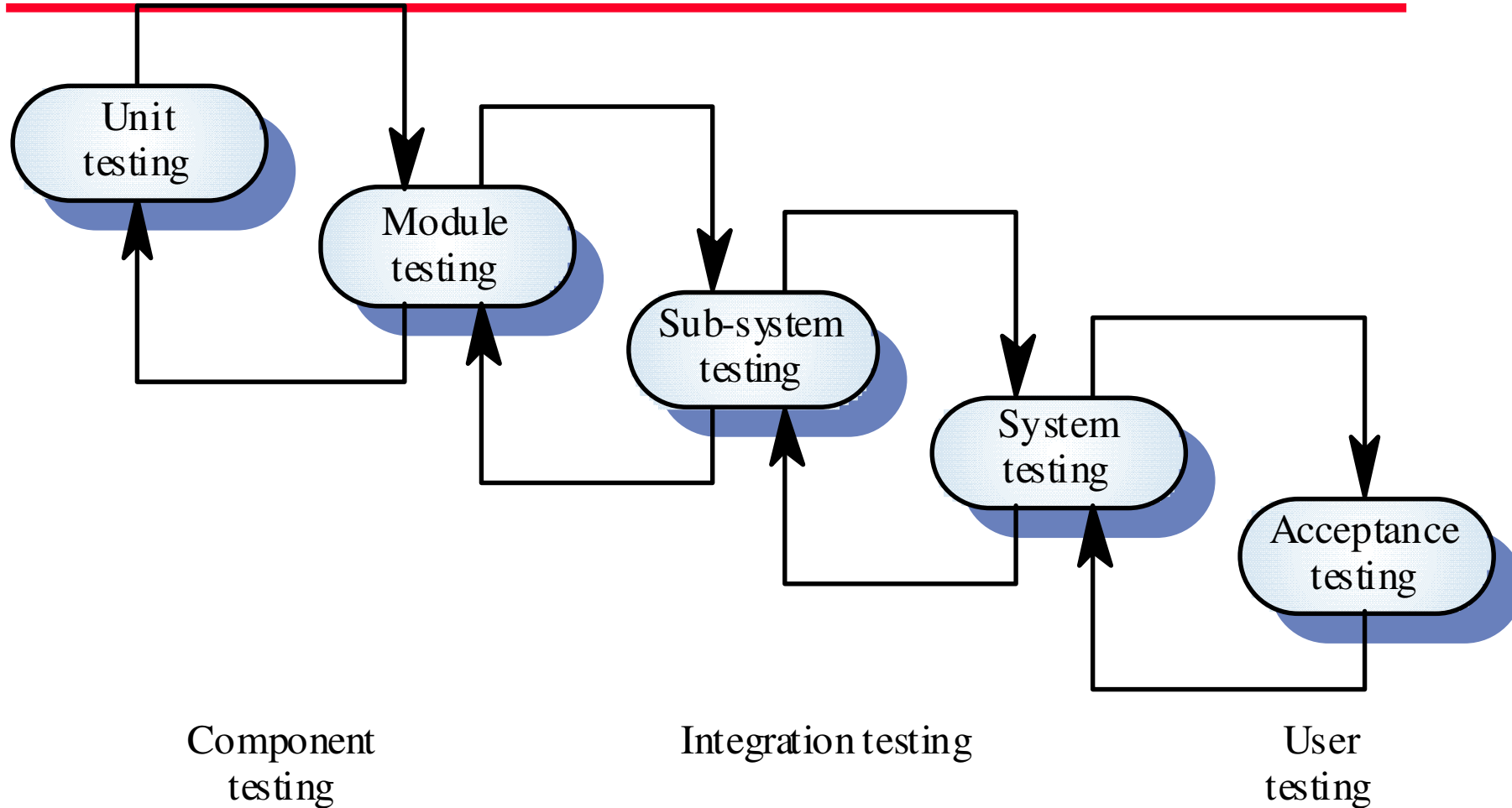
Métodos de Design

- Abordagens sistemáticas para o design de software
- O projeto/design é geralmente **documentado** como um conjunto de **modelos gráficos**
- Ex. de Modelos :
 - Modelo de fluxo de dados (DFD)
 - Modelo entidade-relação-atributo (MER)
 - Modelo estrutural
 - Modelo de objetos

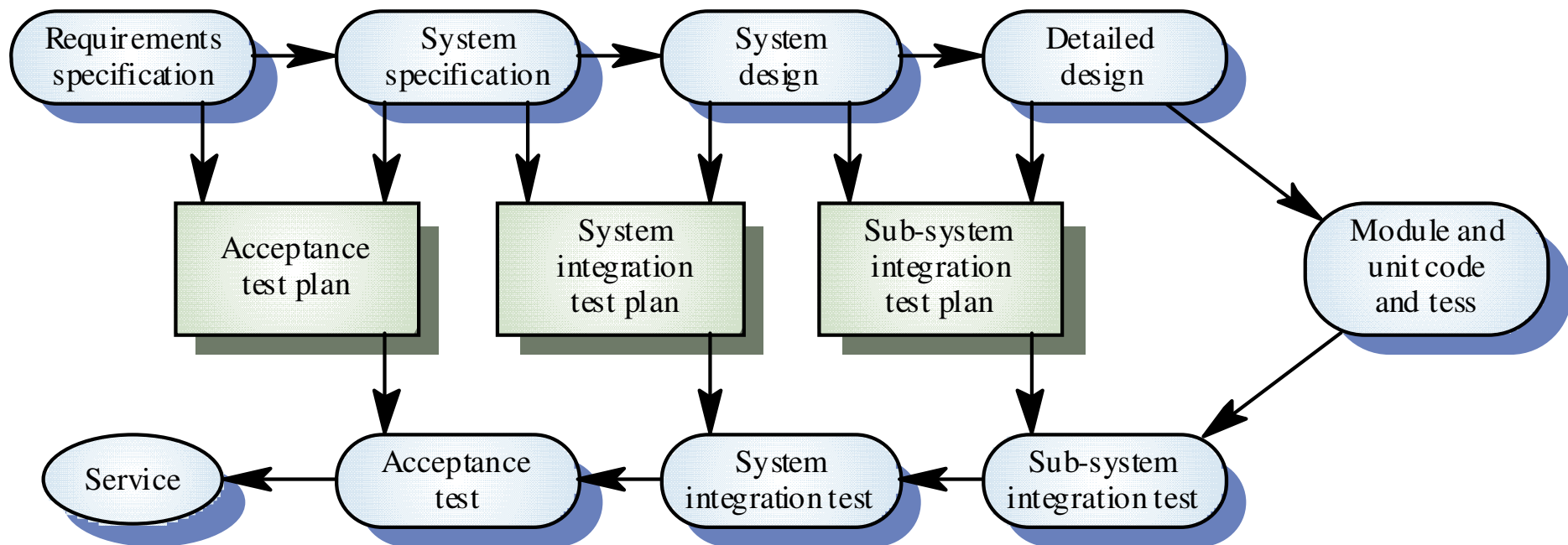
Validação de Software

- Verificação e validação objetivam mostrar que
 - O sistema está em **conformidade com sua especificação**
 - e **satisfaz os requisitos** do usuário/cliente
- Envolve checar e revisar processos e teste de sistema
- **Teste de sistema** envolve executar o sistema com casos de teste que são derivados da especificação de dados reais a serem processados pelo sistema

Um processo (geral) de Testes

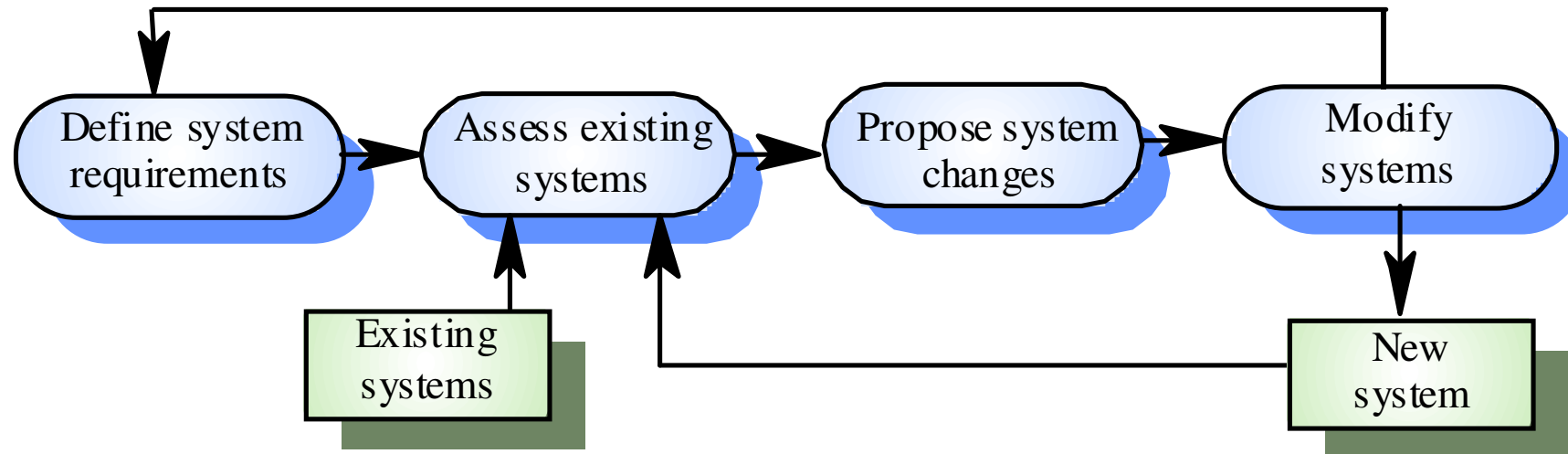


Fases de Teste



©Ian Sommerville 2000 Software Engineering, 6th edition. Chapter 1

Evolução de Sistema



©Ian Sommerville 2000 Software Engineering, 6th edition. Chapter 1

Custos de Software

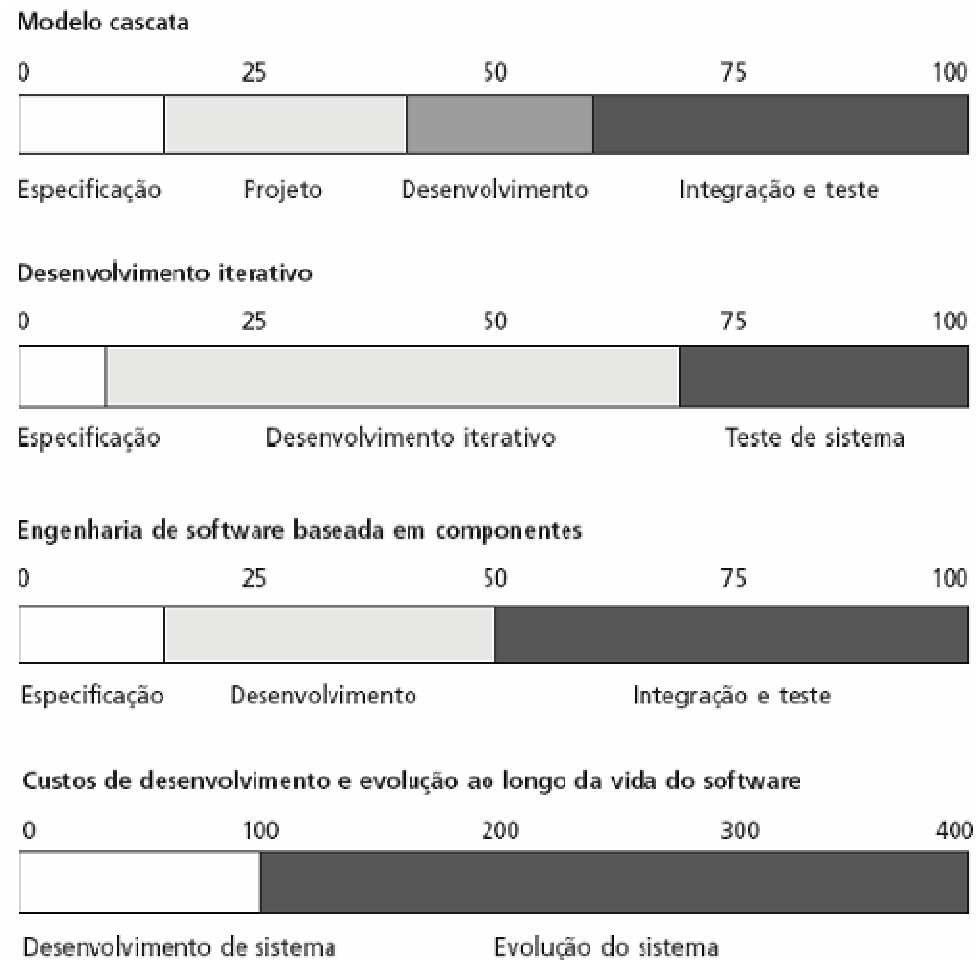
- Em geral dominam custo do sistema.
 - Os custos do software em um PC em geral são maiores que o custo do hardware
 - Para sistemas de vida longa, manutenção custa várias vezes mais que custos de desenvolvimento
- Engenharia de Software compreende desenvolvimento de software com custo viável

Quais são os custos da engenharia de software?

- 60% dos custos são de desenvolvimento
- 40% são custos de testes.
- Para software personalizado custos da evolução excedem custos do desenvolvimento
- Custos variam dependendo do tipo de sistema sendo desenvolvido e dos requisitos e atributos do sistema (performance, segurança)
- Distribuição dos custos depende do modelo de desenvolvimento utilizado

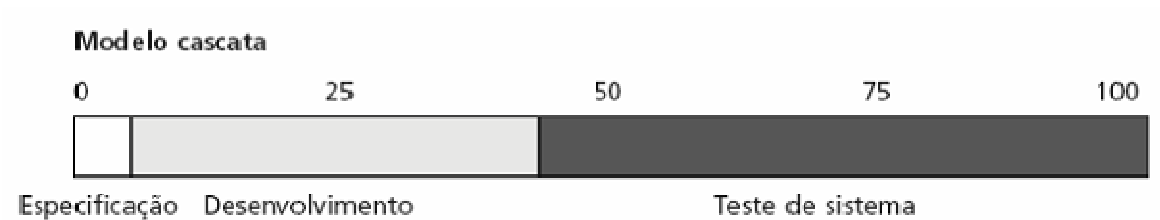
Distribuição de custos nas atividades

Figura 1.1
Distribuição de custos nas atividades de engenharia de software



Custos de desenvolvimento de produto

Figura 1.2
Custos de desenvolvimento do produto



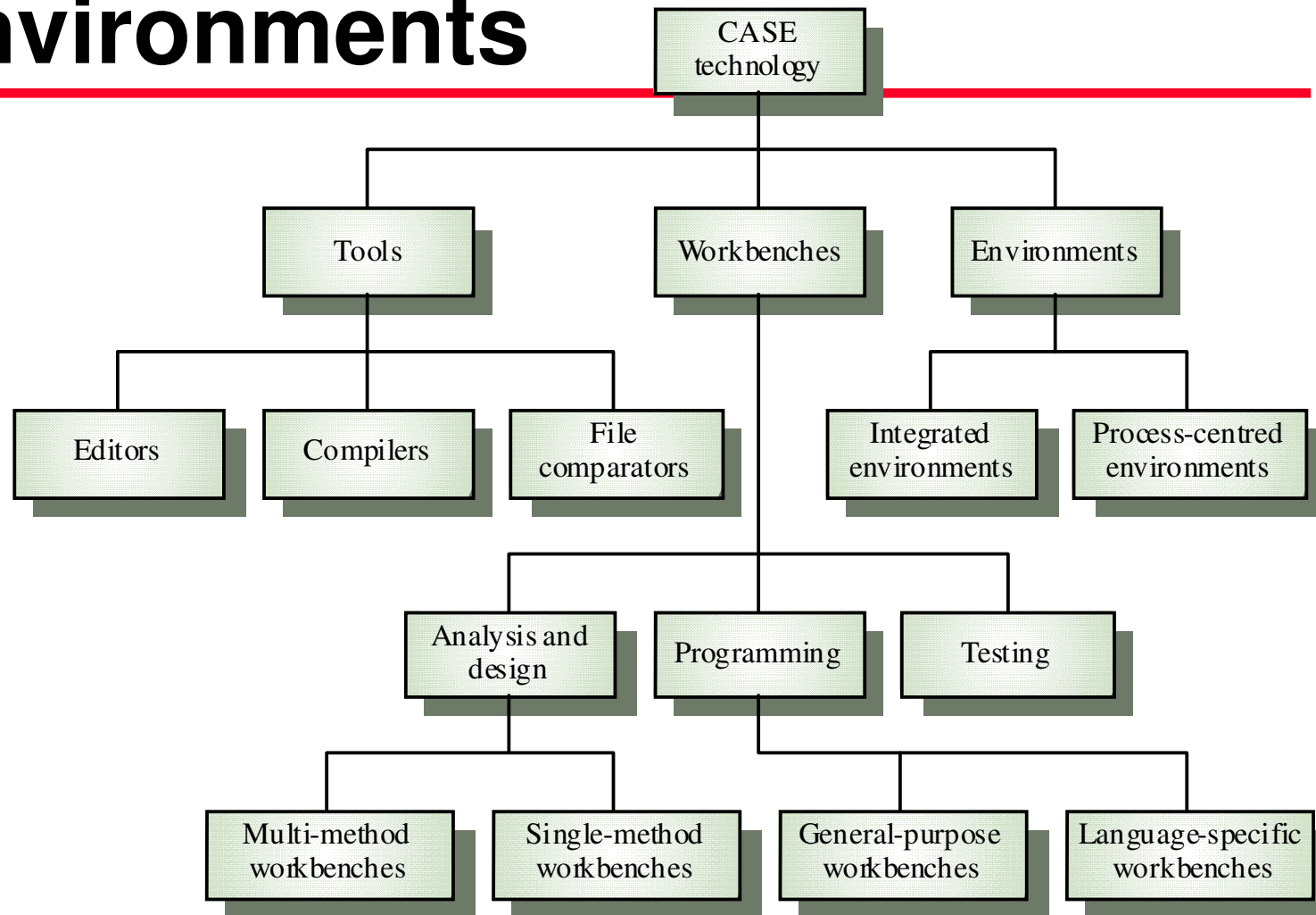
Suporte automatizado ao processo (CASE)

- Computer-aided software engineering (CASE) é software para apoio aos processos de desenvolvimento e evolução de software
- Automação de atividades:
 - Editores gráficos para modelagem
 - Dicionário de dados para gerenciar as entidades do projeto
 - Construtores gráficos para construção da UI
 - Debuggers

Tecnologia CASE

- Tools
 - Suporte a tarefas individuais como consistência do projeto, edição de texto, etc.
- Workbenches
 - Suporte a fases como especificação ou design
 - Normalmente incluem ferramentas integradas
- Environments
 - Suporte a partes substanciais do processo de software
 - Normalmente incluem vários workbenches integrados

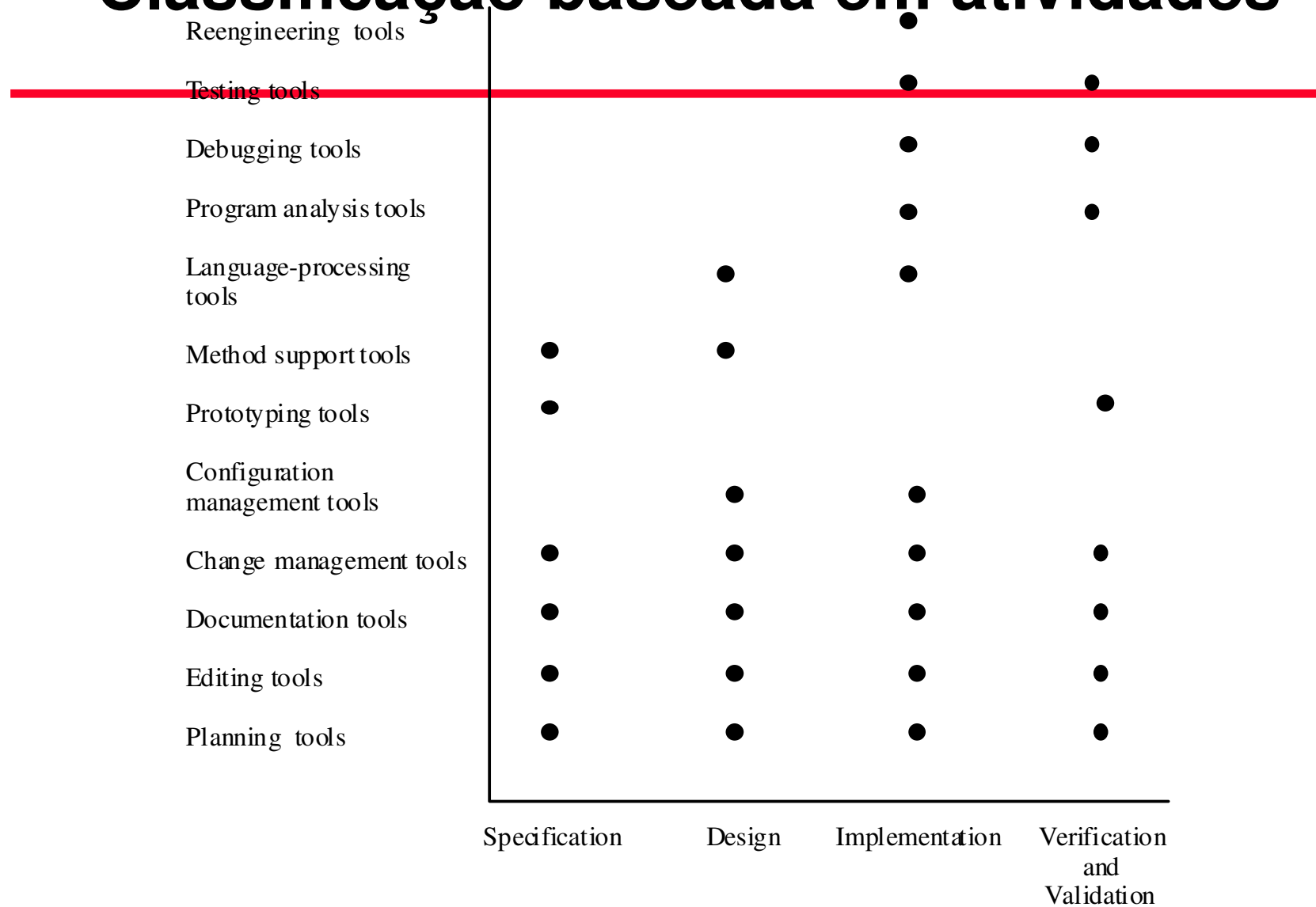
Tools, workbenches, environments



Based on ©Ian Sommerville 2000 Software Engineering, 6th edition. Chapter 1

MC 426 IC Unicamp – M. Cecilia C. Baranauskas

Classificação baseada em atividades



Síntese

- **Processos de Software** são atividades envolvidas na produção e evolução de um sistema de software
 - São representados em um **Modelo de Processo de Software**
- Atividades gerais envolvem:
 - Especificação, design, implementação, validação e evolução
- Processos genéricos descrevem uma dada organização das atividades de desenvolvimento de software
- **Modelos Iterativos** de processo descrevem o processo de software como um ciclo de atividades

Síntese

- **Engenharia de Requisitos** é um processo de desenvolvimento de especificação de software
- Processos de design e implementação transformam a especificação em um programa executável
- **Validação** envolve checar se o sistema está em conformidade com sua especificação e necessidades do usuário
- Evolução refere-se a modificações no sistema depois que está em uso
- Tecnologias CASE apoiam processos e atividades de desenvolvimento de software

Referências

- ©Ian Sommerville 2001 Software Engineering, 6th edition. Chapter 3
- ©Ian Sommerville 2007 Engenharia de Software, 8ª. edição. Chapter 4, Pearson, Addison Wesley
- Carvalho, A.M.B.R.; Chiossi, T.C.S. 2001, Introdução à Engenharia de Software